

Article 26 as Runtime Governance

Evidence of Control, Human Oversight and Deployer Responsibility under the EU AI Act

From Compliance Documentation to Evidence-Based AI Control

AUTHOR	Jozsef Fodor
PROJECT / FRAMEWORK	NextOne Observatory
DOCUMENT TYPE	Professional Study / Operational Governance Framework
VERSION	V1.0 - Final Public Release
DATE	16 July 2026
STRUCTURE	34 main chapters plus Appendices A-F
CANONICAL PAGE	https://nextone.ie/observatory/article26/

Publication Status

Final Public Release. This V1.0 incorporates the reconstructed editable source, updated document control, article-level legal anchors and a current official-source record. Author approval was recorded on 16 July 2026, followed by final visual and structural QA and evidence-package generation.

Document Control

Document Title	Article 26 as Runtime Governance
Subtitle	Evidence of Control, Human Oversight and Deployer Responsibility under the EU AI Act
Author	Jozsef Fodor
Project / Framework	NextOne Observatory
Version	V1.0 - Final Public Release
Previous Version	V0.9 - Public Review Draft (20 June 2026)
Status	Final Public Release - published as V1.0
Intended Release Status	V1.0 - Final Public Release
Primary Focus	Article 26 deployer obligations for high-risk AI systems
Secondary Focus	Runtime governance, evidence of control, human oversight, input data governance, monitoring, logs, incident response, agentic AI, context governance, fallback governance, authority continuity, board oversight and audit readiness
Intended Audience	AI deployers, governance teams, compliance, risk, legal, internal audit, boards, technology leaders, AI platform teams and regulated-sector operators
Canonical Page	https://nextone.ie/observatory/article26/
Copyright	Copyright © 2026 Jozsef Fodor. All rights reserved.
Law and guidance reviewed as at	16 July 2026

Disclaimer and Use Notice

This study is provided for research, governance design, operational analysis and professional discussion purposes only. It is not legal advice.

It does not replace professional legal, regulatory, compliance, technical, sector-specific, supervisory, audit, procurement, data protection or risk-management advice. Organizations should obtain appropriate legal and professional advice before relying on this material for formal compliance decisions.

This study is based on an operational governance interpretation of deployer obligations under Regulation (EU) 2024/1689, commonly referred to as the EU Artificial Intelligence Act. The official legal text, including any corrigenda, amendments, implementing acts, delegated acts, harmonised standards or official supervisory guidance, remains the authoritative source.

Where this study uses terms such as should, must, control, evidence, validity, pause, refuse, human-in-control or runtime governance, readers should distinguish between official legal requirements, context-dependent obligations, operational interpretations and NextOne Observatory framework concepts.

Nothing in this study should be understood as guaranteeing compliance with the EU AI Act or any other law, regulation, standard, supervisory expectation or contractual obligation.

Source-currency boundary. Formal legal propositions are anchored to the official text of Regulation (EU) 2024/1689 and any formally published amendment. As at 16 July 2026, the Commission high-risk classification guidelines remain draft and non-binding, with consultation open until 23 July 2026. The Digital Omnibus on AI received final Council approval on 29 June 2026 and was signed on 8 July 2026. Until Official Journal publication and entry into force are verified, this study relies on formally published legal text for operative propositions and treats the adopted amendment as current legislative context.

Abstract

This study examines Article 26 of the EU AI Act as more than a deployer compliance checklist. It argues that Article 26 can be understood as an operational runtime governance model for high-risk AI systems.

Legal basis for the abstract. [Regulation \(EU\) 2024/1689, Article 26](#) is the central legal anchor. Evidence of control, continued validity, human-in-control, authority continuity and ALLOW / HOLD / ESCALATE / REFUSE are operational interpretations or NextOne Observatory framework concepts, not statutory terms.

The central thesis is that a deployer should be able to prove that a high-risk AI system was used under valid, monitored, human-controllable and reconstructable conditions when its output influenced consequence.

The study develops an operational interpretation of deployer responsibility around evidence of control, continued validity, human-in-control oversight, input data governance, runtime monitoring, log retention, incident response, authority continuity, context governance, agentic AI and board-level accountability.

Across 34 chapters and six appendices, the study proposes a practical governance framework built around four runtime states: ALLOW, HOLD, ESCALATE and REFUSE.

Core Operating Sentence

A deployer should be able to prove that a high-risk AI system was used under valid, monitored, human-controllable and reconstructable conditions when its output influenced consequence.

Executive Summary

Most AI governance failures do not begin with the absence of principles. They begin when principles are not translated into operational control.

Organizations may have policies, governance committees, system registers, vendor documents, training records and monitoring dashboards. These are important, but they do not automatically prove that a high-risk AI system was controlled when its output affected a real workflow, person, process or decision.

This study argues that Article 26 of the EU AI Act should be read not only as a deployer compliance provision, but as an operational governance challenge: can the deployer prove that the AI-supported workflow was allowed to continue under valid control conditions?

The study develops ten central ideas: evidence of control; continued validity; control before consequence; human-in-control; runtime governance; ALLOW / HOLD / ESCALATE / REFUSE; evidence architecture; authority continuity; context as control; and decision traceability.

The final conclusion is that Article 26 is not only about AI compliance. It is about evidence-based runtime control. A mature deployer should not only know that an AI system produced an output. It should know why that output was allowed to matter.

How to Use This Study

Audience	Suggested Use
Deployers	Use as a practical framework for understanding Article 26 operationally. Start with the readiness matrix, evidence pack templates and 90-day roadmap.
Governance, Risk and Compliance Teams	Translate legal and policy obligations into evidence-producing controls. Focus on the toolkit chapters and appendices.
Technical and AI Platform Teams	Use runtime governance and evidence architecture sections to understand logging, monitoring, context records, fallback records and decision trails.
Human Oversight Roles	Assess whether oversight is meaningful or merely procedural. Focus on competence, authority, information, time and ability to intervene.
Boards and Senior Management	Use the executive briefing, red flags and dashboard sections to understand AI control maturity and governance exposure.
Internal Audit	Use the audit questions, case sampling approach and evidence quality scale to test whether controls operate in practice.
Legal and Data Protection Teams	Use this study as an operational discussion aid, not as legal advice. Verify all legal interpretations against official sources and applicable guidance.

Version History

Version	Date	Description
V0.1	17 June 2026	Full draft assembly completed: 34 main chapters plus Appendices A-F.
V0.9	20 June 2026	Public review draft prepared with revised front matter, publication status, copyright notice and canonical page reference.
V1.0	16 July 2026	Final Public Release approved by the author, with reconstructed editable master, updated release control, legal anchors, current source-status record and publication evidence.

Copyright and Attribution Notice

Copyright © 2026 Jozsef Fodor. All rights reserved. This work may be cited, referenced and shared with clear attribution to the author and the NextOne Observatory framework.

Suggested citation: Fodor, Jozsef. Article 26 as Runtime Governance: Evidence of Control, Human Oversight and Deployer Responsibility under the EU AI Act. NextOne Observatory, 2026.

Final Table of Contents

Section	Page
Front Matter	1
Part I - Article 26 Foundations and Core Deployer Obligations	
Chapter 1 - Why Article 26 Matters	7
Chapter 2 - The Legal Position of the Deployer	20
Chapter 3 - High-Risk AI Systems and the Deployer Context	34
Chapter 4 - Use According to Instructions	48
Chapter 5 - Human Oversight	64
Chapter 6 - Input Data Relevance and Representativeness	81
Chapter 7 - Monitoring the Operation of High-Risk AI Systems	97
Chapter 8 - Logs, Records and Evidence Retention	113
Chapter 9 - Serious Risk, Incident and Escalation Pathways	133
Chapter 10 - Workplace AI and Affected-Person Information	149
Part II - Evidence, Runtime Control and Decision Validity	
Chapter 11 - Evidence of Control	165
Chapter 12 - Runtime Governance	182
Chapter 13 - Authority, Scope and Operational Boundaries	200
Chapter 14 - Evidence Freshness and Decision Validity	218
Chapter 15 - Pause, Escalate and Refuse Conditions	235
Part III - Sector Applications	
Chapter 16 - Financial Services and Capital Markets	251
Chapter 17 - Human Resources and Employment	273
Chapter 18 - Procurement and Supplier Risk	292
Chapter 19 - Healthcare and Clinical Decision Support	312
Chapter 20 - Education, Public Sector and Citizen-Facing AI	332
Part IV - Advanced Runtime Governance Challenges	
Chapter 21 - Agentic AI and Article 26	351
Chapter 22 - Model Substitution, Fallback and Authority Continuity	372
Chapter 23 - Context Management as Control Management	389
Chapter 24 - Reasoning Layers, Traceability and Hybrid AI	406

Final Table of Contents - Continued

Section	Page
Part V - Practical Implementation Toolkit	
Chapter 25 - Article 26 Deployer Readiness Matrix	422
Chapter 26 - Human Oversight Checklist	451
Chapter 27 - Evidence of Control Template	475
Chapter 28 - Runtime Monitoring and Incident Response Checklist	498
Chapter 29 - Board Briefing and Executive Summary Template	535
Chapter 30 - Audit Questions and Internal Review Guide	558
Part VI - Strategic Interpretation and Final Conclusions	
Chapter 31 - From Human-in-the-Loop to Human-in-Control	584
Chapter 32 - From Compliance to Evidence Architecture	605
Chapter 33 - Article 26 as Runtime Governance	627
Chapter 34 - Final Conclusions and Future Outlook	649
Appendices	
Appendix A - Article 26 Clause-by-Clause Operational Mapping	666
Appendix B - ALLOW / HOLD / ESCALATE / REFUSE Decision Tables	685
Appendix C - Evidence Pack Templates and Sample Forms	706
Appendix D - Glossary of Key Terms	738
Appendix E - 90-Day Implementation Roadmap	770
Appendix F - Final Checklist and Document Control Page	795
Legal Sources, Guidance Status and Citation Record	810

Structure Note

This final public release contains 34 main chapters plus Appendices A-F.

The chapter and appendix structure has been verified against the V0.9 publication and the recovered editable source files.

Canonical page: <https://nextone.ie/observatory/article26/> | Law and guidance reviewed as at 16 July 2026.

Chapter 1

Why Article 26 Matters

Legal anchor and framework boundary. [Regulation \(EU\) 2024/1689](#); [Article 26](#). This chapter develops an operational interpretation of deployer responsibility; runtime governance and evidence of control are framework concepts.

1.1 Article 26 as a Practical Governance Article

Article 26 of the EU AI Act may become one of the most important provisions for practical AI governance.

Not because it is the most visible article in the regulation.

Not because it creates the most dramatic headline.

And not because it is only a legal compliance requirement.

Article 26 matters because it moves responsibility into the operational environment.

It asks what happens when a high-risk AI system is no longer only designed, marketed, procured, tested or documented, but actually used inside an organization.

That shift is important.

Many discussions about AI governance begin with the AI system itself. They ask whether the model is accurate, whether the provider has met its obligations, whether the system was trained responsibly, whether documentation exists, whether technical standards were followed, and whether the system satisfies the requirements expected before it is placed on the market or put into service.

Those questions are necessary.

But they are not enough.

Once an AI system enters a real organizational workflow, a different set of questions appears.

Who is using the system?

For what purpose?

Under whose authority?

With what input data?

Under what oversight?

Inside which process?

With what consequences?

What happens if the system is used outside the provider's intended instructions?

What happens if the context changes after approval?

What happens if the person responsible for oversight is present in name only, but not practically able to intervene?

What happens if the AI output becomes an operational signal for another workflow?

What happens if the log exists, but does not preserve enough context to explain why the system was allowed to continue?

These are deployer questions.

This is why Article 26 matters.

It places attention on the organization using the high-risk AI system. It recognizes that deployment is not a passive act. The deployer is not merely the recipient of a product. The deployer becomes part of the governance environment in which the AI system produces effects.

In this sense, Article 26 should not be read only as a checklist of obligations.

It should be read as a practical test of whether the organization can operate high-risk AI systems under valid control conditions.

This is a different standard from simply possessing documentation.

A policy can state that human oversight exists.

But Article 26 raises the deeper operational question:

Was oversight actually capable of functioning when the system was used?

A procedure can state that input data should be appropriate.

But the deeper question is:

Were the actual input conditions valid for the intended purpose at the moment of use?

A monitoring process can be described in an internal document.

But the deeper question is:

Could the organization detect when the system, data, workflow, authority or risk context had changed enough to require reassessment?

A log retention policy can exist.

But the deeper question is:

Can the retained logs reconstruct the conditions under which the AI-supported workflow became consequential?

This is where Article 26 begins to point beyond traditional compliance.

It points toward operational governance.

It points toward runtime evidence.

It points toward human oversight as a practical capability.

It points toward the need to prove that control conditions were not only designed, but active and valid when the AI system was used.

For this reason, Article 26 is not only a deployer obligation.

It is a blueprint for operational AI governance.

1.2 From Provider Responsibility to Deployer Responsibility

The EU AI Act places many obligations on providers of high-risk AI systems. That is understandable. Providers design, develop, train, test, document, place on the market or put into service the AI systems that others may later use.

But the provider is not the only actor that matters.

An AI system can be designed for one intended purpose and then used in a very different operational setting.

It can be technically compliant at the point of deployment, yet later used with unsuitable data.

It can be supplied with instructions, yet those instructions may be misunderstood, ignored or translated poorly into internal procedures.

It can include human oversight measures, yet the deploying organization may fail to assign people with the competence, authority, time or information needed to use those measures.

It can produce outputs that are meant to support decision-making, yet those outputs may become de facto decisions inside automated or semi-automated workflows.

It can be monitored at a high level, yet operational drift may occur beneath the threshold of formal incident detection.

This is why the deployer role matters.

The deployer is the actor closest to actual use.

The deployer knows, or should know, how the AI system is being inserted into organizational reality. The deployer knows the workflow, the users, the internal authority structures, the data sources, the operational dependencies, the downstream effects and the practical consequences.

A provider can describe the intended use of a system.

But the deployer controls the actual use.

A provider can design oversight features.

But the deployer determines whether oversight is practically assigned, trained, resourced and respected.

A provider can explain logging capabilities.

But the deployer determines whether logs are retained, connected to internal processes and available for audit or incident reconstruction.

A provider can define instructions.

But the deployer determines whether those instructions become operational practice.

This is the fundamental importance of Article 26.

It moves AI governance from the product layer into the use layer.

It says, in effect:

The organization using a high-risk AI system must also govern the conditions of use.

That is a powerful shift.

It means organizations cannot outsource the entire governance problem to the provider.

They cannot say:

The vendor supplied the system, therefore the vendor owns the risk.

They cannot say:

The model was compliant, therefore our use of it was automatically compliant.

They cannot say:

The system had oversight features, therefore oversight existed.

They cannot say:

The system produced a log, therefore we have evidence of control.

The deployer must be able to show that the system was used appropriately, monitored properly, supervised meaningfully and operated within valid boundaries.

In high-risk AI environments, this distinction will become critical.

The provider may be responsible for the system as designed.

The deployer is responsible for the system as used.

And the difference between those two states may be where many practical AI governance failures occur.

1.3 Why Deployers Become Part of the Control Architecture

A high-risk AI system does not create risk only because of what it is.

It creates risk because of where it is placed, how it is used, what it affects and how much authority is attached to its output.

The same AI capability can have different consequences in different settings.

An AI system used to rank low-risk internal documents may create one level of concern.

A similar system used to support hiring, credit, healthcare, education, law enforcement, access to public services or financial decisions creates a very different risk profile.

The difference is not only technical.

The difference is operational.

This is why the deployer becomes part of the control architecture.

The deployer determines how the AI system enters the organization's decision environment.

The deployer decides whether the system is used as a recommendation tool, a triage tool, a scoring system, an automation trigger, a monitoring aid, a decision-support tool or a practical decision-maker in all but name.

The deployer determines whether the output is treated as advisory, authoritative, binding, default, escalatory or executable.

The deployer determines whether humans can challenge the output, whether they usually do, whether they have the knowledge to do so, and whether the surrounding workflow makes intervention realistic.

The deployer determines whether the system is connected to downstream tools, case management systems, procurement processes, HR workflows, clinical pathways, financial controls or public-sector services.

This means that AI governance cannot be understood only at the level of the model or the provider.

It must be understood at the level of operational integration.

A model output does not become consequential by itself.

It becomes consequential when an organization allows it to influence a workflow, decision, action, entitlement, risk assessment, prioritization, recommendation, intervention, denial, escalation or allocation.

That is the moment where deployer governance matters.

The deployer is responsible for the environment in which the AI output gains practical meaning.

This includes:

- the organizational process into which the system is inserted;
- the authority granted to the system's output;
- the human oversight assigned around the system;
- the data and context supplied to the system;
- the monitoring mechanisms surrounding its use;
- the escalation procedures available when something changes;
- the logs retained to reconstruct what happened;
- the boundaries that prevent intended use from drifting into unintended consequence.

In this sense, the deployer is not merely using a high-risk AI system.

The deployer is constructing the conditions under which the system becomes operational.

That construction is a control architecture.

It may be formal or informal.

It may be well designed or accidental.

It may be visible or hidden.

It may be documented or implied.

But it exists.

Every deployment creates a control architecture, even when the organization does not explicitly design one.

The question is whether that architecture is safe, visible, accountable and capable of proof.

This is where Article 26 becomes deeply practical.

It forces the deployer to ask:

Have we created a real control environment around this system?

Or have we simply placed a high-risk AI system inside a workflow and assumed that provider compliance, internal policy and human presence will be enough?

For high-risk systems, assumption is not enough.

The deployer must be able to show control.

1.4 The Gap Between Compliance Language and Operational Proof

A major challenge in AI governance is the gap between what organizations can say and what they can prove.

An organization may say that it has human oversight.

But can it prove that the person assigned to oversight had the authority to intervene?

An organization may say that it follows provider instructions.

But can it prove that internal users were trained, that the system was used within its intended scope, and that deviations were detected?

An organization may say that it monitors the system.

But can it prove that monitoring was active, meaningful and connected to escalation?

An organization may say that logs are retained.

But can it prove that the logs preserve the right information, at the right level of detail, for the right period, in a way that supports reconstruction?

An organization may say that the system is used only for decision support.

But can it prove that the output did not become an automatic decision in practice?

This is the difference between compliance language and operational proof.

Compliance language is what an organization writes down.

Operational proof is what the organization can demonstrate.

The gap between the two is where many AI governance failures may occur.

In traditional compliance environments, documentation is often treated as evidence.

That is understandable. Documents matter. Policies matter. Procedures matter.

Training records matter. Vendor records matter. Risk assessments matter.

But AI systems introduce a more dynamic problem.

The conditions under which an AI-supported workflow is valid may change during operation.

Input data may become stale.

A policy may be updated.

A human overseer may move roles.

A downstream tool may change behaviour.

A plugin may gain new capability.

A data source may stop refreshing.

A model may be substituted.

A risk threshold may change.

A workflow may be repurposed.

A recommendation may begin to function as an executable signal.

In those situations, static documentation may not be enough.

The question is not only:

Did the organization have a policy?

The question is:

Was the policy still true at the time the AI-supported workflow became consequential?

The question is not only:

Was approval granted?

The question is:

Was the approval still valid under current conditions?

The question is not only:

Was the system monitored?

The question is:

Did monitoring detect that continuation should be questioned?

The question is not only:

Did the system complete the workflow?

The question is:

Should the workflow have been allowed to continue?

This is why Article 26 should be interpreted through the lens of evidence of control.

Evidence of control is not merely evidence that something existed.

It is evidence that the relevant control conditions were active, valid and inspectable when they mattered.

This includes evidence of:

- system identity;
- intended purpose;
- actual use;
- authority;
- human oversight;
- input data condition;
- monitoring status;
- logs;
- escalation;
- boundary validity;
- decision context;
- continued validity.

The future of AI compliance will require organizations to move from policy possession to proof of operational control.

This does not mean every organization needs excessive bureaucracy.

It means that organizations deploying high-risk AI systems need evidence that matches the level of consequence.

The higher the consequence, the stronger the evidence should be.

This principle is especially important for high-risk AI systems, because the harm may not come from obvious technical failure.

It may come from a workflow that appears normal, completes successfully and produces a plausible output, while the conditions that justified continuation have already changed.

In such cases, the failure is not the absence of activity.

The failure is the absence of a pause condition.

Article 26 should be read as an invitation to build those pause conditions into deployer governance.

1.5 Article 26 as a Bridge Between Legal Obligation and Runtime Control

Article 26 sits at an important boundary.

On one side, it is a legal provision.

It defines obligations for deployers of high-risk AI systems.

On the other side, it points toward operational realities that cannot be solved by legal interpretation alone.

This is why Article 26 should be treated as a bridge.

It connects legal requirements to systems architecture.

It connects compliance to evidence.

It connects human oversight to practical intervention.

It connects log retention to reconstructability.

It connects monitoring to runtime observability.

It connects intended use to operational boundary management.

It connects deployer responsibility to control conditions.

This bridge matters because many organizations will be tempted to treat AI Act compliance as a documentation project.

They may prepare policies.

They may create registers.

They may assign owners.

They may collect vendor documents.

They may conduct training.

They may create risk assessment templates.

All of that is necessary.

But it is not sufficient.

The real test will come when a regulator, auditor, board member, affected person, internal reviewer or incident investigator asks:

Show what happened.

Show who was responsible.

Show what authority existed.

Show what evidence was used.

Show whether the input data was suitable.

Show whether oversight was available.

Show whether the system was monitored.

Show whether the workflow remained within scope.

Show whether the output was allowed to influence the decision.

Show why the system was allowed to continue.

Those questions cannot be answered by policy language alone.

They require operational evidence.

They require traceability.

They require context.

They require logs that are useful, not merely retained.

They require human oversight that is meaningful, not decorative.

They require an organization to know not only what the AI system can do, but when the AI-supported workflow should pause, escalate or stop.

This is the core argument of this study.

Article 26 is not only about deployers obeying instructions.

It is about deployers becoming accountable for the operational conditions under which high-risk AI systems are used.

That accountability requires a new kind of governance thinking.

Not only governance before deployment.

Not only audit after execution.

But runtime governance during use.

A mature Article 26 response should therefore include three layers:

First, legal and policy alignment.

The organization must understand its obligations, classify systems properly, follow provider instructions, assign responsibilities and maintain appropriate documentation.

Second, operational implementation.

The organization must translate those obligations into workflows, roles, training, monitoring, escalation paths, input controls, log retention and oversight mechanisms.

Third, runtime evidence of control.

The organization must be able to show that control conditions were valid at the moment the system was used and when its output became operationally relevant.

This third layer is the most difficult.

It is also the most important.

Without it, Article 26 compliance may remain formal.

With it, Article 26 becomes a foundation for trustworthy deployment.

This is why Article 26 matters.

It is not only a legal duty for deployers.

It is a practical test of whether organizations can operate high-risk AI systems with evidence, oversight and control.

And in the age of agentic AI, autonomous workflows, hybrid decision systems and AI-supported operations, that test will become increasingly important.

The organizations that understand this early will not treat Article 26 as a burden.

They will treat it as a design requirement.

They will build systems that can show:

what was used,

why it was used,

who had authority,

what evidence was available,

what oversight existed,

what changed,

what was logged,

what was escalated,

and whether continuation was still justified.

That is the future of practical AI governance.

Not governance as a document.

Governance as evidence of control.

Chapter 2

The Legal Position of the Deployer

Legal anchor and framework boundary. [Article 3\(4\)](#); [Article 26](#). The legal definition of deployer and the obligations in Article 26 are authoritative; the control-architecture framing is operational interpretation.

2.1 What Is a Deployer Under the EU AI Act?

The deployer is one of the most important practical actors in the EU AI Act.

The provider creates, develops, places on the market or puts into service an AI system.

The deployer uses it.

That distinction may sound simple, but in high-risk AI governance it becomes extremely important.

A deployer is not merely a passive recipient of technology. A deployer is the organization, public authority, agency, body or other person using an AI system under its own authority, except where the system is used in the course of a purely personal non-professional activity.

This means the deployer is the actor responsible for the real operational use of the AI system.

The deployer is the actor that decides how the system enters a workflow.

The deployer determines who uses it.

The deployer determines whether the system is used in employment, finance, healthcare, education, public services, law enforcement, procurement, compliance, risk management or another operational setting.

The deployer determines how much practical influence the AI output has.

The deployer determines whether the output remains advisory or becomes operationally decisive.

The deployer determines whether the system is surrounded by meaningful human oversight or whether oversight exists only in policy language.

The deployer determines whether the system is monitored during operation.

The deployer determines whether logs are preserved in a way that can support later reconstruction.

This is why the deployer role is not a secondary role.

It is the role closest to consequence.

A provider may define the intended purpose of a system, but the deployer determines the actual use of the system in an organization.

A provider may supply instructions, but the deployer determines whether those instructions are converted into training, process design, controls and daily practice.

A provider may build logging capability, but the deployer determines whether logs are retained, protected, reviewed and connected to accountability.

A provider may design oversight measures, but the deployer determines whether oversight is assigned to competent people with the authority and support to intervene.

The deployer therefore sits at the point where a high-risk AI system becomes real.

This is the central reason why Article 26 matters.

It recognizes that AI risk does not end when the system leaves the provider.

AI risk continues when the system is used.

It may even increase when the system enters an organizational workflow, because the deployer's environment determines the system's practical authority, data context, oversight model and downstream effect.

For this reason, the legal position of the deployer should be understood in operational terms.

The deployer is the actor responsible for ensuring that high-risk AI is not only procured or switched on, but used under valid conditions.

2.2 Deployer Versus Provider

The provider and deployer occupy different positions in the AI value chain.

The provider is responsible for the AI system as designed, documented, tested, placed on the market or put into service.

The deployer is responsible for the AI system as used.

This distinction is fundamental.

The provider's obligations focus heavily on system design, conformity, documentation, risk management, technical performance, instructions for use, transparency, logging capability, corrective action and market-facing responsibility.

The deployer's obligations focus on appropriate use, human oversight, input data relevance where under the deployer's control, monitoring, logs under the deployer's control, escalation, workplace notification, affected-person information and cooperation with relevant authorities.

In simpler terms:

The provider must make the system compliant.

The deployer must use the system responsibly.

But the relationship is not one-directional.

The deployer depends on the provider's documentation, instructions, transparency, technical design and system capabilities.

The provider depends on deployers to use the system in a way that remains consistent with the intended purpose and instructions for use.

This creates a shared governance surface.

The provider can define the intended purpose.

The deployer can drift from it.

The provider can define instructions.

The deployer can operationalize them well or badly.

The provider can design human oversight measures.

The deployer can make them meaningful or decorative.

The provider can supply information to support interpretation.

The deployer can use that information to support oversight, or fail to do so.

The provider can build logs.

The deployer can preserve them or lose their evidential value.

This means that the provider-deployer relationship is not merely contractual.

It is operational.

It defines how AI governance crosses from product compliance into organizational use.

The most serious governance failures may occur when the provider assumes the deployer will manage operational context, while the deployer assumes the provider's compliance is enough.

That gap is dangerous.

A high-risk AI system may be technically compliant as a product, but poorly governed as a deployed system.

The provider may have performed the required conformity work, but the deployer may use the system in a way that changes its risk profile.

The provider may have supplied clear instructions, but the deployer may fail to integrate those instructions into workflow controls.

The provider may have designed oversight measures, but the deployer may assign oversight to people who lack time, training, authority or practical ability to challenge the output.

The provider may have enabled logging, but the deployer may not retain logs in a way that supports audit or incident reconstruction.

This is why Article 26 must be read together with provider obligations.

The provider and deployer are different actors, but their obligations meet in the operational environment.

The provider supplies the system.

The deployer supplies the context.

The provider defines the intended use.

The deployer controls actual use.

The provider may create the technical possibility of governance.

The deployer determines whether governance becomes real.

2.3 Deployer Versus User

In ordinary language, people often say “user” when they mean anyone who uses a system.

In AI governance, this is not precise enough.

A deployer is not simply an individual end-user pressing a button.

A deployer is the organization or body using the AI system under its authority.

This distinction matters because high-risk AI systems are usually embedded in organizational processes, not isolated individual actions.

For example, a recruitment officer may interact with an AI hiring tool.

But the deployer is likely the employer or organization using that tool in its recruitment process.

A nurse or doctor may interact with a clinical decision-support system.

But the deployer is likely the healthcare organization that has put the system into use.

A compliance analyst may interact with an AI transaction-monitoring tool.

But the deployer is likely the financial institution using that system inside its compliance operations.

A public official may interact with an AI system used to support public-service decisions.

But the deployer is likely the public authority responsible for that use.

This matters because Article 26 is not only concerned with individual usage behaviour.

It is concerned with organizational responsibility.

The deployer must organize the conditions of use.

The deployer must assign oversight.

The deployer must monitor operation.

The deployer must retain logs where under its control.

The deployer must ensure input data relevance where it controls input data.

The deployer must inform workers or affected persons where required.

The deployer must suspend use and escalate where risk conditions require it.

These are not responsibilities that can be reduced to the individual user.

They require governance structure.

They require management decisions.

They require technical controls.

They require training.

They require record-keeping.

They require internal accountability.

They require operational discipline.

The difference between a deployer and an individual user is therefore crucial.

The individual user may interact with the system.

The deployer is responsible for the environment in which that interaction becomes consequential.

This includes the workflow design, role assignment, access control, oversight structure, escalation procedure, data pipeline, logging policy and decision pathway.

If a high-risk AI system is misused, the question will not only be:

Did an individual use the tool incorrectly?

The question will also be:

Did the deployer create a governance environment in which correct use was likely, supported, monitored and evidenced?

This is the operational meaning of deployer responsibility.

2.4 Deployer Obligations for High-Risk AI Systems

Article 26 sets out specific obligations for deployers of high-risk AI systems.

These obligations should not be read as isolated compliance tasks.

They should be read as components of an operational control model.

The main deployer obligations include:

- using the high-risk AI system in accordance with the instructions for use;
- assigning human oversight to natural persons with necessary competence, training, authority and support;
- ensuring input data is relevant and sufficiently representative, to the extent the deployer controls that input data;
- monitoring the operation of the high-risk AI system on the basis of the instructions for use;

- informing the provider, distributor or market surveillance authority where relevant risk conditions are identified;
- suspending use where the system may present a relevant risk;
- informing relevant parties in the event of serious incidents;
- keeping automatically generated logs to the extent they are under the deployer's control;
- informing workers and workers' representatives before using high-risk AI systems in the workplace;
- complying with registration obligations where the deployer is a public authority or Union institution, body, office or agency;
- using provider information to support data protection impact assessment obligations where applicable;
- informing natural persons where Annex III high-risk AI systems are used to make or assist decisions relating to them, where applicable.

Each of these obligations has an operational meaning.

Using the system according to instructions means the deployer must understand the provider's instructions and convert them into internal controls.

Human oversight means the deployer must identify real people with competence, authority and support.

Input data relevance means the deployer must understand whether the data it supplies fits the system's intended purpose.

Monitoring means the deployer must observe operation, detect problems and act when conditions change.

Log retention means the deployer must preserve automatically generated logs where under its control and ensure they remain useful for reconstruction.

Worker notification means the deployer must ensure that affected workers and representatives are not left unaware that high-risk AI systems are being used.

Affected-person information means the deployer must recognize when AI systems are involved in decisions or decision assistance relating to natural persons.

These obligations create a practical governance burden.

But they also create an opportunity.

A mature deployer can use Article 26 as the basis for a strong operational AI governance architecture.

Instead of asking only:

What documents do we need?

The deployer should ask:

What operational evidence do we need to prove that these obligations were active when the system was used?

For each obligation, the deployer should identify an evidence object.

For use according to instructions, the evidence may include internal use procedures, user training, system configuration records and workflow mapping.

For human oversight, the evidence may include named oversight roles, competence records, escalation paths, intervention logs and review outcomes.

For input data relevance, the evidence may include data source records, freshness checks, quality controls, representativeness assessment and intended-purpose mapping.

For monitoring, the evidence may include monitoring dashboards, alert records, performance reviews, risk triggers and suspension decisions.

For logs, the evidence may include retained system logs, access records, output records, decision trails and retention controls.

For notification, the evidence may include communication records, employee notices, affected-person notices and policy acknowledgements.

This is how Article 26 moves from obligation to evidence architecture.

Each legal requirement should become an operational control.

Each operational control should produce evidence.

Each evidence object should support reconstruction.

Each reconstruction should help answer the core question:

Was the system used under valid control conditions when its output became operational?

2.5 The Operational Consequences of Deployer Responsibility

The deployer's legal position has direct operational consequences.

It affects procurement.

It affects system design.

It affects workflow integration.

It affects human oversight.

It affects logging.

It affects monitoring.

It affects data governance.

It affects incident response.

It affects audit readiness.

It affects board responsibility.

It affects how organizations should think about AI risk.

The first operational consequence is that deployers need to understand intended purpose.

An organization cannot responsibly deploy a high-risk AI system if it does not understand what the provider intended the system to be used for, under what conditions, and with what limitations.

Intended purpose is not a marketing phrase.

It is a boundary.

If actual use drifts beyond intended purpose, the deployer may move into a different risk condition.

The second operational consequence is that deployers need internal AI system registers.

An organization cannot govern what it cannot identify.

For each high-risk AI system, the deployer should know:

- system name;
- provider;
- intended purpose;
- actual use;
- business owner;
- technical owner;
- process owner;
- oversight owner;
- data sources;

- user groups;
- affected persons;
- downstream systems;
- logging capability;
- monitoring approach;
- escalation pathway;
- review date;
- risk classification.

The third operational consequence is that deployers need evidence mapping.

For every Article 26 obligation, the deployer should be able to identify:

- the control;
- the owner;
- the evidence produced;
- the storage location;
- the review cycle;
- the escalation trigger;
- the audit use.

The fourth operational consequence is that human oversight must become operationally real.

A human cannot be “in the loop” only as a symbolic reassurance.

The human must have competence.

The human must have authority.

The human must have enough information.

The human must have enough time.

The human must have a clear escalation path.

The human must be able to disagree with the system.

The human must be able to pause or stop the workflow where necessary.

If those conditions are missing, oversight may exist in form but not in substance.

The fifth operational consequence is that monitoring must include context.

Many organizations may monitor model performance, uptime or error rates.

That is not enough.

For high-risk AI systems, deployers may also need to monitor whether the operational context remains valid.

Has the intended use changed?

Has the policy changed?

Has the input data source changed?

Has the human overseer changed?

Has the downstream effect changed?

Has the system been connected to new tools?

Has the output become more influential than originally intended?

Has the workflow moved from advisory use to practical automation?

These are deployer-level monitoring questions.

They are not always model-performance questions.

The sixth operational consequence is that deployers must prepare for regulator-facing proof.

A regulator may not only ask whether a policy exists.

A regulator may ask for evidence.

Who was accountable?

Who used the system?

What was the system used for?

What instructions applied?

What oversight existed?

What data was used?

What logs were retained?

What monitoring occurred?

What incidents were identified?

What actions were taken?

Was the system suspended when needed?

Were affected persons informed?

Could the organization reconstruct the workflow?

This means deployer responsibility is not only about internal compliance.

It is about external explainability and audit readiness.

The seventh operational consequence is that deployer governance must address change.

AI systems do not remain static.

Organizations do not remain static.

Data sources change.

Policies change.

Models change.

Vendors change.

Users change.

Workflows change.

Permissions change.

Downstream integrations change.

Risk appetite changes.

Regulatory expectations change.

This means deployer governance must include reassessment.

A system that was appropriate at deployment may not remain appropriate forever.

An approval that was valid at one point may not remain valid under changed conditions.

A human oversight model that worked at small scale may fail under operational pressure.

A context that supported safe use may later become incomplete, stale or misleading.

This is why deployer responsibility should be understood as continuing responsibility.

Not only pre-deployment responsibility.

Not only procurement responsibility.

Not only documentation responsibility.

Continuing responsibility.

The deployer must be able to ask:

Does this still belong here under current conditions?

Does this approval still apply?

Does this oversight model still work?

Does this input data still support the intended purpose?

Does this system still operate within its original boundaries?

Does this output still deserve the authority attached to it?

Should the workflow continue, pause, escalate or stop?

These questions define the practical meaning of Article 26.

They are also the foundation for runtime governance.

Chapter 2 Summary

The deployer is the actor closest to actual use and operational consequence.

The provider is responsible for the system as designed and supplied.

The deployer is responsible for the system as used.

This distinction is central to Article 26.

A high-risk AI system does not become safe merely because it was provided with documentation, instructions, conformity markings or technical features.

It becomes governable only when the deploying organization converts those elements into working operational controls.

The deployer must therefore be able to demonstrate appropriate use, meaningful human oversight, relevant input conditions, active monitoring, useful log retention, escalation pathways, notification where required and evidence that the system remained within valid boundaries.

This is why the deployer's legal position cannot be separated from operational governance.

The deployer is not only a legal category.

The deployer is part of the control architecture.

And Article 26 is the point where that control architecture becomes visible.

Chapter 3

High-Risk AI Systems and the Deployer Context

Legal anchor and framework boundary. [Article 6](#); [Annex III](#); [Article 26](#). Classification depends on the Regulation and current official guidance; the chapter's consequence and context analysis is operational interpretation.

3.1 What Makes a System High-Risk?

A high-risk AI system is not high-risk simply because it uses advanced technology.

It is high-risk because of the role it plays in human, institutional, operational or safety-critical contexts.

This distinction matters.

Many organizations still think about AI risk in terms of technical capability.

They ask:

How powerful is the model?

How accurate is it?

How large is it?

How complex is it?

How autonomous is it?

How advanced is the underlying architecture?

Those questions are important, but they do not fully explain why an AI system becomes high-risk under the EU AI Act.

The deeper issue is consequence.

An AI system becomes high-risk when its use can affect health, safety, fundamental rights, access to opportunity, legal standing, public services, employment, education, financial access, critical infrastructure, justice, migration, democratic participation or other high-consequence areas.

In other words, high-risk status is not only about the AI system's internal technical design.

It is about the relationship between the system and the world it influences.

A simple model used in a sensitive context may create serious governance concerns.

A technically complex system used in a low-consequence experimental environment may create fewer immediate Article 26 concerns.

A scoring tool used to rank entertainment preferences may not create the same risk as a scoring tool used to rank job applicants, students, patients, loan applicants, public-benefit claimants or people subject to law-enforcement assessment.

The same AI capability can therefore have different governance significance depending on where it is deployed.

This is why deployer context matters.

The provider may describe a system in technical terms.

But the deployer determines where the system enters real operations.

The deployer determines whether the system influences a person's access to a job, an education opportunity, a medical pathway, a credit decision, a public service, a security assessment or a regulated financial process.

The deployer determines whether the output remains advisory or becomes operationally decisive.

The deployer determines whether human oversight is meaningful or symbolic.

The deployer determines whether the AI output is one input among many, or whether it effectively becomes the decision.

High-risk AI governance therefore begins with a contextual question:

What does this system affect when used here?

That question is essential.

Without it, an organization may underestimate risk because it focuses on the model rather than the deployment environment.

A high-risk AI system may look harmless when described technically, but become significant when mapped into an organizational workflow.

For example:

A model that classifies documents may become high-consequence if those documents affect asylum decisions, legal proceedings or public benefit eligibility.

A model that predicts behaviour may become high-consequence if used in employment, education, insurance, credit or policing.

A model that ranks people may become high-consequence if the ranking affects access, priority, opportunity or scrutiny.

A model that summarizes cases may become high-consequence if human reviewers rely on the summary without checking the underlying evidence.

A model that recommends action may become high-consequence if downstream systems treat the recommendation as a trigger.

This is the central lesson for deployers:

High-risk AI is not only a product category.

It is a deployment reality.

Article 26 becomes relevant because the deployer is the actor closest to that reality.

The deployer must therefore understand not only what the AI system is, but what it does inside the organization and what consequences attach to its output.

3.2 Annex III High-Risk Systems

Annex III of the EU AI Act identifies categories of AI systems that may be considered high-risk because of their use in sensitive or consequential domains.

These categories matter because they show the regulation's practical concern.

The concern is not AI in the abstract.

The concern is AI used in situations where errors, bias, opacity, overreliance, automation pressure, authority drift or poor oversight may affect people's rights, opportunities, safety or institutional treatment.

Annex III includes areas such as:

- biometrics;
- critical infrastructure;
- education and vocational training;
- employment, worker management and access to self-employment;
- access to and enjoyment of essential private services and essential public services and benefits;
- law enforcement;
- migration, asylum and border control management;
- administration of justice and democratic processes.

Each category introduces a different deployer context.

In biometrics, the deployer context may involve identification, verification, classification, access control, public security, workplace access, border management or law enforcement.

In critical infrastructure, the deployer context may involve transport, energy, water, digital infrastructure or other systems where operational failure may affect public safety or essential services.

In education, the deployer context may involve admissions, assessment, placement, progression, monitoring or evaluation of learners.

In employment, the deployer context may involve recruitment, selection, promotion, termination, task allocation, performance evaluation or worker monitoring.

In essential services, the deployer context may involve credit, insurance, public benefits, healthcare access, social support, housing or other services that affect people's life chances.

In law enforcement, the deployer context may involve risk assessment, evidence analysis, prioritization, investigation, profiling or operational decision support.

In migration and border control, the deployer context may involve eligibility, risk assessment, identity assessment, document analysis, border decisions or asylum processes.

In justice and democratic processes, the deployer context may involve legal research, case analysis, decision support, dispute resolution, electoral processes or democratic participation.

These categories are not merely legal labels.

They are signals of consequence.

For a deployer, each Annex III category should trigger a practical governance inquiry:

What exactly is the system doing?

Which process does it affect?

Which persons may be affected?

What decision or action may be influenced?

What authority is attached to the output?

What human oversight exists?

What input data is used?

What evidence supports the output?

What logs are retained?

What happens if the system is wrong?

What happens if the system is right but used in the wrong context?

What happens if the system is formally advisory but practically decisive?

This last question is especially important.

Many organizations may describe AI systems as decision-support tools.

But in practice, decision support can become decision shaping.

A score may become a default recommendation.

A ranking may become a shortlist.

A risk flag may become a trigger for deeper scrutiny.

A summary may become the main evidence reviewed by a human.

A recommendation may become the action that staff are expected to follow.

A workflow may remain formally human-led, while the AI output becomes the practical centre of gravity.

This is why Annex III deployer analysis should not stop at system classification.

It should examine actual operational influence.

The question is not only:

Is this system listed in an Annex III category?

The question is also:

How does this system shape decisions, actions, priorities, access or consequences inside this organization?

That is where Article 26 becomes operational.

3.3 AI Systems Embedded in Regulated Products

High-risk AI is not limited to standalone software used in organizational decision-making.

An AI system may also be high-risk where it is used as a safety component of a product, or is itself a product, covered by relevant Union harmonisation legislation, and where the product requires third-party conformity assessment.

This matters because some AI systems operate inside physical, technical or safety-critical environments.

They may be embedded in machinery, medical devices, transport systems, industrial systems, radio equipment, aviation systems, automotive systems, marine equipment or other regulated product environments.

In such cases, the deployer context may look different from the Annex III organizational decision-support context.

The risk may not arise primarily from a person being ranked, scored or assessed.

The risk may arise from the AI system affecting physical safety, system performance, operational reliability or technical control.

A deployer using an AI-enabled safety component must therefore understand not only the software output, but the product environment in which the AI system operates.

This includes questions such as:

What physical process does the AI system influence?

What safety function does it support?

What failure modes are possible?

What human intervention is available?

What alarms, overrides or fail-safe states exist?

What operational data is used?

What maintenance or calibration is required?

What logs are generated?

What downstream systems depend on the AI-enabled component?

What happens when the AI system behaves unexpectedly?

This is important because Article 26 deployer obligations may intersect with sector-specific safety obligations, product-safety requirements, technical maintenance duties, cybersecurity expectations and operational risk controls.

The deployer may not be designing the embedded AI system.

But the deployer may be responsible for using the product in real conditions.

This means the deployer should understand:

- the intended operating conditions;
- the boundaries of safe use;
- the limitations described by the provider or manufacturer;
- the training required for operators;
- the monitoring needed during operation;
- the escalation path for anomalies;
- the evidence needed if something goes wrong.

In embedded systems, human oversight may also look different.

It may not be a person reviewing an AI-generated recommendation on a screen.

It may be an operator able to override a machine.

It may be a technician monitoring system diagnostics.

It may be a clinician interpreting output from a medical device.

It may be a transport operator responding to an automated control signal.

It may be a safety officer reviewing logs after an anomaly.

This reinforces an important point:

Human oversight is not one thing.

It must be interpreted in context.

In a recruitment workflow, oversight may involve reviewing the basis of an AI-generated shortlist.

In a clinical workflow, oversight may involve assessing whether an AI-supported recommendation fits the patient's condition.

In an industrial environment, oversight may involve interrupting a physical process before unsafe operation occurs.

In a financial workflow, oversight may involve reviewing whether a model-generated risk assessment should influence a regulated decision.

In a public-sector workflow, oversight may involve ensuring that an AI-supported recommendation does not improperly affect access to rights or services.

The deployer must therefore translate Article 26 into the specific operational environment in which the high-risk system is used.

For embedded AI, this means connecting AI governance to safety engineering, product operation, maintenance, incident response and system assurance.

The governance question becomes:

Can the deployer prove that the AI-enabled product or safety component was used under valid operating conditions?

This includes evidence of:

- appropriate use;
- operator competence;
- maintenance or calibration where relevant;
- monitoring;
- incident response;
- log retention;
- escalation;
- continued safe operation.

The more closely the AI system is connected to physical safety, critical infrastructure or regulated products, the more important this evidence becomes.

3.4 Intended Purpose and Actual Use

One of the most important distinctions in deployer governance is the distinction between intended purpose and actual use.

The intended purpose is defined by the provider.

It describes the use for which the AI system is intended, including the specific context and conditions of use.

Actual use is determined by what the deployer does with the system in practice.

The gap between intended purpose and actual use is one of the most important risk surfaces in AI governance.

A system may be provided for one type of workflow but gradually used for another.

A system may be introduced as a decision-support tool but become a practical decision-making tool.

A system may be intended for internal prioritization but later used for external-facing decisions.

A system may be intended for low-risk classification but later used in a high-consequence domain.

A system may be intended to assist trained specialists but later be used by staff without the same competence.

A system may be intended to support review but later be used to reduce review.

A system may be intended to provide a warning signal but later be used as an automatic trigger.

These changes may not happen through a single formal decision.

They may happen gradually.

They may happen because the system is useful.

They may happen because staff trust it.

They may happen because management wants efficiency.

They may happen because the workflow is under pressure.

They may happen because integration makes automation easier than review.

They may happen because the output is available, formatted and convenient.

This is why actual use must be monitored.

The fact that a system was deployed under a valid intended purpose does not prove that it remains used within that purpose.

Article 26 deployer governance must therefore include intended-purpose discipline.

This means the deployer should be able to answer:

What did the provider say the system was for?

What internal process is the system actually used in?

Who uses it?

What decisions or actions does it influence?

Has use expanded since deployment?

Has the output become more authoritative?

Has the system been connected to downstream tools?

Has the system's practical function changed?

Are users relying on the system more heavily than expected?

Is the system used for persons, contexts or decisions beyond its original scope?

These are not minor administrative questions.

They are central to deployer responsibility.

A system used outside intended purpose may no longer be operating within the assumptions that supported its original risk assessment.

The provider's documentation may no longer fit the actual use.

The oversight measures may no longer be sufficient.

The input data may no longer be appropriate.

The monitoring approach may no longer detect relevant risks.

The logs may not capture the expanded consequence.

The organization may believe it is still operating the same system, while the system's role has changed.

This is a classic governance failure:

the form remains visible, but the conditions that made the form meaningful have changed.

The system still exists.

The policy still exists.

The approval still exists.

The workflow still runs.

But the deployment context has shifted.

This is why intended purpose must be treated as a living boundary.

It is not only a statement in documentation.

It is a control condition.

When actual use changes, continued validity should be reassessed.

The deployer should not ask only:

Did we approve this system?

The deployer should ask:

Does the way we are using this system still belong within the conditions under which it was approved?

This question is central to runtime governance.

3.5 Why Deployer Context Matters

The same AI system can produce different risk depending on deployer context.

This may be the most important practical point in Chapter 3.

An AI system does not operate in a vacuum.

It operates inside an organization.

It is surrounded by people, workflows, incentives, deadlines, data pipelines, permissions, technical integrations, policies, oversight structures and operational pressures.

That environment determines the system's practical significance.

A model output may be harmless in one context and highly consequential in another.

A recommendation may be optional in one workflow and effectively binding in another.

A score may be advisory in one organization and determinative in another.

A summary may be helpful in one team and dangerously incomplete in another.

A risk flag may trigger a manual review in one setting and automatic exclusion in another.

The deployer context determines how AI output becomes meaningful.

This has several implications.

First, deployers must map AI systems to workflows.

A system inventory is not enough.

Organizations need to know which process each system affects.

Second, deployers must map workflows to consequences.

A workflow is not only a sequence of steps.

It is a pathway through which decisions, rights, obligations, access, risk, money, safety, opportunity or institutional treatment may be affected.

Third, deployers must map consequences to controls.

The greater the consequence, the stronger the control should be.

Fourth, deployers must map controls to evidence.

A control that cannot be evidenced may be difficult to defend.

Fifth, deployers must map evidence to reassessment.

Evidence should not only explain what happened after the fact.

It should help the organization know when conditions have changed enough to question continuation.

This means deployer context is the bridge between classification and control.

A high-risk classification tells the organization that the system deserves serious governance attention.

But deployer context tells the organization what kind of governance is needed.

The governance needed for a hiring system may differ from the governance needed for a clinical decision-support system.

The governance needed for a public-benefit triage system may differ from the governance needed for a financial risk scoring system.

The governance needed for an AI-enabled medical device may differ from the governance needed for an AI agent connected to internal workflow tools.

Article 26 does not remove the need for contextual judgment.

It increases it.

The deployer must translate the obligation into the environment of use.

This translation should include:

- process mapping;
- role mapping;
- authority mapping;
- data-source mapping;
- oversight mapping;
- consequence mapping;
- log mapping;
- escalation mapping;
- boundary mapping;

- evidence mapping.

Without these maps, the deployer may know that a high-risk AI system is present, but not understand how risk becomes operational.

This is one of the most dangerous states for an organization.

The system is identified.

The policy exists.

The responsible team is named.

But the real pathway from AI output to consequence remains unclear.

In such an environment, governance may appear mature while operational control remains weak.

This is why deployer context matters.

It determines whether Article 26 becomes a meaningful operational discipline or a formal compliance exercise.

A mature deployer should be able to say:

We know what the system is.

We know why it is high-risk.

We know where it sits in the workflow.

We know who uses it.

We know what data it receives.

We know what output it produces.

We know what decisions it can influence.

We know who has oversight.

We know when escalation is required.

We know what evidence is retained.

We know when conditions have changed enough to reassess.

This is the level of clarity that Article 26 should push organizations toward.

Not simply awareness of AI systems.

Operational understanding of AI systems.

Not simply classification.

Contextual control.

Not simply compliance.

Evidence of continued validity.

Chapter 3 Summary

High-risk AI systems should not be understood only as technical systems.

They should be understood as systems operating inside consequential human, organizational, public, financial, safety, legal or institutional contexts.

The same AI capability may create different risk depending on where it is deployed, what process it affects, what data it uses, how much authority is attached to its output and what downstream consequences follow.

For deployers, this means classification is only the beginning.

The deeper task is contextual governance.

The deployer must understand the difference between intended purpose and actual use.

The deployer must identify whether use has drifted.

The deployer must understand whether an AI output remains advisory or has become practically decisive.

The deployer must connect high-risk classification to workflow mapping, consequence mapping, oversight mapping, evidence mapping and reassessment capacity.

This is why Article 26 cannot be applied properly without understanding deployer context.

The law may define the category.

But the deployment environment defines the control problem.

And the deployer is responsible for that environment.

Chapter 4

Use According to Instructions

Legal anchor and framework boundary. [Article 13](#); [Article 26\(1\)](#). The chapter translates instructions-for-use duties into deployer controls; suggested control objects are not statutory terms.

4.1 Provider Instructions as Operational Boundaries

Article 26 begins with a deceptively simple obligation:

Deployers of high-risk AI systems must take appropriate technical and organisational measures to ensure that they use those systems in accordance with the instructions for use accompanying the systems.

At first glance, this may appear to be a basic compliance requirement.

Read the instructions.

Follow the instructions.

Document that the instructions were followed.

But in high-risk AI governance, instructions for use are much more important than ordinary user guidance.

They are operational boundaries.

They describe the conditions under which the provider expects the system to be used.

They help define the intended purpose.

They describe limitations, assumptions, performance characteristics, expected input conditions, known risks, human oversight measures and proper use conditions.

They are the bridge between the provider's design assumptions and the deployer's real-world use.

For this reason, instructions for use should not be treated as documents that are simply stored during procurement.

They should be translated into operational controls.

A deployer should ask:

What does the provider say this system is intended to do?

What does the provider say this system should not be used for?

What types of input data are expected?

What are the known limitations?

What human oversight measures are required?

What performance assumptions apply?

What risks are identified?

What monitoring or logging capabilities are described?

What conditions should trigger caution, review or suspension?

What user competence is assumed?

What organizational measures are needed to make the instructions real?

The central governance issue is that provider instructions are often written as documentation, but deployment happens as workflow.

There is a translation step.

That translation step is where risk can enter.

A provider may state that a system is intended only to assist human review.

But the deployer may insert the output into a workflow where human review becomes superficial.

A provider may state that the system requires trained users.

But the deployer may allow staff to use it without sufficient training.

A provider may state that the system should not be used outside specific data conditions.

But the deployer may connect it to broader or lower-quality data sources.

A provider may state that the output should not be used as the sole basis for a decision.

But the deployer may design internal targets, dashboards or automation pathways that cause the output to become practically decisive.

A provider may state that certain limitations are known.

But the deployer may fail to communicate those limitations to actual users.

In such cases, the deployer may believe it is using the same system the provider supplied.

But operationally, the system has changed.

Not because the model changed.

Because the context changed.

The instructions for use are therefore not merely legal paperwork.

They are a boundary document.

They tell the deployer what conditions must remain true for use to remain within the provider's intended framework.

If those conditions are lost, ignored or not implemented, the deployer may no longer be operating within the original control assumptions.

This makes the first Article 26 obligation foundational.

Before human oversight, monitoring, logging or escalation can work properly, the deployer must understand what the system is supposed to be, how it is supposed to be used, and where the limits of appropriate use sit.

This is the first step in evidence of control.

4.2 The Problem of Instruction Drift

Instruction drift occurs when the actual use of an AI system gradually moves away from the provider's instructions for use.

This drift may be obvious.

For example, an organization may knowingly use a system for a purpose that the provider explicitly excluded.

But more often, instruction drift is subtle.

It happens through convenience.

It happens through pressure.

It happens through workflow integration.

It happens through user habit.

It happens through efficiency targets.

It happens because the system appears useful in adjacent contexts.

It happens because staff begin to trust the output more than originally intended.

It happens because a recommendation becomes easier to accept than to challenge.

It happens because the output is available, structured and fast.

The organization may not consciously decide to violate the instructions.

The use simply evolves.

This is especially likely with AI systems because their outputs can be flexible, reusable and easy to insert into different workflows.

A system intended for internal analysis may begin to influence external decisions.

A system intended for prioritization may become a ranking mechanism.

A system intended for triage may become a denial pathway.

A system intended to assist experts may begin to be used by non-experts.

A system intended for one population may be used on another.

A system intended for one data environment may be connected to different data sources.

A system intended for decision support may become a workflow trigger.

This is instruction drift.

It is dangerous because the organization may continue to believe that the system is being used within approved conditions.

The documentation remains the same.

The procurement record remains the same.

The policy remains the same.

The system name remains the same.

The approval may remain visible.

But the real use has shifted.

The system is now operating under conditions that may no longer support the original risk assessment, provider instructions or oversight model.

Instruction drift is one of the reasons why static compliance is not enough.

An organization may be compliant at deployment but drift later.

The system may be initially mapped to the correct process but then expand into adjacent processes.

The users may be trained at the start but new staff may join later without equivalent training.

The system may be deployed under one policy version but continue operating after the policy changes.

The provider instructions may be reviewed during procurement but not revisited when the workflow changes.

This is why instructions for use must be treated as living control references.

The deployer should not only ask:

Did we receive the instructions?

The deployer should ask:

Are we still using the system under the conditions described in the instructions?

This requires periodic reassessment.

It also requires operational monitoring.

A mature deployer should be able to detect signals of instruction drift, such as:

- new user groups using the system;
- new workflows connected to the system;
- new data sources added;
- output used in more consequential decisions;
- automation built around system output;
- human review reduced over time;
- known limitations no longer visible to users;
- system use expanding beyond original internal documentation;
- provider updates not reflected in internal controls;
- users treating the output as authoritative despite advisory status.

Instruction drift does not always look like a breach.

Sometimes it looks like successful adoption.

That is what makes it difficult.

A system that is widely adopted, trusted and embedded may appear mature.

But if adoption expands beyond the conditions under which the system was intended to operate, maturity may hide risk.

Article 26 requires deployers to prevent this.

Use according to instructions is not a one-time procurement step.

It is a continuing governance obligation.

4.3 User Training and Internal Deployment Discipline

Instructions for use do not implement themselves.

They must be translated into internal deployment discipline.

This means that the deployer must create practical conditions that make appropriate use likely, consistent and observable.

The first requirement is training.

Users need to understand what the system is for.

They need to understand what the system is not for.

They need to understand the limits of the output.

They need to understand when human judgment is required.

They need to understand when escalation is required.

They need to understand when use should stop.

They need to understand the difference between advisory output and authoritative decision-making.

Training should not be generic.

A general AI awareness module may be useful, but it is not enough for high-risk AI deployment.

Training should be specific to the system, workflow, user role and consequence level.

A person using an AI system in HR needs different guidance from a person using an AI system in financial compliance, clinical support, procurement, public services or critical infrastructure.

The second requirement is role clarity.

Who is allowed to use the system?

Who is allowed to interpret the output?

Who is allowed to override the output?

Who is responsible for escalation?

Who reviews logs?

Who monitors performance?

Who approves changes in use?

Who determines whether use remains within the provider's instructions?

Without role clarity, instructions become passive documents.

The third requirement is workflow design.

The system should be embedded in a workflow that supports correct use.

If the provider says the output should not be used as the sole basis for a decision, the workflow should make additional review visible and required.

If the provider says the system has limitations under certain data conditions, the workflow should include data checks.

If the provider says human oversight is required, the workflow should create a real review point before consequence attaches.

If the provider says the system is not intended for certain populations or contexts, the workflow should prevent or flag such use.

The fourth requirement is access control.

Not every user should have access to every high-risk AI system.

Access should reflect competence, role, training and operational need.

Access control is not only a cybersecurity measure.

It is a governance measure.

If a system is intended for trained specialists, access should not be open to general staff without controls.

If a system is intended for a specific process, access should not make uncontrolled repurposing easy.

If a system output can influence consequential decisions, access should be tied to accountability.

The fifth requirement is change control.

High-risk AI systems should not drift into new uses without review.

A deployer should have a process for approving:

- new use cases;
- new user groups;
- new data sources;
- new workflow integrations;
- new downstream systems;
- changes in decision influence;
- changes in oversight model;
- changes in automation level;
- changes in escalation pathways.

This does not mean organizations must prevent innovation.

It means changes in use should be visible.

A system may be suitable for a new workflow.

But suitability should be assessed.

The sixth requirement is user-facing guidance.

Instructions for use are often too long, technical or provider-focused for daily users.

The deployer should create internal guidance that translates provider instructions into practical operational rules.

For example:

- when to use the system;
- when not to use the system;
- what output means;
- what output does not mean;
- what must be checked;
- when to escalate;
- what to document;
- what to do when the system behaves unexpectedly;
- who to contact;
- how to record a concern.

This guidance should remain aligned with the provider's instructions.

It should also be kept up to date.

If provider instructions change, internal guidance should change.

If actual workflow changes, guidance should be reviewed.

If incidents occur, guidance should be updated.

Internal deployment discipline is therefore the organization's method for turning instructions into behaviour.

Without it, Article 26 compliance may exist only on paper.

With it, the deployer can begin to show that the system was used under intentional, controlled and documented conditions.

4.4 Evidence That Instructions Were Followed

The obligation to use a high-risk AI system according to instructions creates an evidence problem.

It is not enough for the deployer to say:

We followed the instructions.

The deployer should be able to show how.

This requires evidence.

Evidence that instructions were followed may include:

- provider instructions received and stored;
- version history of instructions;
- internal review of instructions;
- mapping of instructions to internal controls;
- system-specific user training records;
- access-control records;
- internal standard operating procedures;
- workflow design documents;
- configuration records;
- use-case approval records;
- user acknowledgement records;
- monitoring records;
- incident and escalation records;
- periodic reassessment records;
- audit logs showing actual use.

The most important point is that evidence should connect the instruction to the operational control.

For example:

If the provider instruction says that the system requires human review before a decision is made, the deployer should be able to show:

- where in the workflow review occurs;
- who performs the review;
- what information the reviewer receives;
- what authority the reviewer has;
- how disagreement is recorded;
- whether the decision can proceed without review;
- whether review actually occurred in relevant cases.

If the provider instruction says that the system should be used only with specific types of input data, the deployer should be able to show:

- what data sources are connected;
- how data suitability is checked;
- how data freshness is monitored;
- how invalid data conditions are flagged;
- whether users can bypass checks;
- how exceptions are handled.

If the provider instruction says that output should not be used as the sole basis for a consequential decision, the deployer should be able to show:

- what other evidence is required;
- how human judgment is documented;
- whether alternative evidence was reviewed;
- whether the AI output was overridden;
- whether the output was treated as advisory or determinative in practice.

If the provider instruction says that certain limitations apply, the deployer should be able to show:

- how those limitations were communicated to users;
- whether limitations are visible in the interface;
- whether training covers limitations;
- whether monitoring detects scenarios where limitations matter;
- whether escalation occurs when limitations are triggered.

This is evidence of instruction compliance.

The evidence should not only show that the deployer received the provider's documentation.

It should show that the deployer converted the documentation into operational controls.

This distinction is critical.

A document repository is not evidence of control by itself.

A training completion record is not evidence of correct use by itself.

A workflow diagram is not evidence that the workflow operated as designed.

A policy statement is not evidence that users followed the provider's instructions.

Evidence becomes useful when it shows the relationship between:

instruction,

control,

operation,

log,

review,

and outcome.

For example:

Provider instruction: human review required before decision.

Internal control: decision cannot be finalized until a named reviewer confirms review.

Runtime evidence: timestamped review record, reviewer identity, evidence reviewed, AI output, reviewer action, escalation if disagreement occurred.

This is stronger than simply saying:

Human review is required by policy.

The goal is to make instruction-following reconstructable.

If a regulator, auditor, board member or internal reviewer asks whether the deployer used the system according to instructions, the organization should be able to produce evidence that shows:

what instruction applied,

what internal control implemented it,
who was responsible,
when the control operated,
what happened,
and whether the workflow remained within the provider's intended use.
This is how Article 26 becomes evidence architecture.

4.5 When Internal Use Departs From Intended Purpose

The most difficult situation arises when internal use departs from intended purpose.

This may happen intentionally or unintentionally.

It may happen formally or informally.

It may happen quickly or gradually.

It may happen because the system works well.

It may happen because the organization wants more value from the tool.

It may happen because users discover new uses.

It may happen because business pressure rewards efficiency.

It may happen because no one owns the boundary.

The key point is that departure from intended purpose must be treated as a governance event.

It should not be invisible.

It should not be informal.

It should not be treated as ordinary adoption without reassessment.

When actual use begins to depart from intended purpose, the deployer should pause and ask:

Is this new use still covered by the provider's instructions?

Does the provider's risk assessment still apply?

Does the system remain within its high-risk classification assumptions?

Does human oversight remain sufficient?

Are the input data conditions still appropriate?

Are the users competent for this use?

Does the output now influence a different decision or consequence?

Does this new use require a separate assessment?

Does this use require provider consultation?

Does this use change the deployer's legal or operational responsibility?

Does this use create new affected persons?

Does this use require additional notification?

Does this use change the logs that must be retained?

Does this use require changes to monitoring?

Does this use require a fundamental rights impact assessment, data protection impact assessment or other internal review?

The answer may not always be that the new use is prohibited.

But it must be assessed.

The danger is silent expansion.

Silent expansion occurs when an AI system's use expands without a visible governance decision.

This can create several risks.

First, the system may be used in a context for which it was not validated.

Second, the provider's instructions may no longer be sufficient.

Third, the original human oversight model may no longer match the consequence level.

Fourth, logs may not capture the right information.

Fifth, users may misunderstand the output's reliability.

Sixth, affected persons may not receive appropriate information.

Seventh, regulators may view the deployer as having failed to control the use of a high-risk AI system.

Silent expansion is especially dangerous in agentic AI and workflow automation.

A system may begin as a support tool.

Then it becomes connected to a ticketing system.

Then its recommendations are used to generate action plans.

Then downstream automation treats those action plans as triggers.

Then the system becomes practically operational, even though it remains described as advisory.

The deployer may still believe it is using the system according to original instructions.

But the workflow has changed.

The consequence pathway has changed.

The authority of the output has changed.

At that point, continued validity should be questioned.

This is one of the central messages of this study:

Presence is not validity.

Continuity is not compliance.

Operational use must remain connected to the conditions that made the original use appropriate.

When those conditions change, the deployer must reassess.

A mature deployer should therefore maintain an intended-purpose control process.

This process should include:

initial intended-purpose mapping;

actual-use mapping;

periodic comparison between intended purpose and actual use;

change triggers;

approval requirements for new use cases;

provider consultation where needed;

risk reassessment;

oversight reassessment;

logging and monitoring updates;

affected-person or worker notification review;

documented decision on whether use may continue.

This does not need to be overly bureaucratic.

But it must be real.

The deployer should know when the system has crossed a boundary.

And when a boundary is crossed, the deployer should not allow inherited confidence to continue automatically.

The workflow should enter a reassessment state.

In practical terms, this may mean:

ALLOW continued use,

HOLD pending review,

ESCALATE to governance or legal/compliance review,

or REFUSE the new use until conditions are satisfied.

This is how use according to instructions becomes runtime governance.

The instruction is not merely read.

It becomes a live boundary.

The deployer's responsibility is to know whether that boundary still holds.

Chapter 4 Summary

Using a high-risk AI system according to instructions is not a simple administrative requirement.

It is a foundational deployer obligation.

Provider instructions define operational boundaries. They describe the assumptions, limits, intended purpose, oversight requirements, data expectations and use conditions under which the system is meant to operate.

The deployer must translate those instructions into internal controls.

This requires training, role clarity, access control, workflow design, change control, monitoring and evidence.

The most serious risk is instruction drift.

A system may remain formally the same while actual use gradually expands beyond the conditions described by the provider.

This may happen through convenience, trust, automation, workflow integration or organizational pressure.

For this reason, instructions for use must become living control references.

The deployer must be able to show:

- what instructions applied;
- how they were interpreted;
- how they were implemented;
- who was trained;
- how access was controlled;
- how workflow controls reflected the instructions;
- how actual use was monitored;
- how deviations were detected;
- how changes were reassessed;
- how evidence was retained.

The central question is not only:

Did the deployer receive the provider's instructions?

The central question is:

Did the deployer operate the system within the conditions that made the instructions meaningful?

If actual use departs from intended purpose, the deployer must treat that departure as a governance event.

The system should not silently inherit confidence from a use case that no longer applies.

Article 26 therefore makes instructions for use part of the deployer's control architecture.

Use according to instructions is not only about reading documentation.

It is about preserving continued validity between provider assumptions and operational reality.

Chapter 5

Human Oversight

Legal anchor and framework boundary. [Article 14](#); [Article 26\(2\)](#). Human-in-control is a NextOne Observatory term used to operationalise effective oversight; it is not a defined statutory term.

5.1 Human Oversight as a Deployer Obligation

Human oversight is one of the most important concepts in practical AI governance.

It is also one of the most easily misunderstood.

Many organizations will be tempted to treat human oversight as a formal requirement.

They may assign a person.

They may document a role.

They may add a review step.

They may create a policy stating that a human remains responsible.

They may describe the system as “human-in-the-loop.”

But formal human presence is not the same as meaningful human oversight.

Article 26 makes this distinction important because it does not merely require deployers to mention human oversight. It requires them to assign human oversight to natural persons who have the necessary competence, training, authority and support.

Those words matter.

Competence.

Training.

Authority.

Support.

Together, they show that human oversight is not decorative.

It is not a symbolic reassurance.

It is not a governance label.

It is an operational capability.

A human can only oversee a high-risk AI system if that human has enough understanding to interpret the system, enough training to recognize limitations, enough authority to intervene, and enough organizational support to act when intervention is necessary.

Without these conditions, human oversight may exist in form but not in substance.

A human may be present but unable to understand the output.

A human may understand the output but lack authority to challenge it.

A human may have authority but lack time.

A human may have time but lack access to relevant evidence.

A human may have access to evidence but work inside a culture where challenging the AI system is discouraged.

A human may be named as responsible but not practically able to stop the workflow.

In all of these cases, the organization may claim human oversight, but the oversight may not be real.

This is why human oversight must be treated as part of the deployer's control architecture.

For high-risk AI systems, the deployer must ask:

Who is the human overseer?

What exactly are they overseeing?

What information do they receive?

What competence do they have?

What training have they completed?

What authority do they hold?

Can they challenge the AI output?

Can they delay the workflow?

Can they escalate concerns?

Can they override the system?

Can they stop the process before consequence attaches?

Is their intervention recorded?

Is their disagreement respected?

Does the workflow make intervention realistic?

These questions move human oversight from policy language into operational reality.

The key point is simple:

Human oversight is not the presence of a human.

Human oversight is the presence of a capable human control function.

That function must be designed, supported, evidenced and tested.

5.2 Human-in-the-Loop Versus Human-in-Control

The phrase “human-in-the-loop” is widely used in AI governance.

It is useful, but it can also be misleading.

A human may be in the loop in a purely procedural sense.

A human may click approve.

A human may review a dashboard.

A human may receive a recommendation.

A human may be copied into a workflow.

A human may have theoretical authority to intervene.

But being in the loop does not necessarily mean being in control.

This distinction is central to deployer governance.

Human-in-the-loop describes placement.

Human-in-control describes capability.

A person can be placed inside a workflow without having meaningful control over that workflow.

For example:

A recruiter may receive an AI-generated candidate ranking but be under pressure to follow the ranking because the system is considered objective.

A loan officer may see an AI-generated risk score but lack access to the underlying factors needed to challenge it.

A clinician may receive an AI-supported recommendation but not have enough time to verify the evidence behind it.

A public-sector case worker may be able to override an AI-supported recommendation in theory, but face internal pressure not to deviate from system output.

A compliance analyst may review AI-generated alerts but not understand how the model prioritizes cases.

A procurement manager may approve an AI-generated supplier recommendation without knowing that the supplier data source has not refreshed.

In each example, the human is present.

But control may be weak.

This is why organizations must be careful not to overstate human oversight.

A system should not be described as meaningfully human-controlled unless the human can actually influence the outcome.

Human control requires several conditions.

First, the human must understand the task.

Second, the human must understand the AI system's role in the task.

Third, the human must understand the output.

Fourth, the human must understand the limitations of the output.

Fifth, the human must have access to relevant evidence.

Sixth, the human must have authority to act.

Seventh, the human must have enough time to act.

Eighth, the workflow must allow intervention before consequence attaches.

Ninth, the organization must support intervention rather than punish it.

Tenth, the system must record oversight actions in a way that can later be inspected.

If these conditions are missing, the organization may have human-in-the-loop design without human-in-control governance.

This distinction matters deeply for Article 26.

A deployer cannot rely on human oversight as a control unless the human can actually perform the control function.

This means deployers should move away from vague claims such as:

A human reviews the output.

A human is involved.

A human remains responsible.

A human can override the system.

These statements are not enough.

The stronger questions are:

What does the human review?

What evidence is available?

What does the human know about the system?

What action can the human take?

What happens if the human disagrees?

Can the workflow continue without human review?

Is the human given enough time?

Is intervention logged?

Are overrides monitored?

Does management support human challenge?

Can the human stop the workflow?

In high-risk AI systems, human oversight should not be a passive role.

It should be an active control state.

The practical goal is not only to put a human somewhere in the process.

The goal is to ensure that the system cannot move from AI-supported output to operational consequence unless human control conditions are satisfied where required.

That is the difference between human-in-the-loop and human-in-control.

5.3 Competence, Authority and Support for Oversight

Article 26's language around competence, training, authority and support is extremely important because it makes human oversight practical.

Each word represents a different failure mode.

Competence addresses whether the human can understand the system and the domain.

Training addresses whether the human has been prepared for the specific oversight task.

Authority addresses whether the human is permitted to act.

Support addresses whether the organization gives the human the tools, time, information and backing needed to act effectively.

A failure in any one of these can weaken oversight.

Competence is not generic intelligence.

A person may be highly capable in one role but not competent to oversee a specific high-risk AI system.

For example, a senior manager may have organizational authority but lack technical or domain understanding.

A technical specialist may understand the model but lack understanding of the regulated process.

A domain expert may understand the decision context but not understand the AI system's limitations.

A compliance officer may understand policy but not operational workflow.

A clinician may understand the patient domain but not know how to interpret the AI system's confidence, scope or failure modes.

Therefore, competence must be defined in relation to the system and the workflow.

A deployer should ask:

What does the overseer need to understand?

Do they understand the domain?

Do they understand the AI system's role?

Do they understand the provider's instructions?

Do they understand the system's limitations?

Do they understand when the output should not be trusted?

Do they understand escalation triggers?

Do they understand what evidence should be checked?

Do they understand the consequences of approving, rejecting or escalating the output?

Training is the next layer.

Competence may exist in principle, but training ensures alignment with the specific system and use case.

Training should not be limited to general AI awareness.

For high-risk AI systems, training should cover:

- intended purpose;
- instructions for use;
- limitations;
- expected input conditions;
- known risks;
- output interpretation;
- automation bias;
- escalation pathways;
- override procedures;
- log and evidence requirements;
- incident reporting;
- affected-person implications;
- internal accountability.

Training should also be updated when the system, workflow, data sources, provider instructions or regulatory expectations change.

Authority is the third layer.

A human cannot oversee a system if they cannot act.

Authority should include the power to:

- question the output;
- request additional evidence;
- delay the workflow;

- escalate the case;
- override the output;
- reject the recommendation;
- stop the process;
- report concerns;
- trigger review;
- require reassessment.

Authority must be more than theoretical.

If a human can override an AI output in theory but is punished, ignored or pressured not to do so, then authority is weak.

If a human can escalate in theory but escalation is slow, unclear or culturally discouraged, then authority is weak.

If a human can stop the workflow in theory but the system is designed to proceed automatically unless they act instantly, then authority is weak.

Authority must be operationally usable.

Support is the fourth layer.

Support includes:

- time;
- tools;
- access to underlying evidence;
- access to provider information;
- access to domain experts;
- access to technical support;
- escalation channels;
- management backing;
- clear procedures;
- legal/compliance support;
- interface design that makes oversight possible.

A human reviewer who is expected to oversee hundreds of AI-supported outputs per day may not have meaningful support.

A human reviewer who sees only the final score but not the underlying evidence may not have meaningful support.

A human reviewer who must approve within seconds may not have meaningful support.

A human reviewer who lacks the ability to compare the AI output with source data may not have meaningful support.

A human reviewer who is never told when the system's data source is stale may not have meaningful support.

Support is therefore not optional.

It is the condition that makes competence, training and authority usable.

For Article 26, deployers should therefore treat human oversight assignment as an evidence requirement.

The organization should be able to show:

- who was assigned oversight;
- why they were competent;
- what training they received;
- what authority they had;
- what support was provided;
- what information they could access;
- what actions they could take;
- whether they actually intervened;
- whether their interventions were respected;
- whether oversight remained effective under operational pressure.

This is the evidence of human oversight.

Without it, human oversight may become a claim rather than a control.

5.4 Oversight Under Operational Pressure

Human oversight often looks stronger in policy than it does under operational pressure.

This is one of the central risks in AI deployment.

A policy may describe careful review.

A governance framework may assign responsibility.

A workflow diagram may show a human decision point.

A training document may explain escalation.

But the real question is:

What happens during actual operation?

What happens when there are too many cases?

What happens when the AI output is usually right?

What happens when staff trust the system more than themselves?

What happens when management wants faster decisions?

What happens when the system output is presented with confidence?

What happens when the interface makes approval easier than challenge?

What happens when the human reviewer lacks time?

What happens when the cost of disagreement is high?

What happens when the workflow is designed around throughput?

This is where human oversight can degrade.

The formal control remains visible, but the practical capacity to exercise the control narrows.

This is not always intentional.

It may happen gradually.

At first, reviewers carefully examine AI outputs.

Then they begin to accept outputs more often.

Then they trust the system because it appears reliable.

Then they review only unusual cases.

Then they become uncomfortable challenging the system.

Then the organization begins to treat override as an exception.

Then the human becomes a confirmation layer rather than an oversight layer.

This is automation bias.

It is also governance drift.

The system still has a human in the loop.

But the human's role has changed.

The human no longer actively controls the workflow.

The human legitimizes the workflow.

This is dangerous in high-risk AI systems because it creates the appearance of human oversight without the substance of human control.

Operational pressure can weaken oversight in several ways.

First, time pressure.

If a human reviewer does not have enough time, review becomes superficial.

Second, volume pressure.

If too many outputs require review, the human may rely on shortcuts.

Third, confidence pressure.

If the system presents outputs confidently, the human may assume correctness.

Fourth, authority pressure.

If management treats the AI system as more objective than human judgment, disagreement becomes difficult.

Fifth, interface pressure.

If the system makes approval easy and challenge difficult, the human is nudged toward acceptance.

Sixth, evidence pressure.

If the human cannot access the evidence behind the output, review becomes symbolic.

Seventh, cultural pressure.

If overrides are viewed as inefficiency, oversight becomes performative.

A mature deployer should therefore test human oversight under real conditions.

It is not enough to ask whether the oversight role exists.

The organization should ask:

Can the oversight role function under load?

Can reviewers challenge the system without penalty?

Do users understand when to distrust the output?

Do reviewers have enough time?

Are overrides monitored positively or negatively?

Are escalations treated as control signals or operational friction?

Does the interface support critical review?

Are humans shown uncertainty, limitations and evidence?

Are reviewers informed when input data may be stale?

Are reviewers alerted when system use drifts beyond intended purpose?

Does the workflow pause when required oversight is missing?

These questions reveal whether oversight is resilient.

Human oversight should also be monitored.

The deployer should collect evidence such as:

- review rates;
- override rates;
- escalation rates;
- time spent on review;
- cases approved without meaningful review;
- patterns of overreliance;
- reviewer feedback;
- false acceptance patterns;
- instances where humans lacked information;
- situations where intervention occurred too late;
- pressure points in the workflow.

This type of monitoring helps detect oversight degradation.

Oversight degradation is the gradual loss of meaningful human control while the formal oversight structure remains in place.

It is one of the most important risks in practical AI governance.

For high-risk AI systems, the deployer should not only assign human oversight.

The deployer should maintain it.

Human oversight is not a one-time role assignment.

It is a continuing operational condition.

5.5 Evidence That Intervention Was Possible

The strongest test of human oversight is not whether a human was present.

The strongest test is whether intervention was possible.

A deployer should be able to show that a human could intervene before the AI-supported workflow created consequence.

This is the practical meaning of human control.

If a human can only review after the decision has already taken effect, then oversight may be retrospective rather than preventive.

If a human can only complain after the system has acted, then oversight may be weak.

If a human can see the output but cannot stop the downstream process, then oversight may be incomplete.

If a human can override in theory but the system does not allow override in practice, then oversight is not meaningful.

If a human can intervene only after harm occurs, then the oversight may fail its protective purpose.

For Article 26, the deployer should therefore preserve evidence that intervention was possible.

This evidence may include:

- workflow records showing review before consequence;
- system design showing pause points;
- access records showing reviewer involvement;
- interface records showing available actions;
- timestamps showing review occurred before decision finalization;
- logs of overrides, escalations and holds;
- records of cases where the AI output was rejected;
- records of cases where the workflow was paused;
- records of cases where human review changed the outcome;
- evidence that users were trained to intervene;
- evidence that management supported intervention;
- evidence that escalation pathways were active.

The key question is:

Could the human change what happened?

If the answer is no, the oversight may be formal but not operational.

This is especially important when AI outputs feed into downstream systems.

For example:

An AI system may produce a recommendation.

That recommendation may enter a case-management system.

The case-management system may generate a task.

The task may trigger a notification.

The notification may trigger a customer-facing action.

The customer-facing action may create consequence.

If the human review occurs only at the beginning, but downstream automation changes the effect of the output later, oversight may not cover the real consequence point.

The deployer must therefore understand the whole workflow.

Intervention must be possible where consequence attaches.

This may require multiple oversight points.

It may require a control gate before final action.

It may require escalation when the output moves from advisory to operational.

It may require human review before external communication.

It may require approval before system write-back.

It may require pause conditions when authority, evidence or context changes.

This is why human oversight is closely connected to runtime governance.

A human must not only be assigned.

The system must know when human intervention is required.

For example, intervention may be required when:

- the AI output affects a high-consequence decision;
- input data is stale;
- confidence is low;
- the system detects uncertainty;

- the output conflicts with other evidence;
- the use case is outside normal scope;
- the affected person belongs to a sensitive category;
- the decision has legal or significant effect;
- policy context has changed;
- the system has been updated;
- model fallback has occurred;
- the output will trigger downstream automation;
- the human overseer is unavailable;
- the system cannot explain the basis of the output sufficiently.

In these situations, the correct governance response may not be to continue.

It may be to hold, escalate or refuse.

This turns human oversight into a control state.

The workflow should be able to say:

Human oversight satisfied: continue.

Human oversight required: hold.

Human oversight unavailable: escalate.

Human oversight insufficient: refuse continuation.

Human oversight conflicted: require additional review.

Human oversight overridden: log and justify.

This is much stronger than a generic human-in-the-loop claim.

It creates evidence that the organization did not simply assume oversight existed.

It shows that oversight operated as a condition for continuation.

This is the core of evidence of human control.

A deployer should be able to prove not only that a human was involved, but that the human could affect the outcome at the right time.

If this cannot be shown, the organization should be cautious about relying on human oversight as a compliance control.

Chapter 5 Summary

Human oversight is one of the most important deployer obligations under Article 26.

It is also one of the most vulnerable to misunderstanding.

Human oversight is not satisfied by simply naming a person, adding a review step, or describing the system as human-in-the-loop.

Meaningful oversight requires competence, training, authority and support.

A human must understand the system, the domain, the output, the limitations, the evidence and the consequences.

A human must have the authority to challenge, delay, escalate, override or stop the workflow.

A human must have the organizational support, time, tools and information needed to exercise that authority.

This chapter distinguishes between human-in-the-loop and human-in-control.

Human-in-the-loop describes placement.

Human-in-control describes capability.

For high-risk AI systems, the deployer's responsibility is not merely to place a human somewhere in the process.

The deployer must ensure that human oversight can function as a real control.

The greatest risks arise under operational pressure.

Volume, time pressure, automation bias, interface design, management expectations and lack of evidence can turn human oversight into a symbolic approval layer.

For this reason, deployers should monitor oversight itself.

They should examine review quality, override rates, escalation patterns, time pressure, reviewer feedback and evidence access.

Most importantly, deployers should preserve evidence that intervention was possible before consequence attached.

The central question is:

Could the human change what happened?

If the answer cannot be shown, human oversight may not be meaningful.

Article 26 therefore pushes deployers toward a stronger model of oversight:

not human presence,
but human control;
not symbolic review,
but practical intervention;
not policy language,
but evidence that oversight operated when it mattered.

Chapter 6

Input Data Relevance and Representativeness

Legal anchor and framework boundary. [Article 26\(4\)](#). The statutory duty applies to the extent the deployer controls input data; freshness and contextual-validity controls are operational recommendations.

6.1 Input Data as a Deployer Responsibility

Input data is one of the most important control points in high-risk AI deployment.

An AI system does not operate only on its model, architecture, provider documentation or technical design.

It operates on the data, information, prompts, records, files, cases, signals, observations and contextual inputs supplied to it during use.

This makes input data a practical governance issue.

Article 26 recognizes this by placing a specific obligation on deployers where they exercise control over input data. In such cases, deployers must ensure that the input data is relevant and sufficiently representative in view of the intended purpose of the high-risk AI system.

This obligation matters because the deployer may not control how the system was trained, validated or tested by the provider.

But the deployer may control what the system receives during operation.

That operational input may determine whether the system's output is meaningful, misleading, incomplete, biased, stale or outside the conditions originally assumed.

A high-risk AI system may be designed properly by the provider and still produce poor or unsafe outputs if the deployer supplies unsuitable input data.

The provider may define an intended purpose.

The provider may describe the expected input conditions.

The provider may explain limitations.

The provider may provide instructions for use.

But the deployer often controls the real data environment.

The deployer decides which internal records are connected.

The deployer decides which data sources feed the workflow.

The deployer decides whether data is current.

The deployer decides whether missing data is allowed.

The deployer decides whether users can enter free-text context.

The deployer decides whether downstream systems provide additional signals.

The deployer decides whether the system sees only a narrow case record or a broader decision environment.

This makes input data part of the deployer's control architecture.

The central question is not only:

Is the AI system capable?

The question is:

Was the system given input data that could support a valid output for the intended purpose?

A strong model with poor input data may produce a confident but unreliable result.

A compliant system with stale input data may produce an output that appears legitimate but no longer reflects the current situation.

A high-risk AI system with incomplete input context may make an output appear valid while ignoring relevant constraints.

A system used with data outside its intended scope may create a governance risk even if the model itself performs as designed.

This is why input data must be treated as evidence, not only as fuel.

The deployer should be able to show:

what data was supplied,

where it came from,

whether it was current,

whether it was relevant,

whether it was sufficiently representative,

whether it matched the intended purpose,

whether limitations were known,

whether missing values were detected,

whether human reviewers understood the data condition,
and whether the output was allowed to proceed despite any data limitations.

For high-risk AI systems, input data is not merely a technical input.

It is a condition of admissibility.

If the input data does not support the intended use, the workflow should not silently continue with inherited confidence.

The system may need to hold, escalate or refuse continuation.

This is the core practical meaning of Article 26(4).

The deployer's obligation is not only to provide data.

It is to provide data that supports valid use.

6.2 Intended Purpose and Data Suitability

Input data relevance cannot be assessed in isolation.

Data is not relevant in the abstract.

Data is relevant in relation to a purpose.

This is why the intended purpose of the high-risk AI system is central.

A dataset, case record or user-provided input may be suitable for one purpose and unsuitable for another.

For example:

A short customer support history may be suitable for drafting a basic response, but not suitable for deciding whether a fraud flag should be applied.

A résumé may be suitable for generating interview questions, but not sufficient for making a hiring recommendation.

A clinical summary may be useful for triage support, but not sufficient for diagnosis without current test results and clinician review.

A supplier profile may be suitable for initial procurement screening, but not sufficient for high-value supplier approval if financial status, sanctions status or performance history is stale.

A student performance record may be useful for learning support, but not sufficient for consequential educational placement without broader context.

A public-service case record may support administrative review, but not justify an automated or AI-supported decision affecting access to benefits without complete and current evidence.

The same input data can therefore be appropriate or inappropriate depending on how the AI system is used.

This means deployers must connect input data controls to intended purpose.

A deployer should ask:

What is the system intended to do?

What input data does the provider expect?

What input data is required for this use case?

What data is actually supplied?

Is the data current enough?

Is the data complete enough?

Is the data representative enough?

Is the data source reliable enough?

Are there known gaps?

Are there known biases?

Does the data describe the affected person, case, environment or process accurately?

Does the data support the level of consequence attached to the output?

If the system output will be used in a high-consequence decision, the input data standard should be higher.

This is a practical proportionality principle.

A low-consequence internal suggestion may tolerate more uncertainty.

A high-risk decision affecting employment, healthcare, finance, public services, education, justice or safety should not.

The deployer should therefore define input data suitability thresholds.

These thresholds may include:

- required data fields;
- accepted data sources;

- freshness requirements;
- completeness requirements;
- representativeness requirements;
- missing-data rules;
- outlier detection;
- bias checks;
- human verification requirements;
- escalation triggers;
- refusal conditions.

For example, a procurement AI system may require supplier financial data to be no older than a defined number of days before a recommendation can influence approval.

A HR AI system may require that candidate records exclude protected characteristics unless legally justified and properly controlled.

A clinical support system may require that patient data comes from validated records and that recent clinical changes are flagged.

A public-sector decision-support system may require that all legally relevant case facts are present before an AI-supported recommendation is shown.

The key point is that input data suitability should be explicit.

If suitability is implicit, users may assume the system has enough information because the system produces an answer.

This is dangerous.

AI systems often produce fluent, structured and confident outputs even when input data is incomplete or unsuitable.

The presence of an answer does not prove the adequacy of the input.

For this reason, deployers should design workflows that make input data conditions visible.

The user should be able to see whether required inputs are present.

The system should indicate when input data is stale, incomplete or outside expected scope.

The workflow should prevent high-consequence continuation when essential input conditions are not met.

The evidence record should preserve input data status.

This is how intended purpose becomes operational.

The system is not used appropriately merely because the output appears useful.

It is used appropriately only if the input data supports the purpose for which the system is being used.

6.3 Relevance, Representativeness and Operational Context

Article 26 refers to input data that is relevant and sufficiently representative.

These two concepts are related, but they are not the same.

Relevance asks whether the data matters for the intended purpose.

Representativeness asks whether the data adequately reflects the persons, cases, circumstances or conditions to which the system is being applied.

A dataset may be relevant but not representative.

For example, employment history is relevant to recruitment, but if the data reflects only a narrow type of candidate background, it may not be sufficiently representative.

A dataset may be representative in a general sense but not relevant to the specific decision.

For example, broad demographic information may describe a population, but may be irrelevant or inappropriate for a particular assessment.

A dataset may be relevant and representative at one point in time but become less so as conditions change.

For example, market data used in financial risk assessment may become stale during volatility.

A dataset may be relevant for one location, population or process but not another.

For example, a model used with data from one jurisdiction, institution or operational environment may not transfer cleanly to another.

This is why deployers must assess relevance and representativeness in context.

Operational context includes:

- the affected persons or groups;
- the decision or workflow being supported;

- the source and quality of the data;
- the time at which the data was collected;
- the environment in which the system is used;
- the assumptions made by the provider;
- the organization's actual use case;
- the level of consequence attached to the output.

A deployer should not ask only:

Do we have data?

The deployer should ask:

Is this the right data for this purpose, in this context, for this population, at this time?

This question is especially important in high-risk settings.

In employment, representativeness may concern whether input data fairly reflects relevant candidate or worker characteristics without reinforcing historical exclusion.

In education, representativeness may concern whether student data reflects the learning context adequately, rather than reducing students to narrow performance indicators.

In healthcare, representativeness may concern whether patient data reflects relevant clinical factors, population differences, and current condition.

In financial services, representativeness may concern whether risk data reflects actual exposure, market conditions, customer circumstances or current regulatory requirements.

In public services, representativeness may concern whether case data reflects the full legal and factual context of the affected person.

In critical infrastructure, representativeness may concern whether sensor or operational data reflects real operating conditions.

This is not only a fairness issue.

It is also a validity issue.

If input data is not representative, the system may produce outputs that appear structured but do not properly fit the situation.

The deployer should therefore treat data representativeness as a control condition.

This may require:

- data source review;

- population coverage assessment;
- case completeness checks;
- missing-data analysis;
- bias and proxy analysis;
- temporal relevance checks;
- domain expert review;
- exception handling;
- monitoring of output patterns;
- affected-group impact review where relevant.

The key governance challenge is that users may not see representativeness problems at the point of use.

The interface may show the output, not the data condition.

The system may present a score, summary or recommendation without exposing what is missing or underrepresented.

This creates a risk of false confidence.

The deployer should therefore build visibility into the workflow.

For example:

A reviewer should know whether key fields are missing.

A decision-maker should know whether data is outside expected population scope.

A compliance officer should know whether the data source has changed.

A clinician should know whether recent patient information is absent.

A procurement manager should know whether supplier risk data is outdated.

A public-sector official should know whether legally relevant facts are incomplete.

This visibility is part of evidence of control.

The organization must not only use relevant and representative data.

It must be able to show that it assessed and monitored the data conditions that made the output meaningful.

6.4 Data Freshness and Stale Evidence

Data freshness is one of the most important practical issues in AI deployment.

A high-risk AI system may produce a plausible output based on data that was accurate yesterday, last week or last month, but no longer valid today.

This is not always a model failure.

It may be a freshness failure.

The model may perform exactly as designed.

The workflow may complete successfully.

The logs may show normal operation.

The human may approve the output.

But the evidence used by the system may no longer support continuation.

This is why stale data can create silent failure.

Stale evidence does not always produce an error message.

It may produce a confident output.

It may produce a reasonable-looking recommendation.

It may produce a score that appears precise.

It may produce a summary that appears complete.

But the decision environment has degraded.

Examples are easy to imagine.

A procurement AI system recommends a supplier based on a profile that has not refreshed since before a financial downgrade.

A customer support AI system drafts a refund decision without seeing a fraud flag added after the context was retrieved.

A HR AI system evaluates internal mobility based on performance data that does not include recent manager changes or updated role requirements.

A financial services AI system uses risk thresholds that were valid before a policy update.

A public-service AI system processes a case without updated eligibility information.

A clinical AI system summarizes a patient record without the most recent lab result.

A critical infrastructure AI system responds to sensor data without knowing that a sensor has degraded or gone offline.

In each case, the AI system may not “fail” in a conventional technical sense.

It may simply operate on data that no longer supports the level of authority attached to the output.

This is why data freshness should be treated as a runtime control condition.

Deployers should define freshness expectations for different types of input data.

Not all data needs the same freshness standard.

Some data changes slowly.

Some data changes rapidly.

Some data becomes invalid immediately after a trigger event.

Some data remains useful for background context but not for consequential decisions.

A mature deployer should define freshness rules based on risk and purpose.

For example:

- real-time operational data may require immediate freshness;
- financial risk data may require daily or event-triggered refresh;
- regulatory policy data may require version tracking;
- HR data may require review when role, manager or employment status changes;
- clinical data may require current episode-level validation;
- supplier data may require freshness checks before approval;
- public-service case data may require confirmation before eligibility decisions.

Freshness should not be defined only by time.

It should also be event-based.

A data source may become stale because:

- a policy changed;
- a person’s status changed;
- a supplier’s risk changed;
- a case was updated;
- a new document was added;
- a system integration failed;
- a model was updated;
- a downstream process changed;

- a human approver changed role;
- a risk threshold changed;
- a regulator issued new guidance;
- an incident occurred.

This means freshness monitoring should include both temporal and contextual signals.

The deployer should ask:

When was the data last refreshed?

What event could make this data invalid?

Has such an event occurred?

Does the AI system know?

Does the human reviewer know?

Does the log record the data state?

Should the workflow continue?

Should the workflow hold pending refresh?

Should escalation be triggered?

Freshness is therefore closely linked to admissibility.

If the evidence is stale, the output may not be admissible for the intended consequence.

The correct response may be HOLD.

Not because the AI system crashed.

Not because the output is impossible.

But because the evidence no longer supports continuation.

This is one of the clearest examples of why deployer governance must become runtime governance.

The question is not only:

Did the system have data?

The question is:

Was the data still fresh enough to justify the use being made of the output?

6.5 Evidence That Input Conditions Were Valid

If input data relevance and representativeness are deployer obligations, then the deployer must be able to produce evidence that input conditions were valid.

This is the evidence challenge.

The deployer should not only assert that appropriate input data was used.

The deployer should be able to show it.

Evidence of valid input conditions may include:

- data source records;
- data lineage records;
- data freshness timestamps;
- data quality checks;
- completeness checks;
- missing-field reports;
- representativeness assessments;
- bias or proxy checks;
- data validation logs;
- user-entered input records;
- system context snapshots;
- versioned policy references;
- human verification records;
- exception records;
- escalation records;
- audit trails connecting input data to output.

The purpose of this evidence is not to create unnecessary bureaucracy.

The purpose is reconstructability.

If an AI-supported workflow later comes under review, the organization should be able to reconstruct:

what input data was used,

where it came from,

why it was considered relevant,

whether it was sufficiently representative,

whether it was fresh,
whether any required fields were missing,
whether any warnings were present,
whether the human reviewer saw the data condition,
whether the output was allowed to proceed,
and why continuation was considered acceptable.

Without this evidence, the organization may be unable to prove that the system was used under valid input conditions.

This matters because many AI failures are not caused by the model alone.

They are caused by poor data context.

The model may have generated a plausible output from incomplete input.

The system may have classified a case based on stale records.

The agent may have taken action based on outdated policy.

The workflow may have proceeded without required contextual evidence.

The human reviewer may have approved without knowing that data was missing.

In such cases, the key question becomes:

Did the deployer know the input condition?

If the answer is no, then the deployer's control environment may be weak.

A mature deployer should therefore create input condition records.

These records do not need to expose every underlying data item in every case.

But they should preserve enough information to understand the quality and status of the input environment.

For example, an input condition record may include:

- system identity;
- workflow identity;
- intended purpose;
- input data sources;
- source version;
- last refresh time;

- completeness status;
- missing required fields;
- representativeness status;
- known limitations;
- user-entered context;
- data validation result;
- human verification result;
- escalation status;
- decision to continue, hold or escalate.

This type of record transforms input data from invisible background into inspectable governance evidence.

It also supports human oversight.

A human reviewer cannot meaningfully oversee an output if they do not know whether the input data was valid.

The reviewer should be able to see:

This output is based on current data.

This output is based on incomplete data.

This output lacks required fields.

This output uses a data source outside normal scope.

This output is based on stale evidence.

This output should not be used without additional review.

This connects input data governance directly to human oversight.

It also connects input data governance to monitoring.

If input data quality begins to degrade across multiple cases, the organization should detect it.

If a data source stops refreshing, the organization should know.

If missing fields increase, the organization should investigate.

If a workflow repeatedly proceeds despite input warnings, the organization should review the control design.

If user-entered context frequently overrides structured data, the organization should assess why.

Input data governance is therefore not a one-time validation task.

It is a continuing operational discipline.

The deployer must maintain enough visibility to know whether the system is receiving inputs that can support the outputs being used.

This is why Article 26(4) belongs at the centre of deployer governance.

A high-risk AI system cannot be governed only at the model layer.

It must also be governed at the input layer.

The system's output is only as trustworthy as the conditions that produced it.

Chapter 6 Summary

Input data is not merely technical fuel for an AI system.

For high-risk AI systems, input data is a control condition.

Where the deployer controls input data, Article 26 requires the deployer to ensure that the data is relevant and sufficiently representative in view of the intended purpose.

This obligation matters because the deployer often controls the operational data environment.

A provider may design a system properly, but the deployer may still create risk by supplying stale, incomplete, irrelevant or insufficiently representative data.

The key concepts are relevance, representativeness and freshness.

Relevance means the data supports the intended purpose.

Representativeness means the data adequately reflects the persons, cases, circumstances or operating conditions to which the system is applied.

Freshness means the data is current enough for the consequence attached to the output.

The deployer should therefore define input data suitability thresholds.

These may include required fields, accepted sources, freshness rules, completeness checks, representativeness assessments, missing-data rules, bias checks, escalation triggers and refusal conditions.

For high-consequence workflows, the standard should be higher.

The central governance risk is that an AI system may produce a fluent, confident or structured output even when the input data does not support the decision environment.

This creates silent failure.

The system may complete the workflow successfully while the evidence behind the workflow has already degraded.

For this reason, deployers should preserve evidence that input conditions were valid.

This may include data lineage, freshness timestamps, completeness checks, source records, validation logs, user-entered context, representativeness assessments, human verification and escalation records.

The core question is:

Was the AI system given input data that could support a valid output for the intended purpose at the moment of use?

If the answer cannot be shown, the workflow may not be sufficiently admissible to continue.

Article 26(4) therefore pushes deployers toward input data governance as part of runtime governance.

Not only data in the database.

Data as evidence.

Not only data availability.

Data validity.

Not only output confidence.

Input condition proof.

Chapter 7

Monitoring the Operation of High-Risk AI Systems

Legal anchor and framework boundary. [Article 26\(5\)](#); [Article 72](#). The Regulation requires monitoring based on instructions and action on identified risk; governance-signal monitoring is an operational extension.

7.1 Monitoring as a Continuing Obligation

Monitoring is one of the most important practical obligations for deployers of high-risk AI systems.

It is also one of the most misunderstood.

Many organizations think of monitoring as a technical activity.

They monitor uptime.

They monitor latency.

They monitor system errors.

They monitor API failures.

They monitor infrastructure cost.

They monitor user activity.

These forms of monitoring are useful, but they are not enough for high-risk AI governance.

For high-risk AI systems, monitoring must ask a deeper question:

Is the system still operating under valid control conditions?

This is not only a technical question.

It is an operational governance question.

A high-risk AI system may be online, fast, stable and technically functional while the conditions that make its use valid have changed.

The system may not crash.

The workflow may complete.

The dashboard may remain green.

The log may record normal activity.

But the decision environment may no longer support continuation.

This is why monitoring must be understood as a continuing deployer obligation.

At deployment, the organization may have reviewed the provider's instructions, assigned human oversight, mapped the intended purpose, assessed input data, configured logging and created an internal policy.

But those conditions may not remain stable.

The system may be used by new teams.

The workflow may expand.

The input data may change.

The output may become more influential.

The human overseer may change role.

The downstream process may be automated.

The policy environment may change.

The provider may update the system.

The model may be substituted.

The risk threshold may change.

The business process may become more dependent on the AI output.

Monitoring is the mechanism that allows the deployer to notice when these changes matter.

Without monitoring, the deployer may possess a compliant deployment record but lack awareness of current operational reality.

This creates a dangerous gap.

The organization may believe the system remains under control because the original approval remains visible.

But approval is not permanent evidence of continued validity.

Monitoring must therefore track not only whether the AI system is functioning, but whether the deployment conditions remain aligned with the original control assumptions.

This requires monitoring at several levels:

- technical monitoring;
- performance monitoring;
- input data monitoring;
- workflow monitoring;
- oversight monitoring;
- authority monitoring;
- boundary monitoring;
- risk monitoring;
- incident monitoring;
- evidence monitoring.

Technical monitoring asks whether the system is available and functioning.

Performance monitoring asks whether the system continues to perform within expected parameters.

Input data monitoring asks whether the system is receiving relevant, representative and fresh data.

Workflow monitoring asks whether the system is still used in the intended process.

Oversight monitoring asks whether human oversight remains available and meaningful.

Authority monitoring asks whether the output is gaining more practical authority than originally intended.

Boundary monitoring asks whether actual use remains within the provider's instructions and intended purpose.

Risk monitoring asks whether the risk environment has changed.

Incident monitoring asks whether harmful or potentially harmful events have occurred.

Evidence monitoring asks whether the organization is preserving enough information to reconstruct use and control.

Together, these form the deployer's runtime monitoring environment.

This is why monitoring should not be treated as a passive dashboard.

Monitoring is an active governance function.

It helps determine whether the workflow should continue, hold, escalate or stop.

7.2 Detecting Malfunction and Performance Change

One obvious purpose of monitoring is to detect malfunction.

A high-risk AI system may malfunction in ways that are visible:

The system may fail to respond.

The system may return errors.

The system may produce outputs outside expected format.

The system may generate inconsistent results.

The system may become unavailable.

The system may fail to process required inputs.

The system may behave differently after an update.

These failures are important and should be detected.

But AI system malfunction may also be less visible.

A system may continue producing outputs, but the outputs may become less reliable.

A model may degrade under new input patterns.

A system may perform worse for certain groups.

A system may become less accurate when data distributions shift.

A system may behave unpredictably after integration changes.

A model may begin producing outputs that appear plausible but are less grounded.

A decision-support system may become less useful under changed business conditions.

A generative component may produce confident but unsupported reasoning.

An agentic system may select tools in unexpected ways.

This is why monitoring must include performance change, not only failure.

Performance change may be gradual.

It may not trigger a system error.

It may not appear in uptime monitoring.

It may not be obvious to individual users.

It may only become visible through pattern analysis.

For example:

A recruitment AI system may begin recommending a narrower candidate group after changes in input data.

A financial risk tool may begin producing more false positives after a market shift.

A clinical support system may show reduced reliability for a certain patient group.

A procurement recommendation system may begin over-relying on outdated supplier scores.

A public-service triage tool may begin producing different outcomes after a policy update.

A fraud detection system may degrade when adversarial behaviour changes.

A workflow agent may increase tool calls because context management has degraded.

These changes are not always simple malfunctions.

They are operational signals.

The deployer should define what performance changes matter for the use case.

This may include:

- accuracy change;
- error-rate change;
- false positive rate;
- false negative rate;
- group-level performance differences;
- confidence calibration;
- override rates;
- escalation rates;
- human disagreement rates;
- output distribution change;
- increased uncertainty;
- model fallback events;
- increased missing-data warnings;
- increased user correction;
- increased downstream complaints;
- changed decision outcomes.

Monitoring should not be limited to model metrics.

It should include workflow metrics.

A model may appear stable while the workflow changes around it.

For example:

Human reviewers may override less often.

Users may rely on the AI output more often.

Decision time may decrease sharply.

Escalations may drop unexpectedly.

The AI output may become the default decision path.

Downstream systems may begin treating recommendations as triggers.

These are governance signals.

They may indicate automation bias, authority drift or reduced human control.

A mature deployer should therefore monitor both technical performance and behavioural use patterns.

The key question is:

Is the system still behaving as expected, and is the organization still using it as intended?

If the answer becomes uncertain, monitoring should trigger reassessment.

7.3 Monitoring Versus Passive Logging

Monitoring and logging are connected, but they are not the same.

Logging records what happened.

Monitoring evaluates what is happening.

A log may preserve events.

Monitoring detects whether those events indicate a change, risk or need for action.

This distinction matters because some organizations may believe that retaining logs is enough.

It is not.

A log can exist without anyone noticing a problem.

A log can capture a harmful pattern that no one reviews.

A log can preserve evidence after the fact but fail to prevent escalation.

A log can show that the workflow completed, but not that it should have paused.

Monitoring adds interpretation.

It asks whether the pattern in the logs, data, outputs, user behaviour or workflow state requires attention.

For high-risk AI systems, deployers need both.

They need logs for reconstruction.

They need monitoring for detection.

Without logs, the organization may not be able to prove what happened.

Without monitoring, the organization may not know that something is happening.

This creates a practical distinction:

Logging supports accountability.

Monitoring supports control.

Logging supports audit.

Monitoring supports intervention.

Logging supports reconstruction.

Monitoring supports prevention.

A mature Article 26 response should therefore connect logs and monitoring.

The deployer should ask:

What events are logged?

Which logged events are monitored?

Which patterns trigger alerts?

Who receives alerts?

What happens after an alert?

When does monitoring trigger human review?

When does monitoring trigger escalation?

When does monitoring trigger suspension?

When does monitoring trigger provider notification?

When does monitoring trigger authority notification?

When does monitoring trigger reassessment of intended purpose?

A monitoring system that only displays information but does not connect to action may be weak.

A dashboard that shows risk but does not create responsibility may be decorative.

A report that identifies issues but does not trigger response may not be sufficient.

For high-risk AI, monitoring should be connected to defined governance responses.

For example:

If input data becomes stale, the workflow enters HOLD.

If model performance drops below a threshold, escalation is triggered.

If human oversight is unavailable, high-consequence outputs cannot proceed.

If output distribution changes unexpectedly, review is required.

If actual use expands beyond approved scope, governance reassessment is triggered.

If a serious risk is detected, use may be suspended and relevant parties informed.

This is the difference between passive observability and operational governance.

Passive observability says:

We can see what happened.

Operational governance says:

We know when what happened requires action.

This distinction is central to Article 26.

A deployer must not only retain evidence.

It must have the capacity to act on evidence.

7.4 Operational Drift and Context Drift

Monitoring should detect drift.

But drift should not be understood only as model drift.

Many organizations are familiar with model drift.

Data changes.

User behaviour changes.

Performance changes.

Accuracy changes.

Prediction quality changes.

These are important.

But high-risk AI systems can also suffer from operational drift and context drift.

Operational drift occurs when the way the system is used changes over time.

Context drift occurs when the surrounding conditions that make use valid change over time.

These forms of drift may be just as important as model drift.

A model may continue performing as expected, but the workflow may change.

A system may still produce accurate outputs, but those outputs may now influence higher-consequence decisions.

A human oversight process may remain documented, but operational pressure may reduce its practical value.

A recommendation may remain advisory in policy, but become practically decisive in the workflow.

A data source may remain connected, but stop refreshing.

A provider instruction may remain in the documentation, but no longer be reflected in actual use.

An approval may remain visible, but the assumptions behind it may no longer hold.

This is operational and context drift.

It is dangerous because it may not look like technical failure.

The system still works.

Users still use it.

Outputs still appear.

The workflow still completes.

The dashboard may remain green.

But the control environment has changed.

Examples include:

A HR AI tool approved for screening begins to influence final hiring decisions.

A procurement AI tool originally used for low-value supplier triage is later used for high-value supplier approval.

A clinical AI assistant originally used by specialists is later used by general staff.

A public-service triage tool originally used for prioritization begins to influence eligibility decisions.

A compliance agent originally used to draft summaries begins sending workflow triggers to case-management software.

An AI system originally used with internal data begins incorporating external data of uncertain quality.

A model fallback occurs, but the workflow continues with the same authority level.

In each case, the risk is not necessarily that the AI model is malfunctioning.

The risk is that the deployment context has shifted.

Monitoring must therefore track not only model outputs, but also the relationship between the system and its environment.

The deployer should monitor:

- actual use versus approved use;
- current workflow versus original workflow;
- current users versus trained users;
- current data sources versus approved data sources;
- current output authority versus intended output authority;
- current downstream integrations versus original integration map;
- current oversight capacity versus required oversight model;
- current risk context versus original risk assessment;
- current policy environment versus policy version at approval;

- current operational consequence versus intended consequence.

This is a richer form of monitoring.

It requires governance, technical, operational and business teams to work together.

Technical teams may detect API changes, data changes and system behaviour.

Operational teams may detect workflow changes and user reliance.

Compliance teams may detect regulatory or policy changes.

Risk teams may detect changing consequence levels.

Business owners may detect changes in process dependency.

Human overseers may detect practical pressure points.

No single team sees the whole control environment.

This is why deployer monitoring must be cross-functional.

The AI system may be technical, but the deployment context is organizational.

Monitoring must reflect that.

7.5 Monitoring Signals That Should Trigger Reassessment

Monitoring is useful only if it leads to reassessment when conditions change.

A deployer should therefore define monitoring signals that trigger review, escalation, suspension or refusal to continue.

These signals should be risk-based.

Not every change requires the same response.

Some changes may require documentation.

Some may require human review.

Some may require compliance assessment.

Some may require provider consultation.

Some may require temporary suspension.

Some may require notification to relevant authorities or affected parties.

The deployer should define trigger categories.

A practical trigger model may include:

- Technical triggers
- Performance triggers
- Input data triggers
- Human oversight triggers
- Scope and use triggers
- Authority triggers
- Downstream effect triggers
- Incident and harm triggers
- Legal and policy triggers
- Evidence integrity triggers

Technical triggers may include system failure, unexpected output format, model unavailability, integration failure, security events or tool-access anomalies.

Performance triggers may include accuracy degradation, increased false positives, increased false negatives, group-level performance differences, calibration changes or unexpected output distributions.

Input data triggers may include stale data, missing required fields, source failure, unexpected data format, new data source connection or representativeness concerns.

Human oversight triggers may include reviewer unavailability, unusually low override rates, rising reviewer disagreement, review time compression, lack of evidence access or escalation pathway failure.

Scope and use triggers may include new user groups, new workflows, use outside intended purpose, use for higher-consequence decisions or informal repurposing.

Authority triggers may include the AI output becoming more influential, advisory output becoming default, automatic approval pathways, reduced human challenge or model fallback without authority reduction.

Downstream effect triggers may include connection to new systems, output triggering automated actions, workflow integration changes, external communication based on output or decision propagation.

Incident and harm triggers may include complaints, affected-person challenges, near misses, incorrect decisions, safety concerns, legal challenges or serious risk indicators.

Legal and policy triggers may include internal policy change, regulatory change, provider instruction update, risk appetite change or new supervisory expectation.

Evidence integrity triggers may include missing logs, incomplete audit trails, unverifiable input state, broken lineage, unrecorded overrides or loss of decision context.

Each trigger should connect to a governance response.

Possible responses include:

- continue with record;
- review at next scheduled assessment;
- immediate human review;
- escalation to AI governance owner;
- escalation to compliance or legal;
- provider consultation;
- enhanced monitoring;
- temporary HOLD;
- suspension of use;
- affected-person review;
- incident assessment;
- authority notification where required;
- refusal to allow continuation.

This is where runtime governance becomes concrete.

Monitoring is not only about seeing.

It is about deciding when seeing requires action.

A mature deployer should not rely on ad hoc judgment every time a signal appears.

The organization should define response logic in advance.

For example:

If input data freshness exceeds threshold: HOLD.

If human oversight unavailable for high-consequence output: ESCALATE.

If system used outside approved workflow: HOLD pending review.

If model fallback occurs: reduce authority and require fresh evidence.

If output triggers downstream automation: require control-gate review.

If logs fail to record required decision context: suspend consequential use until evidence integrity is restored.

If serious risk is detected: suspend use and follow incident escalation procedure.

This is not about making AI governance rigid.

It is about making control conditions explicit.

The more consequential the workflow, the less acceptable it is for continuation to depend on invisible assumptions.

The deployer should know when the system is allowed to continue.

It should also know when it is no longer sure.

In high-risk AI governance, uncertainty should not always be hidden.

Sometimes uncertainty should become a state.

UNKNOWN should become HOLD.

This may be one of the most important practical principles for Article 26 monitoring.

If the deployer cannot determine whether control conditions remain valid, the system should not automatically proceed as if they are valid.

The absence of evidence should not become permission.

Monitoring should therefore support a simple but powerful question:

Do current conditions still support continuation?

If yes, continue.

If no, stop or refuse.

If uncertain, hold and reassess.

This is monitoring as operational control.

Chapter 7 Summary

Monitoring is not only a technical function.

For high-risk AI systems, monitoring is a deployer governance obligation and a runtime control mechanism.

Traditional technical monitoring may track uptime, latency, errors and availability. These are necessary, but not sufficient.

A deployer must also monitor whether the system remains used according to instructions, whether input data remains suitable, whether human oversight remains meaningful, whether actual use remains within intended purpose, whether authority has drifted, whether downstream consequences have changed and whether evidence remains sufficient for reconstruction.

This chapter distinguishes monitoring from logging.

Logging records what happened.

Monitoring detects whether what is happening requires action.

Both are necessary.

Without logging, the deployer may lack reconstructability.

Without monitoring, the deployer may lack timely control.

The chapter also distinguishes model drift from operational drift and context drift.

Model drift concerns changes in system performance.

Operational drift concerns changes in how the system is used.

Context drift concerns changes in the surrounding conditions that make use valid.

For deployers, all three matter.

A high-risk AI system may remain technically functional while its operational environment changes enough to undermine continued validity.

This is why monitoring must be connected to reassessment triggers.

Deployers should define technical, performance, input data, oversight, scope, authority, downstream effect, incident, legal and evidence-integrity triggers.

Each trigger should connect to a governance response, such as review, escalation, hold, suspension, provider consultation, incident assessment or refusal to continue.

The central monitoring question is:

Do current conditions still support continuation?

If yes, the workflow may continue.

If no, it should stop, escalate or refuse.

If uncertain, it should hold.

Article 26 monitoring should therefore be understood as more than observation.

It is the deployer's capacity to detect when continuation is no longer justified.

Chapter 8

Logs, Records and Evidence Retention

Legal anchor and framework boundary. [Article 12](#); [Article 26\(6\)](#). The Regulation anchors logging and deployer retention; decision trails and reconstructability are operational evidence concepts.

8.1 Log Retention as More Than Storage

Logs are often treated as a technical record.

A system generates events.

The events are stored.

The organization retains them.

An auditor may later inspect them.

This view is too narrow for high-risk AI systems.

For deployers, log retention is not only a storage obligation.

It is part of the organization's evidence architecture.

Article 26 requires deployers of high-risk AI systems to keep automatically generated logs to the extent those logs are under their control. This obligation is important because logs may become one of the main ways an organization proves that an AI-supported workflow was used, monitored, reviewed and controlled under valid conditions.

But the existence of logs is not enough.

A log is useful only if it preserves information that can later support understanding, reconstruction and accountability.

A log that only records that a system was accessed may not show whether the system was used appropriately.

A log that records an output may not show whether the input data was valid.

A log that records a decision may not show whether human oversight occurred.

A log that records a timestamp may not show whether authority was still valid.

A log that records a workflow completion may not show whether the workflow should have paused.

This is the central problem.

Logs can prove activity without proving control.

For high-risk AI systems, deployers need logs that help answer governance questions.

What system was used?

Which version was used?

For what intended purpose?

In which workflow?

By which user or role?

Under whose authority?

With what input data or context?

What output was produced?

What limitations or warnings were present?

Was human oversight required?

Was human oversight performed?

Was the output accepted, modified, rejected or escalated?

Did the workflow continue?

Was any boundary crossed?

Was the output used to affect a person, process, decision or operational consequence?

These questions are not only technical.

They are governance questions.

Log retention therefore becomes a practical foundation for evidence of control.

A deployer should not ask only:

Are logs being stored?

The deployer should ask:

Do the logs preserve enough information to show why the system was allowed to continue?

This distinction matters.

Many systems generate large quantities of logs, but not all logs are meaningful for governance.

Some logs are too technical.

Some are too incomplete.

Some are not connected to user actions.

Some are not connected to workflow stages.

Some do not preserve input state.

Some do not preserve human review.

Some do not preserve policy version.

Some do not preserve decision context.

Some are retained but not searchable.

Some are retained but not understandable.

Some are retained but not linked to the relevant case, person, process or output.

In those situations, an organization may satisfy a basic retention practice while still failing to create useful evidence.

For Article 26 purposes, the value of logs should be assessed by their reconstructability.

Can the organization reconstruct what happened?

Can it reconstruct what the AI system saw?

Can it reconstruct what the AI system produced?

Can it reconstruct who reviewed it?

Can it reconstruct whether the output influenced a decision?

Can it reconstruct whether control conditions were valid?

Can it reconstruct whether escalation should have occurred?

If not, the log environment may be weak.

This is why logs should be designed with future questions in mind.

A mature deployer should assume that one day it may need to explain a specific AI-supported workflow to an auditor, regulator, board member, incident investigator, affected person or court.

The relevant question will not be:

Did the system generate logs?

The relevant question will be:

Do the logs help prove that the system was governed?

8.2 Reconstructability and Audit Usefulness

Reconstructability is the ability to rebuild the relevant facts of an AI-supported workflow after the event.

It is one of the most important purposes of logging.

Without reconstructability, log retention may become a hollow control.

The organization may have stored data, but not the right data.

It may have records, but not evidence.

It may have timestamps, but not context.

It may have outputs, but not the conditions that produced them.

This matters because high-risk AI systems often operate inside complex workflows.

The AI output may be only one part of the process.

The workflow may include data retrieval, user input, model generation, tool use, human review, policy checks, escalation, downstream action and external communication.

If only one part of that process is logged, the organization may not be able to reconstruct the full decision path.

For example:

A HR AI system produces a candidate recommendation.

A log records that the recommendation was generated.

But does the log show which candidate data was used?

Does it show whether protected characteristics were excluded?

Does it show whether the user reviewed the output?

Does it show whether the recommendation was modified?

Does it show whether the candidate was shortlisted because of the output?

Does it show whether another human decision-maker relied on the output?

A procurement AI system recommends a supplier.

A log records the recommendation.

But does the log show supplier data freshness?

Does it show whether sanctions, financial risk or performance data were current?

Does it show whether the approving manager still had authority?

Does it show whether risk thresholds changed after the recommendation?

Does it show whether the output triggered downstream approval?

A clinical AI system generates a summary.

A log records the summary.

But does it show which patient records were included?

Does it show whether recent lab results were missing?

Does it show whether the clinician saw warnings?

Does it show whether the summary influenced the clinical pathway?

Does it show whether the clinician accepted, corrected or rejected the summary?

A compliance AI system drafts a case assessment.

A log records the draft.

But does it show the policy version?

Does it show the evidence sources used?

Does it show whether the system omitted conflicting information?

Does it show whether a human analyst escalated the case?

Does it show whether the draft became part of a regulated decision?

These examples show why reconstructability matters.

A useful log environment should preserve the relationship between:

- system;
- input;
- context;
- output;

- user;
- authority;
- oversight;
- workflow;
- consequence;
- evidence;
- action.

Audit usefulness depends on this relationship.

An auditor does not need every possible technical detail.

An auditor needs enough information to understand whether the system was used properly and whether controls operated as expected.

This means logs should be structured around governance questions.

A useful audit record should help answer:

What happened?

When did it happen?

Which system was involved?

Which version or configuration was involved?

Who initiated the workflow?

Who had responsibility?

What input data was used?

What output was produced?

What human review occurred?

What control checks were applied?

What warnings or exceptions occurred?

What downstream action followed?

What evidence supports the decision to continue?

Was any escalation triggered?

Was any override recorded?

Were logs complete?

Were logs protected from alteration?

This does not mean that every deployer must build an excessive surveillance system around AI use.

It means that logging should be proportionate to risk.

The greater the consequence of the AI-supported workflow, the stronger the reconstructability requirement should be.

A low-impact internal productivity tool may require lighter logging.

A high-risk AI system used in employment, finance, healthcare, public services, education, law enforcement, critical infrastructure or legal processes requires stronger evidence.

For deployers, audit usefulness should be tested before a problem occurs.

The organization should perform reconstruction exercises.

It should select past AI-supported workflows and ask:

Can we reconstruct the case?

Can we identify the system and version?

Can we identify the input condition?

Can we identify the output?

Can we identify human oversight?

Can we identify whether the output affected consequence?

Can we identify whether the workflow remained within intended purpose?

Can we identify whether escalation should have occurred?

If the answer is no, the logging design should be improved.

This is practical Article 26 readiness.

8.3 Logs as Evidence of Control

A log is not automatically evidence of control.

It becomes evidence of control only when it shows that a relevant control condition existed, operated or failed in a way that can be inspected.

This distinction is central to the whole study.

A log may show that a user accessed the AI system.

That is evidence of access.

It is not necessarily evidence of appropriate use.

A log may show that an output was generated.

That is evidence of output.

It is not necessarily evidence that the output was valid.

A log may show that a human clicked approve.

That is evidence of an approval action.

It is not necessarily evidence of meaningful oversight.

A log may show that a workflow completed.

That is evidence of completion.

It is not necessarily evidence that continuation was justified.

To become evidence of control, the record must connect the action to the control condition.

For example:

If the control condition is human oversight, the log should show whether oversight occurred, who performed it, what they saw, what action they took and whether their intervention could affect the outcome.

If the control condition is input data freshness, the log should show data source, last refresh time, freshness status and whether continuation was allowed despite any warning.

If the control condition is authority, the log should show the responsible role, approval state, policy version or delegation state that allowed the workflow to proceed.

If the control condition is intended purpose, the log should show the use case, workflow, process and whether the system was used within approved scope.

If the control condition is escalation, the log should show the trigger, escalation route, recipient, response and final decision.

This is evidence of control.

It shows not only that something happened, but that the organization had a mechanism for determining whether it should happen.

This is especially important in agentic AI systems.

An agent may perform multiple steps:

- retrieve context;
- call tools;
- generate intermediate reasoning;
- route to another model;
- summarize evidence;
- recommend action;
- update a system;
- notify a user;
- trigger a workflow.

If logs record only the final output, the organization may not understand how the workflow became consequential.

For agentic systems, logs should capture key transition points.

When did the agent retrieve evidence?

Which evidence did it use?

Which tools did it call?

Which permissions were active?

Which model or model version was used?

Was any fallback model used?

Was authority reduced after fallback?

Was human review required?

Was the agent allowed to write to a system?

Did a recommendation become an executable signal?

Did the agent encounter uncertainty?

Did the workflow continue despite missing evidence?

These are not merely technical details.

They are control evidence.

The same principle applies to non-agentic high-risk AI systems.

The deployer should identify the control points in the workflow and ensure that logs preserve evidence around those points.

Control points may include:

- system access;
- input validation;
- data freshness check;
- provider instruction boundary;
- output generation;
- human review;
- override;
- escalation;
- approval;
- downstream transfer;
- external communication;
- final decision;
- incident flag;
- suspension;
- refusal to continue.

Logs should help prove whether these control points operated.

This is where Article 26 log retention intersects with runtime governance.

The purpose is not only to keep records.

The purpose is to preserve evidence that the organization controlled the AI-supported workflow.

A mature deployer should therefore classify logs by evidential value.

Some logs are technical health logs.

Some logs are security logs.

Some logs are access logs.

Some logs are workflow logs.

Some logs are decision logs.

Some logs are oversight logs.

Some logs are escalation logs.

Some logs are evidence-condition logs.

Some logs are authority-state logs.

All may matter, but not all serve the same purpose.

The organization should know which logs support Article 26 evidence of control.

This classification helps avoid both under-logging and meaningless over-logging.

The goal is not to collect everything.

The goal is to preserve what matters.

8.4 What Logs Should Preserve

A practical log design for high-risk AI deployers should preserve several categories of information.

The exact logging requirements will depend on the system, use case, legal environment, risk level and technical capabilities. But a strong deployer logging model should consider at least the following categories.

1. System Identity

The log should identify the AI system involved.

This may include:

- system name;
- provider;
- system version;
- model version;
- configuration;
- deployment environment;
- high-risk classification;
- intended purpose;
- approved use case.

System identity matters because organizations may use multiple AI systems, versions or configurations.

Without system identity, later reconstruction becomes difficult.

2. Workflow Identity

The log should identify the workflow in which the system was used.

This may include:

- business process;
- use case;
- case type;
- internal workflow stage;
- decision pathway;
- affected department;
- downstream system;
- consequence category.

Workflow identity matters because the same system may be used in different processes with different risk levels.

3. User and Role Context

The log should identify the relevant human or organizational role.

This may include:

- user identity where appropriate;
- role;
- department;
- permissions;
- oversight assignment;
- approval authority;
- delegation state;
- training status where relevant.

This supports accountability and helps determine whether the system was used by authorized and competent persons.

4. Input Data Condition

The log should preserve enough information about input data to assess validity.

This may include:

- data sources;
- data source versions;
- last refresh time;
- completeness status;
- missing required fields;
- known limitations;

- user-entered information;
- data validation result;
- representativeness or suitability flag;
- data warning status.

This is essential because output validity depends on input conditions.

5. Output and Decision Context

The log should preserve the AI output and the decision context around it.

This may include:

- output generated;
- confidence or uncertainty indicators where available;
- warnings;
- system limitations shown to the user;
- alternative outputs;
- explanation or reasoning artifact where available;
- decision recommendation;
- final human decision;
- modifications made by the human reviewer;
- downstream use of the output.

This helps determine whether the output shaped consequence.

6. Human Oversight Record

The log should preserve evidence of human oversight.

This may include:

whether oversight was required;
 whether oversight occurred;
 reviewer identity or role;
 timestamp;
 information shown to reviewer;
 review decision;
 override;
 escalation;
 refusal;
 comments;
 second review where required.

This is critical for proving that oversight was meaningful.

7. Authority and Policy State

The log should preserve the authority and policy conditions that allowed continuation.

This may include:

- approval status;
- policy version;
- risk threshold;
- authority holder;
- delegation state;
- access permission;
- model authority level;
- fallback status;
- control gate decision;
- allow/hold/escalate/refuse state.

This helps answer whether the workflow was allowed to proceed under current conditions.

8. Monitoring and Alert State

The log should preserve monitoring signals relevant to the workflow.

This may include:

- performance warnings;
- stale data alerts;
- anomaly detection;
- drift signals;
- model unavailability;
- tool failure;
- human oversight unavailable;
- scope warning;
- escalation trigger.

This helps prove whether the organization detected relevant risk conditions.

9. Downstream Consequence

The log should preserve whether the AI output affected another system, person, process or decision.

This may include:

- downstream system write;
- task creation;
- notification;
- customer communication;
- case status change;
- decision finalized;
- recommendation accepted;
- external action triggered;
- report generated;
- regulated process affected.

This is essential because AI governance risk often attaches when output becomes consequence.

10. Integrity and Retention Metadata

The log should preserve metadata that supports trust in the record itself.

This may include:

- timestamp;
- retention period;
- storage location;
- access history;
- alteration history;
- hash or integrity check where appropriate;
- retention policy;
- deletion schedule;
- legal hold status where relevant.

This supports auditability and protects the evidential value of the logs.

These categories are not a universal checklist for every system.

They are a reference model.

A deployer should adapt them to the specific system and risk context.

The guiding principle is simple:

The log should preserve enough information to reconstruct whether the AI-supported workflow remained within valid control conditions.

If the log cannot help answer that question, its governance value is limited.

8.5 From Audit Trail to Decision Trail

Many organizations already understand audit trails.

An audit trail records activity.

It shows who accessed a system, what action occurred and when.

Audit trails are important.

But high-risk AI systems often require something stronger:

a decision trail.

A decision trail does not only show that actions occurred.

It shows how an AI-supported output moved through evidence, oversight, authority and workflow toward consequence.

This distinction is important.

An audit trail may show:

User accessed system at 10:00.

AI output generated at 10:01.

User approved at 10:03.

Case closed at 10:05.

This is useful, but incomplete.

A decision trail should show:

What system and version generated the output?

What input data was used?

Was the data fresh and complete?

What policy version applied?

Was the system used within intended purpose?

Was human oversight required?

What did the human reviewer see?

Did the human reviewer have authority?

Was the output accepted, modified or rejected?

Were any warnings present?

Was escalation triggered?

Did the AI output influence the final decision?

Did the workflow create external consequence?

Why was continuation allowed?

This is a much richer form of evidence.

It connects the AI output to the governance environment around it.

For high-risk AI systems, this may be the difference between formal accountability and practical accountability.

A basic audit trail can prove that a process happened.

A decision trail can help prove whether the process was controlled.

The concept of a decision trail is especially important because AI systems often influence decisions indirectly.

The AI output may not be the final decision.

But it may shape the human decision.

It may frame the evidence.

It may prioritize the case.

It may create the shortlist.

It may suggest the risk category.

It may draft the reasoning.

It may recommend escalation.

It may influence what the human sees first.

It may reduce attention to alternative evidence.

It may become the default path.

If logs record only the final human decision, they may miss the AI system's influence.

The decision trail should therefore preserve the path of influence.

This includes:

- whether the AI output was viewed;
- whether it was relied upon;
- whether it was modified;
- whether it was challenged;
- whether alternative evidence was reviewed;
- whether the human decision differed from the AI output;
- whether the AI output became part of the case record;
- whether downstream actions were triggered.

This does not mean that every human thought process must be captured.

It means that the organization should preserve enough evidence to understand how the AI system participated in the workflow.

This is especially important for affected-person rights, regulatory review, internal accountability and incident investigation.

A person affected by an AI-supported process may not care only that a human made the final decision.

They may care whether the AI system shaped the decision environment.

A regulator may not care only that a policy existed.

They may ask whether the deployer can show how the system was used and controlled.

A board may not care only that logs are retained.

They may ask whether the organization can prove that high-risk AI systems are operating within valid boundaries.

This is why the audit trail must evolve into a decision trail.

For Article 26, this evolution is essential.

The deployer's obligation is not only to preserve automatically generated logs.

The deployer's practical challenge is to preserve evidence that those logs can support.

Logs should help the organization answer:

Was the system used correctly?

Was the input valid?

Was oversight meaningful?

Was authority current?

Was the output allowed to influence consequence?

Was continuation justified?

If logs cannot support these questions, they may not be sufficient for mature deployer governance.

This is the core of Chapter 8.

Log retention is not the end of accountability.

It is the beginning of reconstructability.

Chapter 8 Summary

Logs and records are central to Article 26 deployer governance.

But logs should not be understood only as technical storage.

For high-risk AI systems, logs are part of the organization's evidence architecture.

Article 26 requires deployers to retain automatically generated logs to the extent those logs are under their control, for an appropriate period linked to intended purpose and at least six months unless applicable law provides otherwise. Article 12 complements this by requiring high-risk AI systems to technically allow automatic recording of events over the system's lifetime.

The practical challenge is not only retention.

The practical challenge is usefulness.

A deployer may have logs but still lack evidence of control.

A log may show that a system was accessed, but not whether it was used appropriately.

A log may show that an output was generated, but not whether input data was valid.

A log may show that a human clicked approve, but not whether oversight was meaningful.

A log may show that a workflow completed, but not whether continuation was justified.

For this reason, deployers should design logging around reconstructability.

A mature log environment should help reconstruct the system used, the workflow, the input data condition, the output, the human oversight, the authority state, the monitoring signals, the downstream consequence and the reason continuation was allowed.

This chapter distinguishes between audit trails and decision trails.

An audit trail records activity.

A decision trail records the path by which AI-supported output moved through evidence, oversight, authority and workflow toward consequence.

For high-risk AI systems, decision trails may become essential.

The deployer should be able to answer:

What happened?

Why was it allowed to happen?

Who or what had authority?

What evidence was used?

Was oversight possible?

Was the output allowed to influence consequence?

Were logs complete enough to reconstruct control?

The future of AI governance will not be defined only by whether logs exist.

It will be defined by whether those logs can prove that control existed when it mattered.

Chapter 9

Serious Risk, Incident and Escalation Pathways

Legal anchor and framework boundary. [Article 26\(5\)](#); [Article 73](#); [Article 79](#). The chapter distinguishes statutory notification and suspension duties from recommended evidence-preservation and escalation architecture.

9.1 When Monitoring Becomes Escalation

Monitoring is not complete until it connects to escalation.

A deployer may monitor a high-risk AI system continuously, collect logs, review dashboards, observe outputs, track performance and identify anomalies. But if monitoring does not create a clear pathway for action, it remains incomplete.

Monitoring detects.

Escalation responds.

This distinction matters because high-risk AI systems may generate signals that require different levels of intervention.

Some signals may require routine review.

Some may require a human reviewer to examine a case.

Some may require governance or compliance escalation.

Some may require provider notification.

Some may require suspension of use.

Some may require serious incident assessment.

Some may require communication with relevant market surveillance authorities.

Some may require immediate containment.

For deployers, the practical question is not only:

Did we see the issue?

The question is:

Did we know what to do when we saw it?

This is the operational meaning of escalation.

Escalation is the pathway by which a monitoring signal becomes accountable action.

Without escalation, monitoring may become decorative.

A dashboard may show a risk.

A log may preserve an error.

A user may report concern.

A reviewer may disagree with an output.

A data source may become stale.

An AI output may affect a person unfairly.

A model may behave unexpectedly.

A downstream workflow may change the consequence of an AI recommendation.

But if no one owns the next step, the organization may continue operating under degraded conditions.

This is one of the most dangerous states in AI governance.

The organization sees something, but does not convert seeing into control.

Article 26 makes this especially important because deployers have obligations not only to use and monitor high-risk AI systems, but also to respond when risk conditions arise.

If the deployer has reason to consider that the use of the system in accordance with the instructions may result in the system presenting a relevant risk, it must inform the appropriate parties and suspend use.

This means monitoring cannot be separated from escalation.

A deployer should define in advance:

What signals require review?

What signals require escalation?

Who receives the escalation?

Who decides whether use should continue?

Who decides whether use should be suspended?

Who contacts the provider?

Who contacts the distributor or importer where relevant?

Who contacts the market surveillance authority where required?

Who determines whether an event is a serious incident?

Who preserves evidence?

Who communicates internally?

Who informs affected persons where required?

Who verifies corrective action?

If these questions are not answered before an incident occurs, the organization may lose time when time matters most.

Escalation pathways should therefore be designed before deployment and tested during operation.

The goal is not only to react after harm.

The goal is to detect when harm, serious risk or loss of control may be emerging and prevent further consequence.

This is where Article 26 becomes operationally serious.

The deployer must not only observe the system.

It must know when observation requires action.

9.2 Serious Risk Detection

Not every anomaly is a serious risk.

Not every error is a serious incident.

Not every complaint requires regulatory notification.

But deployers must be able to distinguish ordinary issues from serious risk signals.

This requires a risk classification process.

A serious risk signal may arise when an AI system may affect health, safety, fundamental rights or other high-consequence interests in a way that requires urgent action.

The risk may come from the AI system itself.

It may come from input data.

It may come from actual use.

It may come from a mismatch between intended purpose and deployment context.

It may come from human oversight failure.

It may come from automation bias.

It may come from tool use.

It may come from downstream workflow integration.

It may come from external dependency failure.

It may come from the inability to reconstruct what happened.

Serious risk detection therefore cannot be limited to technical malfunction.

A system can present risk while technically operating normally.

For example:

A recruitment AI system begins excluding a protected group because input data or ranking logic creates discriminatory patterns.

A clinical AI assistant produces summaries that omit recent lab results, leading to unsafe reliance.

A credit scoring system uses stale financial data during market volatility.

A public-benefit AI system applies an outdated eligibility rule.

A procurement agent recommends a supplier despite a new sanctions or financial risk flag.

A law-enforcement support tool produces risk classifications that are treated as authoritative despite intended advisory status.

A workflow agent sends AI-generated output into downstream automation without required human review.

A model fallback occurs but the system continues with the same authority level.

In each case, the issue may not look like a system crash.

The AI system may continue producing outputs.

The workflow may continue completing.

The dashboard may remain green.

But a serious risk may be emerging.

This is why deployers need serious risk detection criteria.

These criteria should be defined by system type, use case and consequence level.

A useful serious risk detection framework may include:

- harm potential;
- affected persons;
- fundamental rights impact;
- safety implications;
- scale of affected workflow;
- recurrence;
- severity;
- reversibility;
- detectability;
- evidence integrity;
- oversight failure;
- use outside intended purpose;
- downstream automation;
- regulatory sensitivity.

The deployer should ask:

Could this issue affect a person's rights, access, opportunity, safety or legal position?

Could this issue affect a regulated process?

Could this issue affect many cases?

Could this issue remain hidden if not escalated?

Could the system continue producing harmful outputs?

Could human oversight fail to detect it?

Could the output become operational before review?

Could the organization reconstruct the issue later?

If the answer is yes, escalation may be required.

A mature deployer should also define near-miss criteria.

A near miss occurs when harm did not occur, but the system came close enough to reveal a control weakness.

For example:

A human reviewer catches a serious AI error before it affects a decision.

A stale data warning appears just before a high-value procurement recommendation is approved.

A public-service AI system almost applies an outdated rule.

A clinical AI system omits a critical fact, but the clinician identifies it.

A workflow agent attempts to use a tool outside approved scope but is blocked.

Near misses are valuable because they reveal where control almost failed.

They should be logged, reviewed and used to improve governance.

Organizations often learn more from near misses than from normal operation.

Serious risk detection should therefore include both actual harm and potential harm.

The key governance question is:

Did the deployer identify that continued use could create unacceptable risk?

If yes, the workflow should not continue as normal.

It should enter escalation.

9.3 Incident Reporting and Internal Escalation

Incident reporting is not only an external regulatory process.

It begins internally.

Before an organization can notify anyone, it must know that an incident may have occurred.

This requires internal escalation pathways that connect front-line observation to accountable decision-making.

A user may notice an unexpected output.

A human overseer may disagree with a recommendation.

A data owner may discover a source failure.

A technical team may detect a model issue.

A compliance team may identify use outside intended purpose.

A customer or affected person may complain.

An audit team may find incomplete logs.

A manager may notice unusual decision patterns.

Each of these signals may be the beginning of an incident pathway.

The organization needs a process that allows such signals to move to the right level.

This process should be simple enough to be used, but structured enough to preserve accountability.

An internal AI incident escalation process should define:

- what must be reported;
- who can report;
- where reports go;
- what information must be included;
- who triages the report;
- how severity is assessed;
- who decides whether use continues;
- when provider notification is required;
- when legal or compliance review is required;
- when regulator notification may be required;
- who preserves evidence;
- who communicates with affected internal teams;
- who records the final decision.

The first internal escalation question should be:

Can the system continue safely while the issue is assessed?

If the answer is unclear, the default should not automatically be continuation.

A HOLD state may be appropriate.

This is especially important for high-risk AI systems because continued use may create additional affected cases.

If the system is producing risky outputs, every continued workflow may multiply the problem.

For this reason, escalation procedures should include temporary containment.

Containment may include:

- pausing the affected workflow;
- suspending use of the AI system;
- disabling a specific feature;

- blocking downstream automation;
- requiring manual review;
- preventing external communication;
- isolating affected cases;
- preserving logs;
- notifying oversight roles;
- reverting to a previous validated process;
- switching to a reduced-authority fallback only where governed.

The organization should also preserve evidence immediately.

Incident evidence may include:

- system logs;
- input data state;
- output records;
- user actions;
- human review records;
- warnings displayed;
- escalation timestamps;
- provider instructions;
- policy version;
- model version;
- data source status;
- affected cases;
- downstream actions;
- communications;
- corrective actions.

Evidence preservation is critical because logs may be overwritten, data may change, users may forget details and workflows may continue.

A deployer should assume that if an incident becomes serious, it may need to reconstruct the event later for regulators, affected persons, auditors, legal teams, providers or internal governance bodies.

Internal escalation should also clarify the relationship with the provider.

Article 26 places deployers in a position where they may need to inform the provider, distributor, importer or market surveillance authority depending on the nature of the issue.

This means the deployer should know:

Who is the provider contact?

What is the incident contact channel?

What information should be supplied?

What contractual or support obligations exist?

What response time is expected?

Who communicates externally?

What happens if the provider cannot be reached?

These details should not be discovered during an incident.

They should be part of deployer readiness.

The goal is to avoid confusion at the moment of risk.

A mature deployer does not wait until a serious incident occurs to design escalation.

It designs escalation as part of deployment.

9.4 Pause, Escalate or Stop Conditions

High-risk AI governance requires more than pass/fail logic.

Many risk situations are not simple errors.

They are uncertainty states.

The system may not be clearly wrong.

The workflow may not have failed.

The output may be plausible.

The data may be partially complete.

The human reviewer may be uncertain.

The issue may be emerging but not confirmed.

The risk may depend on downstream consequence.

In these situations, the deployer needs more than a binary continue/stop model.

A mature runtime governance model should include at least four operational states:

ALLOW.

HOLD.

ESCALATE.

REFUSE.

ALLOW means current conditions support continuation.

HOLD means the workflow should pause because required control conditions are uncertain or incomplete.

ESCALATE means the issue requires review by a higher authority, human overseer, governance function, provider, compliance team or other responsible party.

REFUSE means the workflow should not proceed under current conditions.

These states are useful because they make uncertainty visible.

In many organizations, uncertainty is hidden inside normal workflow.

If the system is uncertain but still produces an answer, users may proceed.

If input data is stale but no hard stop exists, users may proceed.

If human oversight is unavailable but the system output is ready, users may proceed.

If authority is unclear but the workflow has momentum, users may proceed.

If provider instructions do not cover a use case but the system works, users may proceed.

This is how governance failure can occur without obvious technical failure.

The workflow continues because nothing stops it.

Article 26 requires deployers to think differently.

A high-risk AI system should not proceed merely because it can.

It should proceed because current control conditions support continuation.

This creates the need for pause conditions.

A pause condition is a rule or signal that prevents automatic continuation until a required control condition is restored or reviewed.

Examples include:

- input data stale;
- required data missing;
- use case outside intended purpose;

- human oversight unavailable;
- reviewer lacks authority;
- system output conflicts with evidence;
- model fallback occurred;
- monitoring alert active;
- downstream automation detected;
- log integrity failure;
- policy version mismatch;
- affected-person risk identified;
- provider instruction uncertainty;
- serious risk signal detected.

A pause condition does not mean the system has failed.

It means the system is not currently admissible for continuation.

This is a critical distinction.

Stopping can be a feature.

Holding can be a protection.

Escalation can be evidence of control.

Refusal can be responsible governance.

A deployer should therefore define pause, escalation and stop criteria in advance.

For each high-risk AI use case, the organization should identify:

What conditions permit continuation?

What conditions require HOLD?

What conditions require human review?

What conditions require escalation?

What conditions require suspension?

What conditions require provider notification?

What conditions require authority notification?

What conditions require affected-person review?

What conditions require a full incident investigation?

This turns escalation into a structured governance mechanism.

It also supports evidence.

If a case later comes under review, the organization can show not only that it acted, but that it had predefined criteria for action.

This is stronger than ad hoc judgment.

It shows that the organization expected uncertainty and designed for it.

In high-risk AI, this matters.

The most dangerous failure may not be a visible error.

It may be a workflow that completes successfully when it should have stopped.

Pause, escalate and stop conditions are the controls that prevent that failure.

9.5 Evidence That Escalation Pathways Were Real

A documented escalation pathway is not enough.

The deployer must be able to show that the escalation pathway was real.

This means it was known, usable, resourced, tested and acted upon when needed.

Many organizations have escalation policies.

Fewer organizations can prove that escalation works under operational pressure.

An escalation pathway may fail because:

- users do not know it exists;
- users do not understand when to use it;
- the reporting channel is unclear;
- the responsible person is unavailable;
- the escalation owner lacks authority;
- escalation is treated as disruption;
- staff fear blame;
- issues are resolved informally without record;
- provider contact details are outdated;
- governance teams are not connected to technical teams;
- legal and operational teams disagree on severity;
- evidence is not preserved quickly enough;
- the system continues while escalation is pending.

For high-risk AI systems, these weaknesses can be serious.

A deployer should therefore maintain evidence that escalation pathways are real.

This evidence may include:

- incident procedure documents;
- escalation matrix;
- named owners;
- role assignments;
- training records;
- user guidance;
- reporting channel records;
- provider contact records;
- escalation test results;
- tabletop exercises;
- incident simulations;
- actual escalation logs;
- response-time records;
- suspension decisions;
- corrective action records;
- post-incident reviews;
- governance committee minutes;
- evidence preservation records.

The key is not only that a pathway exists.

The key is that the pathway can be used.

A deployer should test escalation pathways before a serious incident occurs.

Testing may include:

- simulated stale data event;
- simulated model failure;
- simulated output affecting a high-consequence decision;
- simulated human oversight unavailability;
- simulated use outside intended purpose;
- simulated serious risk signal;
- simulated provider unavailability;
- simulated log integrity failure;
- simulated downstream automation issue.

These exercises help the organization understand whether escalation is practical.

They reveal gaps.

Who noticed the issue?

Who received the alert?

Who had authority to pause?

Who contacted the provider?

Who preserved evidence?

Who decided whether to continue?

Who documented the outcome?

Who informed relevant parties?

If the exercise reveals confusion, the escalation pathway should be improved.

A mature deployer should also measure escalation quality.

Useful metrics may include:

time from detection to escalation;

time from escalation to decision;

number of escalations by system;

escalation categories;

repeated triggers;

unresolved escalations;

escalation outcomes;

suspension frequency;

false alarms;

near misses;

cases where escalation was bypassed;

user confidence in escalation process.

These metrics support continuous improvement.

Escalation should not be treated as a failure of governance.

Escalation is governance working.

A high-risk AI system that escalates uncertainty is safer than a system that silently continues.

The organization should therefore build a culture in which escalation is respected.

Users should not be punished for raising concerns.

Human overseers should not be pressured to approve.

Technical teams should not hide anomalies.

Compliance teams should not treat every issue as blame.

Management should not treat pause conditions as inefficiency.

For high-risk AI systems, escalation is part of control.

The question is not whether issues will arise.

They will.

The question is whether the organization can detect them, pause when necessary, preserve evidence and respond in a structured way.

This is what makes escalation pathways real.

Chapter 9 Summary

Serious risk, incident and escalation pathways are central to deployer governance under Article 26.

Monitoring is not enough unless it connects to action.

A deployer must be able to detect when a high-risk AI system may present a relevant risk, when serious incident assessment is required, when use should be suspended, and when internal or external notification pathways should be activated.

This chapter distinguishes monitoring from escalation.

Monitoring detects signals.

Escalation converts signals into accountable action.

The deployer should define serious risk criteria based on the system, use case and consequence level.

These criteria should include not only technical malfunction, but also input data failure, oversight failure, authority drift, use outside intended purpose, downstream automation, evidence loss, affected-person harm and near misses.

Internal incident reporting should be designed before deployment.

It should define what must be reported, who can report, who triages, who decides severity, who preserves evidence, who contacts the provider and who decides whether use may continue.

The chapter proposes four practical runtime governance states:

ALLOW.

HOLD.

ESCALATE.

REFUSE.

These states help organizations avoid the mistake of treating all non-error conditions as permission to continue.

For high-risk AI systems, uncertainty should sometimes become a pause condition.

The most dangerous failure may not be a visible error.

It may be a workflow that completes successfully when it should have stopped.

Escalation pathways should therefore be real, tested and evidenced.

The deployer should be able to show that users know how to escalate, responsible people are named, provider contacts are available, evidence is preserved, suspension decisions can be made and post-incident review improves the system.

Article 26 makes escalation part of operational control.

A mature deployer should not ask only:

Did we monitor the system?

It should ask:

Did we know when monitoring required us to act?

Chapter 10

Workplace AI and Affected-Person Information

Legal anchor and framework boundary. [Article 26\(7\)-\(12\)](#); [Article 49](#); [Article 86](#). Workplace, registration, affected-person and cooperation duties are context-dependent; organisations must verify applicability under current law.

10.1 Informing Workers and Affected Persons

Transparency is not only a user-interface feature.

In high-risk AI deployment, transparency is part of the control environment.

Article 26 recognizes this clearly in the context of workplace AI and affected-person information. When high-risk AI systems are used in the workplace, deployers who are employers must inform workers' representatives and the affected workers before putting the system into service or using it. In certain high-risk use cases, deployers must also inform natural persons where an AI system is used to make or assist decisions relating to them.

These obligations matter because AI systems do not only process information.

They can change the relationship between people and institutions.

A worker may be evaluated, ranked, monitored, scheduled, allocated, screened or assessed through an AI-supported workflow.

A job applicant may be shortlisted or deprioritized.

A customer may be scored.

A patient may be triaged.

A student may be evaluated.

A public-service applicant may be prioritized or delayed.

A financial-services customer may be assessed.

A person subject to a regulated process may be affected by an AI-supported recommendation.

In such situations, the person affected may not see the system directly.

They may not know that AI was involved.

They may not know what role it played.

They may not know whether a human reviewed the output.

They may not know whether the AI output was advisory, influential or practically decisive.

This creates an accountability gap.

If affected persons do not know that AI is part of the process, they cannot meaningfully understand, question or challenge the role of the system.

If workers do not know that AI is being used in the workplace, they may not understand how monitoring, evaluation or management decisions are being shaped.

If worker representatives are not informed, collective oversight and workplace governance may be weakened.

This is why information duties should not be treated as minor administrative steps.

They are part of deployer governance.

The deployer should not ask only:

Did we send a notice?

The deployer should ask:

Did the relevant people receive meaningful information about the role of the AI system in the process that may affect them?

This requires clarity.

A meaningful notice should explain:

- that a high-risk AI system is being used;
- the process in which it is used;
- the general purpose of the system;
- the type of output it produces;
- whether the output supports or influences decisions;
- whether human oversight exists;
- who is responsible internally;
- where concerns or questions can be raised;
- whether the person may have relevant rights under applicable law;
- how the organization preserves oversight and accountability.

The exact level of detail will depend on the context, legal requirements and system design.

But the principle is clear:

People should not be affected by high-risk AI systems in ways that are invisible to them.

This is especially important in employment and workplace settings, where power imbalance already exists.

A worker may not feel able to question an automated or AI-supported assessment if the process is opaque.

A job applicant may not know that an AI system helped shape a shortlist.

An employee may not understand that productivity, scheduling or performance decisions are influenced by AI.

A manager may rely on AI-supported metrics without understanding their limitations.

This can lead to overreliance, mistrust, hidden bias and reduced human accountability.

Information duties help prevent this.

They make the presence of the system visible.

But visibility is only the first step.

The deeper governance question is whether the organization can explain the role of the system in a way that supports trust, oversight and challenge.

10.2 AI Use in Employment and Workplace Settings

Employment is one of the most sensitive areas for high-risk AI deployment.

Work affects income, dignity, opportunity, identity, power, autonomy and personal security.

AI systems used in employment can influence hiring, promotion, termination, task allocation, performance evaluation, monitoring, scheduling, training, productivity assessment and access to self-employment.

For this reason, workplace AI requires careful deployer governance.

The risks are not only technical.

They are relational and institutional.

An AI system used in the workplace may change how workers are seen by the organization.

It may convert complex human performance into scores.

It may increase surveillance.

It may create pressure to optimize behaviour according to system metrics.

It may shape management attention.

It may reinforce historical patterns.

It may reduce space for explanation, context or human judgment.

It may create a perception of objectivity even where the output depends on incomplete or biased data.

It may make workers feel that decisions are made by a system they cannot see or question.

This is why workplace AI cannot be governed only through procurement or technical review.

It requires workplace-level transparency and control.

A deployer using high-risk AI in the workplace should map:

- what process is affected;
- which workers or applicants are affected;
- what data is used;
- what output is produced;
- who sees the output;
- how the output influences decisions;
- whether the output is advisory or determinative;
- who has oversight;
- whether workers or representatives have been informed;
- whether workers can raise concerns;
- whether human review is meaningful;
- whether the system creates new monitoring or evaluation pressure;
- whether the system affects rights, opportunities or working conditions.

This mapping is essential because many workplace AI systems do not appear dramatic at first.

They may be introduced as productivity tools.

They may be described as assistance.

They may be framed as efficiency improvements.

They may generate summaries, rankings, recommendations or alerts.

But over time, those outputs may become part of management decisions.

A productivity score may influence performance review.

A scheduling recommendation may affect working patterns.

A monitoring alert may trigger disciplinary scrutiny.

A hiring ranking may shape candidate selection.

A training recommendation may influence career progression.

A risk score may affect worker trust.

The governance risk is that AI influence becomes normal before it becomes visible.

This is why worker information duties matter.

The organization should not wait until workers discover the system indirectly.

It should communicate before deployment or use, where required.

This communication should not be written in vague or overly technical language.

A worker should be able to understand:

What system is being used?

Why is it being used?

What data does it use?

What does it produce?

How may it affect me?

Who reviews it?

Can a human challenge it?

Can I raise a concern?

What safeguards exist?

This is not only about legal compliance.

It is about workplace legitimacy.

A workplace AI system that is technically effective but socially opaque may damage trust.

Trust is not created by saying “the system is fair.”

Trust is supported when people can see that the system is governed.

For deployers, workplace AI transparency should therefore be connected to human oversight, input data controls, monitoring, logs, escalation and evidence retention.

The information duty is not isolated.

It is part of the broader evidence of control.

10.3 Transparency Beyond Formal Notice

A formal notice may satisfy part of an information obligation.

But formal notice is not the same as practical transparency.

This distinction is important.

A deployer may inform workers that a high-risk AI system is being used.

The notice may be sent.

The record may be retained.

The compliance box may be checked.

But the people affected may still not understand the system’s role, influence or consequences.

This creates a gap between notification and understanding.

For high-risk AI governance, deployers should aim for meaningful transparency.

Meaningful transparency does not require exposing every technical detail, model parameter or proprietary design element.

It means providing enough information for affected persons, workers, representatives and human overseers to understand the role of the AI system in the relevant process.

A meaningful transparency model should answer:

Where is AI used?

What is it used for?

What type of output does it produce?

How does the output influence the workflow?

Is a human involved?

Can the AI output be challenged?

What data categories are used?

What limitations are known?

Who is accountable?

What happens if something goes wrong?

This is especially important because AI systems can influence decisions indirectly.

An organization may say:

The AI system does not make decisions.

That may be technically true.

But the AI system may still shape decisions.

It may rank candidates.

It may prioritize cases.

It may flag risk.

It may summarize evidence.

It may recommend action.

It may generate draft reasoning.

It may influence what a human sees first.

It may create default pathways.

It may reduce attention to alternatives.

In such cases, transparency should not focus only on whether the AI system is the final decision-maker.

It should focus on influence.

The question is:

How does the AI system affect the decision environment?

This is critical for affected-person information.

If a natural person is subject to a decision assisted by a high-risk AI system, the person may need to understand that AI played a role even if a human signed off on the final decision.

Otherwise, human involvement may be used to obscure AI influence.

A formal human decision does not erase the need to understand whether the AI system shaped that decision.

For deployers, transparency should therefore include the practical role of the system:

- screening;
- ranking;
- scoring;
- triage;
- classification;
- recommendation;
- summarization;
- alert generation;
- risk assessment;
- decision support;
- workflow routing;
- automation trigger.

This vocabulary helps people understand what the AI system actually does.

Transparency should also be layered.

Different audiences need different information.

Workers may need clear explanation of how the system may affect work conditions, evaluation or monitoring.

Worker representatives may need information sufficient to understand workplace impact and represent collective concerns.

Human overseers may need deeper information about limitations, input conditions, warnings and escalation.

Compliance teams may need evidence of notification and governance design.

Affected persons may need information about the role of the AI system in decisions relating to them.

Regulators may need records showing that information duties were fulfilled.

This means transparency should be designed, not improvised.

The deployer should define:

- what information is provided;
- to whom;
- when;
- in what form;
- at what level of detail;
- in what language;
- with what record of delivery;
- with what route for questions or challenges;
- with what update process if use changes.

This connects transparency to operational governance.

A notice at deployment may not remain sufficient if the system's use changes.

If the AI system begins affecting new workflows, new workers, new decisions or new affected persons, transparency should be reassessed.

The core principle is simple:

When AI influence expands, information duties may need to be revisited.

Transparency is not a one-time communication.

It is part of continued validity.

10.4 Human Oversight in Workplace AI

Human oversight is especially important in workplace AI.

Employment contexts contain inherent power imbalance.

Workers may be reluctant to challenge management decisions.

Applicants may not know how decisions are made.

Human reviewers may feel pressure to follow system outputs.

Managers may treat AI-generated metrics as objective.

Efficiency goals may reduce careful review.

These conditions can weaken human oversight.

A workplace AI system may be designed with human review, but real workplace dynamics may reduce the human's capacity to challenge the output.

For example:

A manager may receive AI-generated performance scores and use them as the default basis for review.

A recruiter may receive AI-ranked candidates and focus only on the top-ranked group.

A scheduling manager may accept AI-generated shifts even when they create unfair patterns.

A HR team may use AI-generated summaries without checking source evidence.

A productivity monitoring system may flag workers for review, creating pressure before context is considered.

In each case, human oversight may exist formally.

But the AI system may shape the human decision environment strongly.

This is why workplace AI oversight must be designed carefully.

A deployer should ask:

Who reviews the AI output?

Do they understand the system's limitations?

Can they challenge the output?

Can they access underlying evidence?

Are they allowed to consider contextual information?

Can they override the AI recommendation?

Are overrides recorded?

Are workers able to provide explanation or correction?

Are worker concerns reviewed by a human?

Is there an appeal or challenge route?

Are managers trained not to over-rely on the system?

Is automation bias monitored?

Human oversight in workplace AI should include the affected worker's practical ability to be heard.

This does not mean every AI-assisted workplace decision requires a formal hearing.

But for significant effects, the organization should consider whether affected persons can provide context, correct errors, challenge assumptions or request review.

A worker may know that the data used by the system is incomplete.

A worker may know that a productivity metric is misleading.

A worker may know that an absence record requires context.

A worker may know that a scheduling pattern creates hardship.

A job applicant may know that their record contains unusual circumstances not captured by a ranking system.

If the AI-supported workflow does not allow such context to enter, human oversight may be incomplete.

Human oversight should therefore be connected to contestability.

Contestability means the ability to question, challenge or seek review of an AI-supported outcome.

In workplace AI, contestability supports trust and fairness.

It also supports evidence of control.

If workers raise concerns and the organization reviews them, the deployer can show that oversight is not purely internal and silent.

It can show that affected persons have a pathway to bring missing context into the process.

The deployer should preserve evidence of:

- worker notification;
- manager training;
- human review procedures;
- override records;
- worker concern submissions;
- appeal or review outcomes;
- corrections made;
- system changes following feedback;
- monitoring of automation bias;
- escalation where significant workplace impact occurs.

This evidence helps prove that human oversight was not only assigned, but operationally meaningful.

Workplace AI is not only a technical governance problem.

It is a social governance problem.

The deployer must manage both.

10.5 Evidence of Meaningful Notification

Information duties create evidence duties.

A deployer should be able to show that workers, worker representatives or affected persons were informed where required.

But evidence of notification should not stop at proving that a message was sent.

The deployer should preserve enough evidence to show that the notification was appropriate, timely and connected to the relevant AI use.

Evidence of meaningful notification may include:

- notification text;
- date of notification;
- recipients or recipient groups;
- worker representative communication records;
- affected-worker communication records;
- acknowledgement records where appropriate;
- internal policy references;
- system description;
- use-case description;
- decision-process description;
- explanation of human oversight;
- route for questions or concerns;
- records of updates when use changed;
- training or awareness materials;
- FAQ documents;
- meeting minutes;
- worker consultation records where applicable;
- records of concerns raised and responses provided.

The deployer should also preserve evidence of timing.

In workplace contexts, information should be provided before the system is put into service or used, where required.

This timing matters because notification after deployment may not provide meaningful transparency.

A worker should not discover after the fact that a high-risk AI system has been used to affect them.

A worker representative should not learn about the system only after concerns arise.

An affected person should not be left unaware that AI supported a decision relating to them where information is required.

Timing is part of meaningful notification.

The deployer should also preserve evidence of scope.

Who was informed?

Were all affected workers included?

Were worker representatives included?

Were new workers informed when the system expanded?

Were affected persons informed when the system began influencing decisions relating to them?

Were contractors, candidates, agency workers or contingent workers considered where relevant?

Did the system affect persons outside the original notification group?

If the use of the AI system changes, notification may need to be reassessed.

For example:

A system initially used for internal HR analytics may later be used in performance review.

A scheduling tool may later incorporate productivity scoring.

A recruitment assistant may later become part of candidate ranking.

A workplace monitoring system may later feed into disciplinary processes.

A decision-support system may later affect external applicants or customers.

In such cases, the original notification may no longer cover actual use.

This connects notification to continued validity.

The deployer should ask:

Does the current use still match what people were told?

If not, the information environment may no longer be valid.

Evidence of meaningful notification should therefore support reassessment.

It should show not only what was said, but whether the notice still corresponds to actual use.

This is important because transparency can drift.

A notice may remain visible, but the system's role may change.

The organization may still claim that people were informed, but the information may no longer reflect operational reality.

This is the same pattern seen throughout this study:

presence is not validity.

A notice exists.

But does it still fit the current use?

A policy exists.

But does it still reflect the workflow?

Human oversight exists.

But can it still intervene?

Logs exist.

But do they still reconstruct control?

Monitoring exists.

But does it still detect relevant drift?

For Article 26, this is the recurring governance question.

Does this still belong here under current conditions?

Meaningful notification is part of that question.

The deployer should preserve enough evidence to show that transparency remained aligned with use.

If the AI system affects people, those people should not be invisible to the governance process.

And the AI system should not be invisible to them.

Chapter 10 Summary

Workplace AI and affected-person information are essential parts of employer governance under Article 26.

When high-risk AI systems are used in the workplace, employers who are employers must inform workers' representatives and affected workers before putting the system into service or using it. In certain high-risk use cases, employers must also inform natural persons where an AI system is used to make or assist decisions relating to them.

These duties should not be treated as minor administrative notices.

They are part of the control environment.

Information makes AI influence visible.

In workplace settings, this is especially important because AI systems may affect hiring, promotion, performance evaluation, monitoring, scheduling, task allocation, disciplinary processes or worker management.

The risks are not only technical.

They are relational, institutional and power-based.

Workers and affected persons may not see how AI has shaped a decision unless the organization makes that role visible.

This chapter distinguishes formal notice from meaningful transparency.

A notice may be sent, but people may still not understand how the AI system affects them.

Meaningful transparency should explain the system's role, purpose, output, influence, human oversight and route for questions or concerns.

The chapter also connects workplace AI to human oversight and contestability.

Human oversight in the workplace must account for power imbalance, automation bias, managerial pressure and the practical ability of workers or applicants to raise concerns or provide missing context.

Finally, the chapter defines evidence of meaningful notification.

Deployers should preserve records showing what information was provided, to whom, when, for which system, for which use case and whether the information remained aligned with actual use.

The central question is not only:

Were people informed?

The central question is:

Were people informed in a way that made the AI system's role in the affected process visible and understandable?

Article 26 information duties therefore support a broader governance principle:

People should not be affected by high-risk AI systems that are invisible to them.

Transparency is not the whole of governance.

But without transparency, governance cannot be trusted.

Chapter 11

Evidence of Control

Legal anchor and framework boundary. [Article 26](#). Evidence of control is the study's operational thesis and is not a phrase defined by the Regulation.

11.1 Defining Evidence of Control

Evidence of control is the central concept of this study.

It is the point where Article 26 moves from legal obligation to operational proof.

Many organizations can produce evidence that an AI system exists.

Many can produce evidence that a policy exists.

Many can produce evidence that a vendor was selected, a risk assessment was completed, a training module was assigned, a workflow was approved, or a log was retained.

But evidence that something exists is not the same as evidence that it was controlled.

Evidence of control means evidence that the relevant control conditions were active, valid and inspectable when the AI-supported workflow became operational.

This definition has three important parts.

First, the control conditions must be active.

A control condition is active when it actually operates in the workflow. A human oversight policy is not active merely because it exists in a document. It is active when the workflow requires oversight, the reviewer is available, the reviewer has the necessary information, and the reviewer can influence the outcome.

Second, the control conditions must be valid.

A control condition is valid when it still fits the current operational context. An approval may have been valid at deployment, but if the system is now used in a different workflow, with different data, different users, different downstream effects or different risk thresholds, the original approval may no longer support continuation.

Third, the control conditions must be inspectable.

A control condition is inspectable when the organization can later show what happened. If the organization cannot reconstruct who had authority, what evidence

was used, whether input data was fresh, whether oversight occurred, what warnings existed or why continuation was allowed, then the control may not be provable.

This is why evidence of control is stronger than ordinary compliance evidence.

Ordinary compliance evidence may show that the organization had a policy.

Evidence of control shows whether the policy operated.

Ordinary compliance evidence may show that the organization had a human-in-the-loop process.

Evidence of control shows whether the human could meaningfully intervene.

Ordinary compliance evidence may show that logs were retained.

Evidence of control shows whether those logs can reconstruct the decision environment.

Ordinary compliance evidence may show that a system was approved.

Evidence of control shows whether the approval still applied under current conditions.

Ordinary compliance evidence may show that the AI system generated an output.

Evidence of control shows whether the output was allowed to influence consequence, and why.

This distinction is critical for high-risk AI systems.

High-risk AI governance cannot depend only on static documentation because AI-supported workflows operate in changing environments.

Data changes.

Policies change.

People change.

Authorities change.

Model versions change.

Tools change.

Permissions change.

Downstream systems change.

Organizational use changes.

Risk appetite changes.

Regulatory expectations change.

The system may remain formally the same while the conditions around it change.

Evidence of control is the discipline that allows an organization to ask:

Are we still operating this system under conditions that support its use?

And if challenged later:

Can we prove it?

This is the practical meaning of Article 26 in operational terms.

The deployer is not only expected to possess policies, instructions, oversight roles, monitoring processes and logs.

The deployer must be able to show that those elements supported valid use.

Evidence of control is therefore not one document.

It is an evidence pattern.

It emerges from the relationship between:

- system identity;
- intended purpose;
- actual use;
- instructions for use;
- input data condition;
- human oversight;
- monitoring signals;
- logs;
- authority state;
- escalation pathway;
- downstream consequence;
- decision to continue, hold, escalate or refuse.

When these elements are connected, the organization can reconstruct governance.

When they are disconnected, the organization may have records but not proof of control.

This chapter defines evidence of control as the deployer's core operational challenge.

11.2 Evidence That a System Existed

The first and simplest level of evidence is evidence that an AI system existed.

This includes evidence that the organization identified the system, procured it, deployed it, registered it internally, assigned ownership and documented its intended purpose.

This evidence is necessary.

An organization cannot govern what it cannot identify.

A deployer should therefore maintain an AI system register that records:

- system name;
- provider;
- system owner;
- business owner;
- technical owner;
- risk classification;
- high-risk status;
- intended purpose;
- actual use;
- user groups;
- affected persons or groups;
- data sources;
- workflow location;
- downstream systems;
- deployment date;
- version history;
- provider instructions;
- oversight model;
- monitoring model;
- logging capability;
- review cycle.

This type of evidence establishes visibility.

It answers basic questions:

What AI systems do we use?

Where are they used?

Who owns them?

What are they for?

Are any of them high-risk?

Who is affected?

What processes do they influence?

This is the foundation of deployer governance.

Without this foundation, Article 26 compliance becomes difficult or impossible.

If the organization does not know that a high-risk AI system is being used, it cannot ensure use according to instructions.

It cannot assign human oversight.

It cannot assess input data relevance.

It cannot monitor operation.

It cannot retain relevant logs.

It cannot notify workers or affected persons.

It cannot escalate risk.

It cannot prove control.

System identification is therefore essential.

But it is only the first level.

Evidence that a system existed is not evidence that it was properly governed.

A register entry does not prove appropriate use.

A procurement record does not prove human oversight.

A vendor contract does not prove input data validity.

A risk assessment does not prove continued validity.

A deployment approval does not prove that actual use remained within intended purpose.

A system inventory can tell the organization what exists.

It cannot by itself show whether what exists is still being used correctly.

This distinction matters because many organizations may begin AI governance by building inventories.

That is a good starting point.

But the inventory must not be mistaken for governance maturity.

The register shows the system.

Evidence of control shows the system's governed operation.

A mature deployer should therefore treat the system register as the entry point into a wider evidence architecture.

Each AI system record should link to:

- provider instructions;
- internal use procedures;
- human oversight assignment;
- training records;
- data source documentation;
- monitoring records;
- log location;
- incident records;
- reassessment records;
- affected-person notification records;
- review history.

This turns the register from a static list into a governance map.

The system register should not only answer:

What AI systems do we have?

It should also help answer:

What evidence shows that each system is used under valid control conditions?

This is the first step from system inventory to evidence of control.

11.3 Evidence That a Process Was Followed

The second level of evidence is evidence that a process was followed.

This includes evidence that the organization followed provider instructions, completed internal reviews, trained users, assigned oversight, performed monitoring, retained logs, escalated incidents and notified affected parties where required.

This kind of evidence is stronger than evidence that a system exists.

It shows activity.

It shows that the organization created procedures and attempted to follow them.

Examples include:

- completed training records;
- signed policy acknowledgements;
- completed risk assessment forms;
- workflow approval records;
- human review records;
- monitoring reports;
- log retention records;
- incident tickets;
- escalation emails;
- meeting minutes;
- system review checklists;
- notification records;
- audit reports.

This evidence is important.

It can show that the deployer took Article 26 obligations seriously.

It can show that the organization did more than merely procure an AI system.

It can show that governance processes existed and were used.

But process evidence has limits.

A completed checklist does not always prove that the control was meaningful.

A training record does not prove that the user understood the system.

A human review record does not prove that the human had enough information to challenge the output.

A monitoring report does not prove that monitoring detected the right signals.

An incident ticket does not prove that escalation was timely.

A notification record does not prove that affected persons received meaningful transparency.

A log retention record does not prove that logs were useful.

This is the central weakness of process-based compliance.

It can produce evidence of activity without producing evidence of control.

For example, an organization may show that a human reviewed every AI-supported recommendation.

But if the human reviewer saw only the final score, had no access to the underlying evidence, and approved 99.9% of outputs within seconds, the evidence may show review activity but not meaningful oversight.

An organization may show that all users completed training.

But if the training was generic and did not explain the system's specific limitations, the evidence may show training completion but not competence for oversight.

An organization may show that input data checks were performed.

But if the checks did not include freshness, representativeness or missing critical fields, the evidence may show a process but not input validity.

An organization may show that logs were retained.

But if the logs cannot connect input, output, human review and downstream consequence, the evidence may show storage but not reconstructability.

For this reason, evidence that a process was followed must be evaluated for control value.

The deployer should ask:

What control was this process meant to create?

Did the process actually create that control?

Can the evidence show that the control operated?

Can the evidence show whether the control was effective?

Can the evidence show whether the control remained valid under current conditions?

This is how process evidence becomes evidence of control.

The key is to connect process completion to control purpose.

For example:

Training exists to create competent use and oversight.

Therefore, evidence should show not only completion, but system-specific competence.

Human review exists to create human control.

Therefore, evidence should show not only approval, but meaningful ability to challenge or intervene.

Monitoring exists to detect change and risk.

Therefore, evidence should show not only reports, but triggers, alerts and actions.

Logging exists to support reconstruction.

Therefore, evidence should show not only retention, but decision-trail usefulness.

Escalation exists to respond to risk.

Therefore, evidence should show not only a ticket, but action, decision and outcome.

This is the difference between process evidence and control evidence.

A mature deployer should not reject process evidence.

It should strengthen it.

The goal is not less process.

The goal is process that proves control.

11.4 Evidence That Control Conditions Were Valid

The third and strongest level of evidence is evidence that control conditions were valid when the AI-supported workflow became operational.

This is the core of evidence of control.

The question is no longer only:

Did the system exist?

Or:

Was a process followed?

The question becomes:

Were the conditions that allowed continuation still true?

This is the hardest question in AI governance.

It requires the deployer to understand not only the system and the process, but the state of the control environment at the relevant moment.

For high-risk AI systems, relevant control conditions may include:

- intended purpose;
- actual use;
- provider instructions;
- user competence;
- human oversight availability;
- human oversight authority;
- input data relevance;
- input data freshness;
- input data representativeness;
- model or system version;
- policy version;
- risk threshold;
- authority holder;
- approval state;
- monitoring status;
- warning state;
- escalation state;
- log integrity;
- downstream consequence;
- affected-person notification;
- workplace notification;
- boundary status.

A control condition is valid when it supports the workflow's continuation.

For example:

Human oversight is valid when the assigned person has competence, training, authority, time, information and practical ability to intervene before consequence attaches.

Input data is valid when it is relevant, sufficiently representative, sufficiently complete and fresh enough for the intended purpose and consequence level.

Authority is valid when the person, role or system approving continuation still has current permission to do so under the applicable policy and operational context.

Monitoring is valid when it detects relevant signals and connects them to action.

Logs are valid as evidence when they can reconstruct the AI-supported workflow and relevant control states.

Intended purpose is valid when actual use still fits within the provider's intended use and internal approval.

Notification is valid when the persons affected by current use have been informed in a way that matches actual system influence.

These conditions may change.

That is why evidence of continued validity matters.

A system approved six months ago may now be used differently.

A human overseer assigned at deployment may no longer be in the role.

A data source may have stopped refreshing.

A provider instruction may have changed.

A policy may have been updated.

A model may have been substituted.

A workflow may have become more automated.

A recommendation may now trigger downstream action.

A notice may no longer describe actual use.

A control may still exist in form, while no longer fitting the current context.

This is where many governance failures emerge.

The form persists.

The meaning changes.

The organization sees continuity.

But continuity may have become inertia.

Evidence of control should therefore help the organization ask:

Does this still belong here under current conditions?

This question applies to systems, controls, approvals, policies, oversight roles, data sources and workflows.

It shifts governance from existence to continued validity.

The deployer should not assume that because a control existed once, it remains valid now.

Instead, the deployer should preserve evidence that control conditions were current at the moment of use.

This may include:

- current provider instruction version;
- current policy version;
- current approval status;
- current data source status;
- current human overseer assignment;
- current training status;
- current monitoring status;
- current authority state;
- current workflow map;
- current downstream integration map;
- current risk classification;
- current notification coverage;
- current log integrity status.

This evidence should be linked to the specific workflow or case where possible.

The strongest form of evidence of control is not only periodic review.

It is runtime state capture.

Runtime state capture means preserving the relevant control state at the moment the AI-supported workflow moves toward consequence.

For example:

At the moment an AI-supported supplier recommendation is approved, the record should show the supplier data freshness, risk threshold, approving authority, human review, model used, output, warnings, and whether any escalation condition existed.

At the moment an AI-supported HR shortlist is generated, the record should show intended use, candidate data condition, system version, human reviewer, output influence, protected-characteristic safeguards, and review action.

At the moment an AI-supported clinical summary is used, the record should show source records included, freshness, missing data warnings, clinician review, system limitations and whether the output was modified.

At the moment a public-sector AI-supported recommendation affects a person, the record should show the legal or policy context, input case data, system role, human review, notification state and final decision pathway.

This is evidence that control conditions were valid.

It does not guarantee that the AI output was correct.

But it helps prove that the deployer operated the system under governed conditions.

That is the standard Article 26 pushes organizations toward.

11.5 Why Evidence of Control Is the Deployer's Core Challenge

Evidence of control is the deployer's core challenge because the deployer is responsible for the system as used.

The provider can supply the AI system, documentation, instructions, technical features and design controls.

But the deployer controls the operational environment.

The deployer controls actual use.

The deployer controls internal workflow integration.

The deployer controls user access.

The deployer controls input data where under its authority.

The deployer controls human oversight assignment.

The deployer controls monitoring response.

The deployer controls internal escalation.

The deployer controls whether AI output becomes operationally influential.

This makes the deployer responsible for proving control in practice.

The main difficulty is that high-risk AI systems operate in dynamic environments.

Static evidence may not be enough.

A deployment approval can become outdated.

A policy can become misaligned.

A human oversight role can become symbolic.

A data source can become stale.

A workflow can expand.

A recommendation can become a trigger.

A fallback model can change the authority condition.

A context window can include outdated assumptions.

A log can preserve activity but not meaning.

A notice can remain visible while actual use changes.

This means the deployer must manage evidence over time.

Evidence of control is not a one-time compliance package.

It is a continuing discipline.

It must support:

- initial deployment;
- operational use;
- monitoring;
- reassessment;
- incident response;
- audit;
- regulator inquiry;
- board oversight;
- affected-person challenge;
- internal learning;
- system retirement.

This is especially important because AI governance failures may not always look dramatic.

A workflow may fail silently.

A system may operate exactly as designed, but under conditions that no longer support use.

A dashboard may stay green.

A human may approve.

A log may record completion.

A decision may be made.

But the evidence, authority, data, policy, oversight or context may no longer justify continuation.

Evidence of control is the discipline that helps detect and prove this.

It enables the organization to ask:

Was the system used according to instructions?

Was human oversight meaningful?

Were input conditions valid?

Was monitoring active?

Were logs reconstructable?

Were risks escalated?

Were affected persons informed?

Was authority current?

Was the output allowed to influence consequence?

Was continuation justified?

Could the system have paused?

Did it pause when it should?

These questions are not theoretical.

They are the practical questions auditors, regulators, boards, courts, affected persons and internal governance bodies may ask after AI systems become embedded in high-risk workflows.

The organizations that can answer them will have a stronger governance position.

The organizations that cannot may discover that they had compliance documents but not evidence of control.

This is why Article 26 should be interpreted as more than a deployer checklist.

It should be interpreted as an evidence architecture requirement.

For each obligation, the deployer should identify:

What control is required?

Who owns it?

Where does it operate?

What evidence does it produce?

How is the evidence retained?

How is the evidence reviewed?

What trigger causes reassessment?

What happens if the control condition is missing?

This turns Article 26 into a practical operating model.

The deployer is no longer asking only:

What must we document?

The deployer is asking:

What must we be able to prove?

And more importantly:

What must our systems be able to know before an AI-supported workflow becomes consequential?

That is the future of deployer governance.

Not only compliance evidence.

Evidence of control.

Not only records of activity.

Records of validity.

Not only audit after execution.

Admissibility before consequence.

Chapter 11 Summary

Evidence of control is the core concept of this study.

It means evidence that the relevant control conditions were active, valid and inspectable when the AI-supported workflow became operational.

This chapter distinguishes three levels of evidence.

The first level is evidence that a system existed.

This includes inventories, registers, procurement records, system ownership, intended purpose and deployment records. It is necessary, but it does not prove that the system was governed.

The second level is evidence that a process was followed.

This includes training, reviews, monitoring reports, logs, incident tickets, human approvals and notification records. This is stronger, but still may not prove that controls were meaningful.

The third and strongest level is evidence that control conditions were valid.

This means showing that the system was used within intended purpose, input data was suitable, human oversight was meaningful, authority was current, monitoring was active, logs were reconstructable, escalation was possible and the workflow remained within valid boundaries when consequence attached.

This is the deployer's central challenge.

A high-risk AI system can remain technically functional while its governance conditions degrade.

A policy can remain visible while its assumptions disappear.

An approval can remain in place while actual use changes.

A human can remain in the loop while losing practical control.

A log can remain stored while failing to reconstruct meaning.

Evidence of control is what allows the deployer to detect and prove whether the system was still allowed to continue.

The central question is:

Can the organization prove that the AI system was used under valid control conditions when its output became operational?

This question is the bridge between Article 26 compliance and runtime governance.

It is also the foundation for the rest of this study.

Chapter 12

Runtime Governance

Legal anchor and framework boundary. [Article 26](#). ALLOW, HOLD, ESCALATE and REFUSE are NextOne Observatory runtime-governance states, not statutory categories.

12.1 What Runtime Governance Means

Runtime governance means the ability to observe, assess and control AI-supported workflows during operation, not only before deployment or after execution.

This distinction is essential.

Many organizations understand pre-deployment governance.

They review the system.

They assess risk.

They approve the vendor.

They classify the use case.

They assign ownership.

They create policies.

They conduct training.

They document intended purpose.

They prepare registers.

These activities are necessary.

Many organizations also understand post-event governance.

They perform audits.

They review logs.

They investigate incidents.

They respond to complaints.

They prepare reports.

They update controls.

These activities are also necessary.

But high-risk AI systems create a third governance requirement.

Governance during use.

This is runtime governance.

Runtime governance asks:

What is happening now?

Are the conditions that justified deployment still true?

Is the system still being used according to instructions?

Is the input data still relevant, representative and fresh enough?

Is human oversight currently available and meaningful?

Is the output still advisory, or has it become operationally decisive?

Has the workflow changed?

Has the system crossed a boundary?

Has authority changed?

Has evidence changed?

Has the risk context changed?

Should the workflow continue, hold, escalate or refuse?

This is the governance layer that sits between policy and consequence.

It is not enough to approve a high-risk AI system once and assume that the approval remains valid forever.

It is not enough to audit after the fact and hope that logs contain enough information.

The deployer must be able to detect whether the system is still operating under valid control conditions when its output becomes operational.

This is the heart of runtime governance.

Runtime governance does not mean that every AI system must be watched manually at every second.

It means that the organization has designed mechanisms to detect, record and respond when relevant control conditions change.

The control conditions may include:

- intended purpose;
- actual use;
- provider instructions;
- model or system version;
- input data condition;
- human oversight availability;
- authority state;
- policy version;
- monitoring status;
- log integrity;
- downstream consequence;
- escalation state;
- risk classification;
- affected-person information;
- workplace notification;
- system boundary.

A runtime governance system should know when these conditions are satisfied.

It should also know when they are not satisfied.

And perhaps most importantly, it should know when it cannot tell.

Unknown should not automatically mean continue.

In high-risk AI systems, uncertainty can be a governance state.

If the system cannot determine whether required evidence is fresh, whether human oversight is available, whether authority is current or whether actual use remains within intended purpose, the workflow should not silently proceed as if everything is valid.

The correct response may be HOLD.

This is why runtime governance is closely connected to the operational states introduced earlier in this study:

ALLOW.

HOLD.

ESCALATE.

REFUSE.

These states make governance actionable.

ALLOW means current control conditions support continuation.

HOLD means required conditions are incomplete or uncertain.

ESCALATE means higher-level review is required.

REFUSE means the workflow should not proceed under current conditions.

Runtime governance is the discipline that determines when each state applies.

It turns Article 26 from a set of deployer obligations into a live operating model.

12.2 Runtime Governance Versus Policy Governance

Policy governance and runtime governance are different, but they must work together.

Policy governance defines what should happen.

Runtime governance detects whether what should happen is still happening.

Policy governance creates rules.

Runtime governance checks conditions.

Policy governance assigns responsibility.

Runtime governance observes whether responsibility can be exercised.

Policy governance states that human oversight is required.

Runtime governance checks whether human oversight is available before consequence attaches.

Policy governance states that input data must be relevant.

Runtime governance checks whether the input data is currently valid for the workflow.

Policy governance states that use must remain within intended purpose.

Runtime governance checks whether actual use has drifted.

Policy governance states that logs must be retained.

Runtime governance checks whether logs are being generated, preserved and connected to the decision trail.

Policy governance states that serious risks must be escalated.

Runtime governance detects when escalation triggers appear.

This difference matters because organizations often assume that a policy creates control.

It does not.

A policy creates a standard.

A control is the mechanism that makes the standard operational.

For example, a policy may say:

AI outputs must be reviewed by a human before use in high-consequence decisions.

That policy is important.

But runtime governance asks:

Is the current workflow high-consequence?

Was human review required?

Was the human reviewer available?

Did the reviewer see the relevant evidence?

Did the reviewer have authority?

Did the review occur before consequence attached?

Was the review recorded?

Did the reviewer approve, modify, reject or escalate?

Could the workflow proceed without review?

These questions determine whether the policy became a control.

Another example:

A policy may say:

AI systems must be used only within approved use cases.

Runtime governance asks:

What use case is active now?

Does it match the approved use case?

Has a new user group started using the system?

Has the output been connected to a new downstream workflow?

Has the recommendation become an automation trigger?

Does current use still fit the provider's intended purpose?

If not, should the workflow hold pending reassessment?

This is where runtime governance adds value.

It prevents a policy from becoming disconnected from operational reality.

Policy governance is necessary for accountability, consistency and organizational direction.

But policy governance can become stale.

A policy may remain visible while the system's use changes.

A policy may remain approved while the workflow expands.

A policy may remain current in the document library while users behave differently.

A policy may define oversight while practical intervention becomes impossible under volume pressure.

Runtime governance helps detect this gap.

It asks whether the policy still belongs in the current context.

This is not a rejection of policy.

It is a way of making policy real.

A mature deployer should therefore connect each Article 26 policy obligation to runtime signals.

For use according to instructions, the runtime signals may include workflow identity, actual use, user group, data source and output consequence.

For human oversight, the runtime signals may include reviewer availability, reviewer authority, review completion, override action and escalation.

For input data, the runtime signals may include freshness, source status, completeness and representativeness indicators.

For monitoring, the runtime signals may include drift, anomaly, performance, user reliance and boundary change.

For logs, the runtime signals may include log completeness, traceability and decision-trail integrity.

For escalation, the runtime signals may include risk triggers, serious incident indicators, user reports and system warnings.

Policy governance tells the organization what must be true.

Runtime governance checks whether it is still true.

That is the core difference.

12.3 Control States and Operational States

A runtime governance model must distinguish between operational states and control states.

Operational states describe what the AI-supported workflow is doing.

Control states describe whether the workflow is allowed to continue.

This distinction is important because a workflow can be operationally successful while governance control conditions are weak.

A system may process the case.

A model may produce the output.

A workflow may route the task.

A user may approve.

A downstream system may accept the result.

An external action may occur.

Operationally, the workflow succeeded.

But from a governance perspective, it may not have been valid.

The input data may have been stale.

The reviewer may not have had authority.

The use case may have drifted outside intended purpose.

The policy version may have changed.

The output may have triggered an action beyond the approved scope.

The log may not preserve enough context.

The human overseer may have been unavailable.

The system may have used a fallback model without authority reassessment.

This is why runtime governance must evaluate control states separately from technical execution.

An operational state may include:

- draft created;
- output generated;
- recommendation produced;
- score calculated;
- case routed;
- alert created;
- tool called;
- summary generated;
- task completed;
- decision submitted;
- system updated;
- notification sent.

A control state may include:

- input valid;
- oversight satisfied;
- authority current;
- policy matched;
- use within intended purpose;
- logs complete;
- monitoring normal;
- escalation not required;
- consequence level acceptable;
- boundary intact;
- evidence sufficient;
- continuation allowed.

A mature deployer should not confuse these two categories.

The workflow should not continue merely because the next operational step is technically available.

It should continue only if the control state permits continuation.

This principle is especially important for agentic AI systems.

Agentic workflows may involve multiple intermediate steps.

A single workflow may retrieve data, call tools, generate reasoning, interact with external systems, request approvals, update records and trigger downstream actions.

At each step, the operational state may advance.

But control states may change during the workflow.

For example:

At step one, the data is fresh.

At step two, the data source fails.

At step three, the system uses cached context.

At step four, a model fallback occurs.

At step five, the output becomes a recommendation.

At step six, a downstream automation treats the recommendation as a trigger.

At step seven, a human approval is assumed but not actually recorded.

If the workflow is governed only at the start, the organization may miss the control state changes.

Runtime governance must therefore support control-state evaluation throughout the workflow.

Not every step requires the same level of control.

But the system should identify critical transition points.

Critical transition points may include:

- input data retrieval;
- context assembly;
- model selection;
- output generation;
- confidence or uncertainty detection;
- human review request;
- approval;
- override;
- escalation;
- tool execution;
- downstream system write;

- external communication;
- final decision;
- case closure.

At each critical transition point, the governance system should ask:

Is continuation allowed?

If yes, the workflow may proceed.

If no, the workflow should stop or refuse.

If uncertain, the workflow should hold.

If higher authority is required, the workflow should escalate.

This is the practical meaning of runtime control.

The system is not only executing.

It is checking whether execution remains admissible.

12.4 Runtime Visibility and Observability

Runtime governance depends on observability.

An organization cannot control what it cannot see.

But observability in high-risk AI governance means more than system monitoring.

It means visibility into the conditions that determine whether the AI-supported workflow remains valid.

Traditional observability may focus on technical metrics:

- uptime;
- latency;
- error rates;
- request volume;
- throughput;
- memory usage;
- API status;
- infrastructure health;
- cost;
- token usage.

These metrics are useful.

But they do not answer the key governance question:

Is this workflow still allowed to continue?

To answer that, the deployer needs governance observability.

Governance observability may include:

- system identity;
- model version;
- intended purpose;
- actual use;
- user role;
- authority state;
- input data freshness;
- input data completeness;
- evidence source;
- policy version;
- human oversight status;
- override status;
- escalation status;
- monitoring alert state;
- log integrity;
- downstream consequence;
- boundary status;
- notification status.

These are decision-grade signals.

They do not merely describe system health.

They describe control health.

This distinction is important.

A green infrastructure dashboard may show that the system is running.

A green governance dashboard would need to show that the system is running within valid boundaries.

These are not the same.

A high-risk AI system may be technically healthy and governance-unhealthy at the same time.

For example:

The API is available, but the data source is stale.

The model is responsive, but the output is being used outside intended purpose.

The dashboard is green, but human oversight is unavailable.

The workflow is completing, but logs are incomplete.

The agent is functioning, but the permission surface expanded after a plugin update.

The system is stable, but downstream automation now attaches consequence to previously advisory output.

Runtime visibility must reveal these conditions.

This is why governance observability should be designed around control questions.

A deployer should be able to see:

Which high-risk AI systems are active?

Which workflows are they affecting?

Which outputs are becoming consequential?

Which systems have stale input data?

Which systems are operating under outdated policy versions?

Which workflows require human review?

Which reviews are overdue?

Which outputs were overridden?

Which use cases are expanding?

Which systems have unresolved monitoring alerts?

Which logs are incomplete?

Which downstream integrations changed?

Which systems are in HOLD, ESCALATE or REFUSE state?

This does not mean every organization needs an advanced AI governance platform on day one.

But it does mean that deployers should develop visibility appropriate to the risk.

At minimum, organizations should know where their high-risk AI systems are, how they are used and whether critical control conditions are being monitored.

As systems become more consequential, more automated or more agentic, observability must become stronger.

The purpose of observability is not only to satisfy audit.

It is to support timely intervention.

If a human overseer cannot see that data is stale, they cannot intervene intelligently.

If a governance owner cannot see that use has drifted, they cannot reassess.

If a compliance officer cannot see that a policy version changed, they cannot evaluate continued validity.

If a technical team cannot see that a fallback model was used, they cannot evaluate authority continuity.

If a board cannot see the state of high-risk AI control, it cannot govern risk effectively.

Observability turns hidden conditions into visible signals.

Runtime governance turns visible signals into action.

Both are needed.

12.5 Why High-Risk AI Requires Live Control Awareness

High-risk AI requires live control awareness because the risk is not fixed at deployment.

The risk changes as the system is used.

It changes when data changes.

It changes when workflows change.

It changes when people change roles.

It changes when downstream systems change.

It changes when the model changes.

It changes when policy changes.

It changes when a recommendation becomes a trigger.

It changes when a human reviewer becomes overloaded.

It changes when the output becomes more trusted.

It changes when users stop questioning the system.

It changes when the system is applied to new cases, populations or contexts.

This is why a deployment approval is not enough.

Approval is a point-in-time decision.

High-risk AI use is an ongoing condition.

The gap between the two is where runtime governance is needed.

Live control awareness does not mean constant manual supervision.

It means the organization has enough current information to know whether the system remains within acceptable control boundaries.

This awareness may be automated, semi-automated or human-supported depending on risk.

For lower-risk internal workflows, periodic review may be sufficient.

For high-risk workflows affecting rights, safety, employment, finance, public services or healthcare, stronger monitoring and state awareness may be required.

For agentic systems connected to tools, downstream workflows or external actions, live control awareness may become essential.

The reason is simple:

The point of consequence may occur after the original output.

An AI system may generate a recommendation.

A human may approve it.

A downstream system may act on it.

A record may be updated.

A person may be notified.

A case may be closed.

A benefit may be denied.

A supplier may be selected.

A patient pathway may be affected.

A worker may be evaluated.

A financial decision may be influenced.

The governance question is not only whether the AI output was produced correctly.

The question is whether the system was still allowed to influence that consequence.

This requires awareness of current conditions at the point where consequence attaches.

In traditional governance, the focus is often before and after:

before deployment,

after audit,

before approval,

after incident.

Runtime governance adds the during:

during use,

during workflow transition,

during escalation,

during tool execution,

during human review,

during downstream propagation,

during consequence formation.

This is the point where governance becomes operational.

The deployer should therefore identify consequence points.

A consequence point is a stage where the AI-supported workflow begins to affect a person, process, right, obligation, operational action or regulated decision.

Examples include:

- candidate shortlist finalized;
- loan recommendation sent to decision-maker;
- clinical summary entered into patient record;
- supplier recommendation approved;

- customer refund decision communicated;
- public benefit case routed;
- risk alert escalated;
- access permission changed;
- automated task created;
- external notification sent;
- case closed.

At each consequence point, the deployer should be able to show that relevant control conditions supported continuation.

This does not always require a separate manual review.

But it does require an evidence model.

The evidence model should answer:

What AI output influenced the consequence?

What evidence supported the output?

Was input data current?

Was the system used within intended purpose?

Was human oversight required?

Was oversight completed?

Was authority current?

Were there warnings?

Were logs complete?

Was escalation required?

Why was continuation allowed?

This is the core of live control awareness.

It does not prevent organizations from using AI.

It allows them to use AI responsibly.

The future of high-risk AI governance will not be defined only by whether systems are deployed.

It will be defined by whether organizations can know, at the right time, when AI-supported workflows remain valid and when they should stop.

Runtime governance is the mechanism for that knowledge.

Chapter 12 Summary

Runtime governance means the ability to observe, assess and control AI-supported workflows during operation, not only before deployment or after execution.

It is the governance layer between policy and consequence.

This chapter distinguishes runtime governance from policy governance.

Policy governance defines what should happen.

Runtime governance checks whether what should happen is still happening under current conditions.

A policy may require human oversight, but runtime governance checks whether oversight is available, competent, informed and able to intervene.

A policy may require input data relevance, but runtime governance checks whether the input data is currently suitable.

A policy may require use within intended purpose, but runtime governance checks whether actual use has drifted.

A policy may require logging, but runtime governance checks whether logs can reconstruct the decision trail.

The chapter also distinguishes operational states from control states.

An operational state describes what the workflow is doing.

A control state describes whether the workflow is allowed to continue.

A workflow can be technically successful while governance control conditions are weak.

This is why runtime governance requires control states such as ALLOW, HOLD, ESCALATE and REFUSE.

The chapter then explains the role of runtime visibility and governance observability.

Traditional observability focuses on technical health.

Governance observability focuses on control health: system identity, intended purpose, actual use, input data condition, authority state, human oversight, monitoring alerts, log integrity and downstream consequence.

Finally, the chapter argues that high-risk AI requires live control awareness because risk changes during operation.

The deployer must know not only that an AI system was approved at deployment, but whether the system remains valid at the point where output becomes consequence.

The central runtime governance question is:

Do current control conditions still support continuation?

If yes, continue.

If no, stop or refuse.

If uncertain, hold and reassess.

This is how Article 26 becomes operational AI governance.

Chapter 13

Authority, Scope and Operational Boundaries

Legal anchor and framework boundary. [Articles 13-14](#); [Article 26](#). Authority, scope and boundary objects are operational controls proposed by the framework.

13.1 Authority as a Runtime Condition

Authority is one of the most important but least visible control conditions in AI governance.

Many governance discussions focus on accuracy, bias, transparency, explainability, monitoring and logs.

These are all essential.

But for deployers of high-risk AI systems, another question is just as important:

Who or what was allowed to act?

This is an authority question.

Authority determines whether an output, recommendation, decision, approval, escalation, workflow transition or system action is permitted to become operational.

In traditional organizations, authority may be expressed through job roles, policies, delegation frameworks, approval matrices, legal mandates, access permissions, procurement limits, clinical responsibility, compliance sign-off, board authority or regulatory powers.

In AI-supported workflows, authority becomes more complex.

The AI system may not formally hold authority.

But its output may influence the person who does.

The system may not make the decision.

But it may shape the evidence environment in which the decision is made.

The model may not approve a supplier.

But it may create the recommendation that becomes the default approval path.

The system may not reject a job applicant.

But it may rank the applicant low enough that no human considers them.

The agent may not formally execute a regulated decision.

But its output may trigger downstream automation that creates operational consequence.

This is why authority must be treated as a runtime condition.

It is not enough to know who had authority on paper.

The deployer must know whether the authority attached to the AI-supported workflow was valid at the moment the workflow moved toward consequence.

Authority can expire.

Authority can drift.

Authority can be delegated incorrectly.

Authority can be assumed.

Authority can be transferred silently.

Authority can be embedded in workflow design.

Authority can be created by default settings.

Authority can be amplified by automation.

Authority can be reduced by model fallback.

Authority can become unclear when AI output moves between systems.

This means authority is not only a legal or organizational concept.

It is also a systems concept.

A runtime governance system should be able to represent authority states.

For example:

- user has authority to view output;
- user has authority to approve;
- user has authority to override;
- user has authority to escalate;
- user has authority to stop workflow;
- AI system has permission to recommend;
- AI system has permission to draft;
- AI system does not have permission to execute;
- downstream system may not treat recommendation as trigger;

- model fallback reduces authority;
- stale evidence suspends authority;
- missing human review blocks authority;
- use outside intended purpose invalidates authority.

These states should not remain implicit.

For high-risk AI systems, authority should be visible, inspectable and connected to workflow control.

The central question is:

Was the workflow still authorized to continue under current conditions?

This is different from asking whether someone once approved the system.

A system may have been approved for deployment.

But a specific workflow instance may still lack valid authority.

For example:

A procurement AI system may be approved for supplier triage, but not for final supplier selection.

A HR AI system may be approved for interview support, but not for candidate ranking.

A clinical AI system may be approved for summarization, but not for diagnosis.

A compliance AI system may be approved for drafting, but not for case closure.

A public-sector AI system may be approved for internal prioritization, but not for eligibility decisions.

In each case, the system may exist within an approved environment, but the specific use may exceed the authorized scope.

This is why authority must be checked at the point of operational transition.

The workflow should not ask only:

Can the system produce this output?

It should ask:

Is this output allowed to influence this consequence?

If the answer is yes, continuation may be allowed.

If the answer is no, continuation should be refused.

If the answer is unclear, the workflow should hold or escalate.

This turns authority into a runtime control condition.

13.2 Who or What Was Allowed to Act?

In AI-supported workflows, action is not always obvious.

An action may be a visible execution step, such as sending an email, approving a transaction, updating a database, rejecting an application or triggering a workflow.

But action can also be indirect.

A system can act by ranking.

A system can act by prioritizing.

A system can act by selecting evidence.

A system can act by summarizing.

A system can act by recommending.

A system can act by flagging risk.

A system can act by routing cases.

A system can act by shaping what a human sees.

A system can act by creating a default.

A system can act by narrowing human attention.

This matters because authority should follow influence, not only formal execution.

If an AI system materially shapes a decision environment, then governance should ask whether that influence was authorized.

A deployer should therefore map the forms of action within an AI-supported workflow.

A practical action map may include:

- viewing information;
- retrieving data;
- interpreting data;
- ranking persons, cases or options;
- scoring risk;

- generating summaries;
- producing recommendations;
- drafting decisions;
- routing workflows;
- triggering tasks;
- requesting approvals;
- updating records;
- communicating externally;
- initiating downstream automation;
- finalizing decisions.

Each action type may require a different authority level.

For example:

The authority required to draft a recommendation may be lower than the authority required to send it externally.

The authority required to summarize internal evidence may be lower than the authority required to use the summary as the basis for a decision.

The authority required to flag a case for review may be lower than the authority required to escalate the case to enforcement.

The authority required to rank options may be lower than the authority required to automatically select one.

The authority required to assist a clinician may be lower than the authority required to alter a clinical pathway.

The authority required to suggest a supplier may be lower than the authority required to approve that supplier.

A deployer should therefore avoid treating all AI outputs as equivalent.

The same output may carry different authority depending on how the organization uses it.

A recommendation shown to a human expert may be one thing.

The same recommendation used as a default approval route may be another.

The same recommendation sent to a downstream automation engine may be something else entirely.

This is why authority mapping must include downstream effects.

The AI system itself may not execute the final action.

But its output may enter a system that does.

For example:

An AI tool generates a compliance recommendation.

The recommendation is entered into a workflow system.

The workflow system assigns a case outcome.

The case outcome triggers customer notification.

The notification creates external consequence.

Where did authority attach?

At the AI output?

At the workflow assignment?

At the human approval?

At the automated notification?

At the final case closure?

A mature deployer should be able to answer this.

The organization should know where AI influence becomes operational authority.

This is especially important for agentic AI systems.

Agentic systems may perform multi-step operations and interact with tools. In such systems, authority should be granular.

The agent may be allowed to read but not write.

It may be allowed to draft but not send.

It may be allowed to recommend but not approve.

It may be allowed to call internal search tools but not external systems.

It may be allowed to prepare a workflow but not trigger it.

It may be allowed to escalate to a human but not bypass human review.

These permissions should be explicit.

They should also be monitored.

A permission boundary that exists at deployment may expand later through integration changes, plugin updates, workflow redesign or user configuration.

If authority expands silently, the system's risk profile changes.

This is why the deployer should preserve evidence of authority state.

For each consequential workflow, the evidence should show:

- who initiated the workflow;
- what the AI system was allowed to do;
- what tools were available;
- what permissions were active;
- what human approvals were required;
- what authority the human reviewer held;
- whether the output was advisory or executable;
- whether downstream systems acted on the output;
- whether any authority boundary changed;
- why continuation was allowed.

The core question is not only:

Who made the final decision?

The stronger question is:

Who or what shaped the path to consequence, and under what authority?

13.3 Scope of Use and Scope of Consequence

Scope is one of the most important governance boundaries for high-risk AI systems.

A system may be approved for a specific scope of use.

But its scope of consequence may become wider.

This distinction matters.

Scope of use describes where and how the system is intended or approved to be used.

Scope of consequence describes what the system's output actually affects.

The two are not always the same.

For example:

A system may be approved for internal triage.

But its outputs may later influence final decisions.

A system may be approved for drafting.

But the drafts may become the main reasoning used in official records.

A system may be approved for recommendation.

But recommendations may become default approvals.

A system may be approved for internal analytics.

But the analytics may later shape resource allocation.

A system may be approved for low-value procurement.

But the same workflow may later be used for high-value suppliers.

A system may be approved for assisting trained specialists.

But the output may later be used by non-specialists.

This is a scope problem.

The deployer may believe the system remains within its approved use, while the consequences attached to the system have changed.

This is why scope must be monitored at runtime.

A deployment approval should define:

- approved use case;
- approved user group;
- approved data sources;
- approved workflow;
- approved output role;
- approved consequence level;
- approved human oversight model;
- approved downstream integrations;
- prohibited uses;
- escalation triggers.

But this approval should not remain static.

The deployer should periodically and operationally compare approved scope with actual scope.

Actual scope may change when:

- new teams start using the system;
- users apply the system to new case types;
- new data sources are added;
- output is connected to new workflows;
- automation is built around the output;
- decision-makers rely more heavily on the system;
- human review is reduced;
- system output becomes part of official records;
- use expands to more affected persons;
- the consequence level increases.

Scope drift is dangerous because it often looks like successful adoption.

A system that spreads across the organization may appear valuable.

A tool that users rely on may appear mature.

A workflow that becomes faster may appear efficient.

But if the system's influence expands beyond the approved scope, governance must reassess.

The deployer should not treat scale as automatic validation.

Scale can amplify risk.

This is especially true where AI output affects people.

A system used for small internal analysis may create limited harm if wrong.

The same system used across thousands of employment, finance, healthcare, education or public-service cases may create systemic impact.

This means scope of consequence should be part of the evidence of control.

The deployer should know:

What decisions can the output influence?

How many persons or cases may be affected?

What is the severity of potential impact?

Can the output affect rights, access, safety, opportunity or legal position?

Does the output remain advisory?

Can the output trigger action?

Can the output be challenged?

Can the workflow be reversed?

Is human review still proportionate to consequence?

If scope of consequence increases, control strength should increase.

A low-consequence recommendation may require lighter oversight.

A high-consequence recommendation may require stronger human review, more complete evidence, stricter logging, escalation triggers and possible refusal conditions.

This creates a practical principle:

Control should follow consequence.

The deployer should not govern the AI system only according to what it was originally called.

It should govern the system according to what it actually affects.

13.4 Boundary Expansion and Hidden Authority Drift

Boundary expansion occurs when an AI system moves beyond its original operational limits.

Authority drift occurs when the system's output gains more practical power than originally intended.

Together, they create one of the most serious risks in AI deployment.

This risk is especially difficult because it can be gradual and invisible.

At deployment, the system may have narrow boundaries.

It may assist.

It may recommend.

It may summarize.

It may prioritize.

It may draft.

It may not decide.

It may not execute.

It may not trigger downstream systems.

It may not communicate externally.

But over time, the boundaries may expand.

A recommendation becomes a default.

A draft becomes final unless edited.

A summary becomes the main evidence.

A risk flag becomes an escalation trigger.

A ranking becomes a shortlist.

A score becomes a threshold.

A workflow integration turns output into action.

A plugin gives the agent access to new tools.

A user grants broader permissions.

A downstream system begins treating output as structured input.

At no single point does the organization announce:

We have transferred authority to the AI system.

But that may be the practical effect.

This is hidden authority drift.

The formal authority structure remains the same.

The organization may still say that humans decide.

The policy may still say that AI is advisory.

The system description may still say decision support.

But in practice, AI output may carry operational authority.

This matters because Article 26 deployer obligations are not satisfied by labels.

Calling a system advisory does not make it advisory if the workflow treats its output as decisive.

Calling a human responsible does not create human control if the human cannot realistically challenge the system.

Calling a system read-only does not eliminate risk if its output triggers action elsewhere.

The deployer must govern practical influence.

Hidden authority drift can happen in several ways.

First, through user reliance.

Humans may gradually trust the system more and challenge it less.

Second, through workflow design.

The interface may make acceptance easier than rejection.

Third, through automation.

Downstream systems may begin acting on AI output.

Fourth, through organizational pressure.

Speed and efficiency may reward following the system.

Fifth, through lack of evidence.

Humans may not have enough information to challenge the output.

Sixth, through system integration.

The AI tool may gain access to more data, tools or workflows.

Seventh, through model substitution.

A fallback model may produce output under different reliability assumptions, but the workflow may preserve full authority.

Each of these changes can increase authority without formal governance approval.

A mature deployer should therefore monitor authority drift.

Useful signals include:

- reduced override rates;
- increased automatic acceptance;
- reduced review time;
- more decisions matching AI output;
- output used in official reasoning;
- new downstream integrations;
- increased user dependence;
- fewer escalations;

- workflow changes that bypass review;
- AI output triggering action;
- model fallback events;
- permission changes;
- new tool access;
- output used outside original scope.

These signals should trigger reassessment.

The deployer should ask:

Has the AI output become more influential than originally intended?

Has an advisory tool become a practical decision tool?

Has a human review step become symbolic?

Has the system gained access to new tools or workflows?

Has the consequence level changed?

Has the authority state changed?

Does the current control model still fit?

If the answer is uncertain, the workflow should not assume continuity.

It should hold or escalate.

This is the practical protection against hidden authority drift.

The system should not silently inherit authority from a context that no longer applies.

13.5 Evidence That Authority Remained Valid

Authority is not sufficient unless it can be evidenced.

A deployer should be able to show that the relevant authority state was valid when the AI-supported workflow became operational.

This requires more than identifying a responsible person.

It requires evidence that authority existed, applied to the specific workflow, remained current and supported the action taken.

Evidence of authority may include:

- role assignment records;
- approval matrices;
- delegation records;
- policy versions;
- access permissions;
- workflow permissions;
- system configuration;
- model authority level;
- human reviewer authority;
- escalation authority;
- override authority;
- control gate decisions;
- audit logs;
- decision records;
- downstream action records;
- authority change history.

The purpose is to reconstruct why the workflow was allowed to continue.

For example:

If a human reviewer approved an AI-supported decision, the organization should be able to show that the reviewer had authority to approve that type of decision at that time.

If an AI system generated a recommendation, the organization should be able to show whether the recommendation was allowed to influence the relevant decision.

If an AI agent called a tool, the organization should be able to show that the tool permission was valid for the workflow.

If a downstream automation acted on AI output, the organization should be able to show that this automation was approved and within scope.

If a model fallback occurred, the organization should be able to show whether the replacement model preserved, reduced or removed authority to continue.

If a workflow entered a higher-consequence stage, the organization should be able to show that required additional authority was obtained.

Authority evidence should also capture boundary decisions.

For example:

ALLOW: authority valid; workflow may continue.

HOLD: authority unclear or incomplete; workflow pauses.

ESCALATE: higher authority required.

REFUSE: authority does not support continuation.

These states should be logged.

The strongest evidence is not only that a final approval occurred.

The strongest evidence shows the authority chain.

An authority chain connects:

- system permission;
- user role;
- provider instruction boundary;
- intended purpose;
- workflow scope;
- human review;
- policy authority;
- downstream consequence;
- final decision.

This chain helps answer:

Who or what had permission to move the workflow from one state to another?

Without this chain, the organization may struggle to prove that AI-supported consequence was authorized.

This is especially important in regulated environments.

In financial services, authority may relate to approval limits, risk thresholds, compliance sign-off or regulated decision pathways.

In healthcare, authority may relate to clinical responsibility, professional judgment and patient safety.

In employment, authority may relate to hiring, promotion, termination, evaluation or worker monitoring.

In public services, authority may relate to legal mandate, administrative discretion and affected-person rights.

In procurement, authority may relate to budget thresholds, supplier risk, sanctions checks and contractual approval.

In critical infrastructure, authority may relate to operational safety and change control.

Each domain has its own authority structure.

The deployer must map the AI system into that structure.

This is why authority cannot be generic.

Authority must be specific to the workflow, consequence and domain.

The deployer should also preserve evidence that authority was reassessed when conditions changed.

Authority may need reassessment when:

- actual use changes;
- consequence level increases;
- data source changes;
- policy changes;
- provider instructions change;
- human overseer changes;
- downstream integration changes;
- model fallback occurs;
- incident signal appears;
- system output becomes more influential;
- new affected persons are involved;
- regulatory expectations change.

This connects authority to continued validity.

The question is not only:

Was authority granted?

The question is:

Was authority still valid under current conditions?

This question should be central to Article 26 governance.

A high-risk AI system should not be allowed to proceed simply because authority existed somewhere in the organization.

The authority must apply to the specific AI-supported workflow and consequence.

Where authority cannot be shown, continuation should be questioned.

Chapter 13 Summary

Authority, scope and operational boundaries are central to deployer governance.

Authority determines who or what is allowed to move an AI-supported workflow toward consequence.

Scope determines where and how the system is allowed to be used.

Operational boundaries determine the conditions under which use remains valid.

This chapter argues that authority should be treated as a runtime condition.

It is not enough to know that a system was approved at deployment or that a human is formally responsible.

The deployer must be able to show whether authority remained valid when the AI-supported workflow became operational.

The chapter distinguishes scope of use from scope of consequence.

A system may be approved for one use but later affect broader or more consequential workflows.

When consequence expands, control strength should also increase.

The chapter also explains hidden authority drift.

This occurs when AI output gains practical power without formal transfer of authority.

A system may remain labelled as advisory while becoming decisive in practice.

A human may remain formally responsible while losing practical control.

A read-only system may still create consequence if its output triggers downstream action.

For this reason, deployers should monitor authority drift, boundary expansion and scope change.

Signals may include reduced override rates, increased automatic acceptance, new downstream integrations, permission expansion, output used as official reasoning, model fallback events and workflow changes that bypass review.

Finally, the chapter defines evidence that authority remained valid.

A mature deployer should preserve the authority chain: system permission, user role, intended purpose, workflow scope, human review, policy authority, downstream consequence and final decision.

The central question is:

Was the workflow still authorized to continue under current conditions?

If yes, continuation may be allowed.

If no, the workflow should stop or refuse.

If unclear, it should hold or escalate.

Article 26 governance therefore requires more than human presence, system approval or policy language.

It requires evidence that authority, scope and boundaries remained valid when consequence attached.

Chapter 14

Evidence Freshness and Decision Validity

Legal anchor and framework boundary. [Articles 13 and 26](#). Evidence freshness and continued validity are operational concepts used to test whether prior approval remains usable under current conditions.

14.1 Why Freshness Matters

Evidence freshness is one of the most important practical issues in high-risk AI governance.

An AI-supported workflow may appear valid because it uses evidence.

But the deeper question is whether that evidence is still current enough to support the consequence attached to the output.

This distinction matters because evidence does not remain valid forever.

A supplier record may become outdated.

A policy may change.

A risk threshold may be updated.

A customer status may shift.

A patient record may receive new information.

A worker's role may change.

A financial indicator may deteriorate.

A fraud flag may be added.

A data source may stop refreshing.

A human approver may lose authority.

A provider instruction may be updated.

A model may be substituted.

A downstream workflow may change the effect of an output.

In each case, the AI system may still function.

The workflow may still complete.

The output may still look plausible.

The logs may still show normal activity.

The human reviewer may still approve.

But the evidence environment may no longer support continuation.

This is why freshness matters.

Freshness is not only a property of data.

It is a property of decision validity.

A decision, recommendation or workflow transition can only be valid if the evidence supporting it is sufficiently current for the purpose and consequence involved.

For high-risk AI systems, the question is not simply:

Did the system use evidence?

The stronger question is:

Was the evidence still fresh enough to justify the authority attached to the output?

Freshness should therefore be treated as a runtime control condition.

If evidence is fresh, continuation may be allowed.

If evidence is stale, continuation may require review.

If evidence freshness cannot be determined, the workflow should not automatically proceed.

The correct state may be HOLD.

This is especially important because stale evidence often creates silent failure.

A stale record does not always trigger an error.

A stale data source may still return data.

A stale policy may still appear in the workflow.

A stale approval may still exist in the system.

A stale supplier profile may still look complete.

A stale human oversight assignment may still appear valid in records.

The problem is not that evidence is absent.

The problem is that evidence is present but no longer reliable enough for the current decision.

This is more dangerous than obvious missing data.

Missing data may be visible.

Stale data may look legitimate.

That legitimacy can create false confidence.

This is why deployers should distinguish between evidence presence and evidence freshness.

Presence asks:

Is there evidence?

Freshness asks:

Is the evidence still valid for this use, at this time, under these conditions?

Article 26 deployer governance requires this second question.

A deployer cannot prove responsible use of a high-risk AI system if it cannot show that the evidence supporting use remained current enough when the system was used.

This applies to input data.

It applies to human oversight.

It applies to authority.

It applies to policies.

It applies to logs.

It applies to monitoring.

It applies to intended purpose.

It applies to risk classification.

Freshness is therefore a general governance property.

Not only a data-quality property.

14.2 Stale Evidence as Silent Failure

Stale evidence is one of the clearest examples of silent failure in AI governance.

A silent failure occurs when a workflow completes successfully even though the conditions required for safe or valid continuation are no longer present.

There may be no system crash.

No red alert.

No obvious exception.

No failed API call.

No visible policy breach.

No user complaint at the moment of execution.

The workflow simply continues.

That is what makes stale evidence dangerous.

The AI system may produce an output based on records that were once valid.

A human may review the output without knowing the records are outdated.

A downstream process may treat the output as reliable.

The organization may retain logs showing that the workflow completed.

But the logs may not reveal that the evidence was stale enough to undermine the decision.

Examples show the problem clearly.

A procurement AI system recommends a supplier based on financial data that was current last week, but the supplier's financial position changed yesterday.

A HR AI system supports an internal mobility decision based on role data that does not include a recent manager change or performance review.

A financial services AI system assesses risk based on an outdated policy threshold.

A public-sector AI system supports eligibility review based on case records that do not include a newly submitted document.

A clinical AI assistant summarizes patient history without the most recent lab result.

A customer support agent recommends a refund response without seeing a fraud flag added after the context was retrieved.

A compliance AI tool drafts a case assessment using a previous version of regulatory guidance.

In each example, the issue is not simply that the system lacked data.

The issue is that the workflow relied on data that appeared usable but no longer supported the level of confidence or authority attached to the output.

This creates a specific governance risk:

The system may inherit confidence from a past context.

That inherited confidence may no longer be justified.

Stale evidence therefore undermines decision validity.

A decision based on stale evidence may not be invalid in every case.

Some evidence remains useful over time.

Some records change slowly.

Some decisions tolerate older background context.

But high-consequence workflows require explicit freshness judgment.

The deployer should not assume that all evidence remains equally valid.

The organization should define freshness expectations based on:

- type of evidence;
- volatility of the evidence;
- intended purpose;
- consequence level;
- affected persons;
- regulatory context;
- risk sensitivity;
- availability of updated sources;
- possibility of harm;
- reversibility of decision.

For example:

A public policy version may remain valid until replaced.

A supplier financial risk score may require frequent refresh.

A sanctions check may require very current evidence.

A clinical test result may require current episode relevance.

A worker role assignment may require event-based updates.

A risk threshold may require version control.

A customer status may require real-time or near-real-time checks.

The key principle is proportionality.

The higher the consequence, the stronger the freshness requirement should be.

A low-risk internal suggestion may tolerate less current evidence.

A high-risk workflow affecting employment, healthcare, finance, essential services, public administration or safety should require stronger freshness proof.

Stale evidence becomes especially dangerous when it combines with automation.

If stale evidence feeds a recommendation, and that recommendation becomes a downstream trigger, the stale context can propagate quickly.

The original problem may be small.

The downstream consequence may be large.

This is why deployers should monitor evidence freshness before consequence attaches.

The system should not only ask:

Can I generate an answer?

It should ask:

Is the evidence current enough for this answer to be used?

If not, the system should not silently continue.

14.3 Evidence Age, Source Reliability and Operational Timing

Freshness is not only about how old evidence is.

Evidence age matters, but it is not the whole issue.

A record may be recent but unreliable.

A record may be older but still valid.

A source may update frequently but contain poor-quality data.

A source may update slowly but be authoritative.

A timestamp alone does not prove decision validity.

For this reason, deployers should assess evidence freshness through three dimensions:

- evidence age;
- source reliability;
- operational timing.

Evidence age asks:

When was the evidence created, updated, retrieved or verified?

This is the most obvious freshness measure.

It may be expressed in seconds, minutes, hours, days, weeks or months depending on the context.

For some workflows, a daily refresh may be enough.

For others, even a few minutes may matter.

Source reliability asks:

Can the source be trusted for this purpose?

A fresh but unreliable source may not support valid use.

A user-entered note may be current but unverified.

An external database may be current but incomplete.

A cached record may be recent but drawn from a source outside approved scope.

A model-generated summary may be recent but may omit important context.

An authoritative system of record may update slowly, but may still be the correct source for a given process.

Source reliability should therefore be assessed alongside freshness.

Operational timing asks:

Was the evidence fresh at the moment it mattered?

This is often the missing piece.

It is not enough to know that evidence was fresh at the start of a workflow if consequence attaches later.

An AI-supported workflow may include multiple steps:

- evidence retrieval;

- context assembly;
- model output;
- human review;
- approval;
- downstream action;
- external communication;
- final decision.

Evidence may be fresh at retrieval but stale by approval.

It may be fresh at approval but invalidated before downstream action.

It may be fresh when the AI output was generated but outdated before the output was used externally.

This is why freshness should be checked at critical transition points.

The most important transition point is the point where consequence attaches.

For example:

Before a supplier recommendation becomes approval, supplier risk data should be fresh.

Before a HR recommendation affects a candidate, candidate and role data should be fresh.

Before a clinical summary informs care, relevant patient data should be current.

Before a public-service recommendation affects eligibility, case evidence should be up to date.

Before a compliance action is taken, policy and evidence sources should be current.

The deployer should identify which evidence must be fresh at which workflow stage.

This is a design task.

A workflow may require different freshness checks at different points.

For example:

At initial analysis, older background evidence may be acceptable.

At final approval, current risk evidence may be required.

At external communication, the latest decision status may be required.

At downstream automation, authority and evidence state may need revalidation.

This creates the concept of freshness checkpoints.

A freshness checkpoint is a workflow stage where evidence age, source reliability and operational timing must be checked before continuation.

Possible freshness checkpoints include:

- before AI output generation;
- before human review;
- before approval;
- before escalation;
- before downstream action;
- before external notification;
- before final decision;
- before case closure.

Freshness checkpoints support runtime governance because they prevent evidence from being treated as permanently valid.

They also help preserve evidence of control.

A log should not only record that evidence was used.

It should record whether evidence freshness was checked at the relevant transition point.

This may include:

- evidence source;
- source version;
- retrieval timestamp;
- last updated timestamp;
- freshness threshold;
- freshness status;
- source reliability status;
- warning state;
- decision to continue, hold or escalate.

This allows later reconstruction.

If a decision is challenged, the organization can show whether the evidence was fresh enough at the moment the AI-supported workflow became consequential.

Without this, the deployer may struggle to prove decision validity.

14.4 Freshness Thresholds by Risk Level

Evidence freshness should be risk-based.

Not all AI-supported workflows require the same freshness threshold.

A deployer should avoid two mistakes.

The first mistake is treating all evidence as equally urgent.

This creates unnecessary burden and may make governance impractical.

The second mistake is treating all evidence as permanently valid.

This creates silent risk.

A mature approach defines freshness thresholds according to risk, volatility and consequence.

A freshness threshold is the maximum acceptable age or condition gap for evidence before the workflow requires review, refresh, escalation or refusal.

Freshness thresholds may be time-based, event-based or condition-based.

Time-based thresholds define maximum acceptable age.

For example:

- supplier financial data must be less than 24 hours old before high-value approval;
- sanctions checks must be current at the point of decision;
- patient data must include all records from the current episode;
- policy version must match the active policy at the time of use;
- eligibility data must include all submissions received before decision.

Event-based thresholds define conditions that invalidate previous evidence.

For example:

- new fraud flag added;
- supplier status changed;
- employee role changed;
- policy updated;
- risk threshold changed;
- provider instruction updated;
- model version changed;
- data source failed;

- human overseer changed;
- downstream integration changed;
- affected person submitted new information.

Condition-based thresholds define quality conditions required before use.

For example:

- required fields complete;
- source verified;
- representativeness check passed;
- human review completed;
- no unresolved monitoring alert;
- no active escalation state;
- log integrity confirmed;
- authority state current.

Freshness thresholds should be tied to consequence.

A useful model may include four levels.

Level 1: Low consequence.

The AI output supports internal productivity, drafting, summarization or low-impact assistance. Evidence freshness may be less strict, but users should understand limitations.

Level 2: Moderate consequence.

The AI output influences internal prioritization, resource allocation or operational decisions. Freshness thresholds should be defined, and stale evidence should trigger warning or review.

Level 3: High consequence.

The AI output affects employment, finance, healthcare, public services, education, legal processes, safety or significant organizational action. Freshness should be checked before consequence attaches.

Level 4: Critical consequence.

The AI output may affect safety, rights, legal status, high-value financial exposure, clinical action, critical infrastructure or irreversible outcomes. Freshness should be strict, event-based and connected to HOLD or ESCALATE states.

This model is not a legal classification by itself.

It is an operational governance tool.

It helps deployers avoid applying the same freshness standard to every workflow.

The purpose is proportional control.

A deployer should define freshness thresholds during system deployment and review them periodically.

The review should ask:

Are thresholds still appropriate?

Has actual use changed?

Has consequence level increased?

Has evidence volatility changed?

Have incidents occurred?

Have users relied more heavily on outputs?

Have downstream integrations changed?

Have regulators or internal policies changed expectations?

Freshness thresholds should also be visible to human overseers.

A human reviewer should not need to guess whether evidence is stale.

The workflow should show freshness status.

For example:

Evidence current.

Evidence near threshold.

Evidence stale.

Evidence source unavailable.

Evidence freshness unknown.

Each state should connect to action.

Evidence current may allow continuation.

Evidence near threshold may warn.

Evidence stale may require refresh or hold.

Evidence source unavailable may require escalation.

Evidence freshness unknown should not automatically allow continuation in high-risk workflows.

This turns freshness into a practical control state.

14.5 When Stale Evidence Should Trigger HOLD

Stale evidence should not always stop a workflow.

But in high-risk AI systems, stale evidence should sometimes trigger HOLD.

HOLD means the workflow pauses because current control conditions do not yet support continuation.

HOLD is not failure.

HOLD is protection.

It prevents the organization from allowing an AI-supported output to become consequential when the evidence environment is uncertain or degraded.

A stale evidence HOLD may be appropriate when:

- required evidence is older than the defined threshold;
- a critical data source has failed;
- evidence freshness cannot be verified;
- new information is known to exist but has not been retrieved;
- a policy version mismatch exists;
- a risk threshold changed after output generation;
- input data became stale before approval;
- downstream consequence is higher than originally expected;
- human reviewer lacks visibility into evidence condition;
- monitoring shows unresolved data quality warnings;
- the AI output relies on cached context not validated for the current workflow;
- a model fallback or context-compaction step may have removed relevant evidence.

The HOLD state should trigger a defined response.

Possible responses include:

- refresh evidence;
- retrieve source record;

- request human review;
- escalate to process owner;
- escalate to compliance or legal;
- re-run the AI system with updated context;
- require second review;
- reduce authority of the output;
- prevent downstream automation;
- refuse continuation if evidence cannot be restored.

A stale evidence HOLD should also be logged.

The log should record:

- what evidence was stale;
- freshness threshold;
- current evidence age;
- source involved;
- workflow stage;
- consequence level;
- who was notified;
- whether evidence was refreshed;
- who approved continuation;
- whether the output changed after refresh;
- final decision state.

This creates evidence of control.

It shows that the organization did not treat stale evidence as invisible.

It recognized the issue, paused, reassessed and either restored validity or refused continuation.

This is much stronger than allowing the workflow to continue while silently hoping that old evidence remains good enough.

The most important principle is this:

The absence of fresh evidence should not automatically become permission.

In many digital workflows, continuation is the default.

If no hard error occurs, the process moves forward.

High-risk AI governance should challenge this default.

For consequential workflows, continuation should depend on valid conditions.

If evidence freshness is required and cannot be shown, the system should not assume validity.

It should hold.

This does not mean every workflow becomes slow.

Good runtime governance can make freshness checks efficient.

Many checks can be automated.

Many low-risk cases can proceed.

Many stale-evidence issues can be resolved quickly through refresh.

But the architecture must support the possibility of stopping.

A system that cannot pause when evidence is stale is not fully governed.

This is especially important for agentic AI.

Agents may retrieve context once and then continue through multiple steps.

By the time the agent acts, the original context may be outdated, incomplete or superseded.

An agent may also compact context, drop details, use cached summaries or route between models.

These steps can weaken evidence freshness.

For agentic workflows, stale evidence HOLD states may be necessary at tool-execution or external-action boundaries.

Before an agent writes to a system, sends a message, triggers a workflow, recommends a regulated action or escalates a case, it should verify whether the evidence supporting action remains current.

If not, the workflow should hold.

This is how evidence freshness becomes part of admissibility before consequence.

Chapter 14 Summary

Evidence freshness is central to decision validity.

A high-risk AI system may produce a plausible output based on evidence that was once valid but no longer supports the consequence attached to the workflow.

This creates silent failure.

The system may not crash.

The output may look reasonable.

The workflow may complete.

The logs may show normal activity.

But the evidence environment may have degraded.

This chapter argues that freshness is not only a data-quality issue.

It is a runtime governance condition.

Freshness applies to input data, policies, authority, human oversight, provider instructions, risk thresholds, logs, monitoring signals and downstream workflow conditions.

The chapter distinguishes evidence presence from evidence freshness.

Evidence presence asks whether evidence exists.

Evidence freshness asks whether the evidence is still current enough for the intended purpose and consequence.

The chapter also explains that freshness depends on evidence age, source reliability and operational timing.

A timestamp is useful, but not sufficient. Evidence must come from a reliable source and must be fresh at the point where consequence attaches.

The chapter proposes risk-based freshness thresholds.

Low-consequence workflows may tolerate less strict freshness requirements.

High-consequence workflows require stronger freshness proof.

Critical workflows may require strict, event-based and condition-based freshness checks.

Finally, the chapter explains when stale evidence should trigger HOLD.

HOLD is not failure.

It is a protective governance state.

It allows the organization to pause, refresh evidence, escalate uncertainty, reduce authority or refuse continuation before stale evidence creates consequence.

The central question is:

Was the evidence still fresh enough to justify the authority attached to the AI-supported output?

If yes, continuation may be allowed.

If no, the workflow should stop or refuse.

If uncertain, the workflow should hold and reassess.

This is how evidence freshness becomes decision validity.

Chapter 15

Pause, Escalate and Refuse Conditions

Legal anchor and framework boundary. [Article 14\(4\)\(d\)-\(e\)](#); [Article 26\(5\)](#). Pause, escalation and refusal states are framework mechanisms inspired by statutory oversight, monitoring and suspension duties.

15.1 Why Successful Completion Is Not Always Success

In ordinary software environments, successful completion often means that a workflow performed as expected.

The system did not crash.

The request completed.

The output was generated.

The task moved to the next stage.

The transaction was recorded.

The dashboard stayed green.

From a technical perspective, this may look like success.

But in high-risk AI governance, successful completion is not always success.

An AI-supported workflow can complete successfully while the conditions required for valid continuation are no longer present.

The system may produce a plausible output.

The workflow may route the case correctly.

The human reviewer may approve.

The downstream process may continue.

The logs may show normal execution.

Yet the workflow may have continued when it should have paused.

This is one of the most important practical problems for deployers of high-risk AI systems.

AI governance cannot be built only around the question:

Did the system complete the task?

It must also ask:

Should the system have been allowed to complete the task under current conditions?

This distinction is central to Article 26.

The deployer's obligations around appropriate use, human oversight, input data, monitoring, logs and risk escalation all point toward the same operational question:

Do current control conditions support continuation?

If the answer is yes, successful completion may be valid.

If the answer is no, successful completion may represent a governance failure.

If the answer is unknown, automatic continuation may be unsafe.

For example:

A procurement AI system may successfully recommend a supplier, but supplier risk data may be stale.

A HR AI system may successfully rank candidates, but the use case may have expanded beyond the approved intended purpose.

A clinical AI assistant may successfully summarize a record, but recent patient information may be missing.

A public-service AI system may successfully route a case, but the relevant policy version may have changed.

A financial services AI system may successfully produce a risk assessment, but human oversight may not have been available.

An agentic AI workflow may successfully complete a tool chain, but a downstream system may treat a recommendation as an executable signal.

In each case, the technical workflow may succeed.

But the governance conditions may fail.

This means that deployers must design for more than success paths.

They must design for pause paths, escalation paths and refusal paths.

A mature AI governance system should not assume that every valid workflow ends in completion.

Sometimes the correct result is interruption.

Sometimes the correct result is escalation.

Sometimes the correct result is refusal.

This is not inefficiency.

It is control.

A system that can stop before consequence attaches may be safer, more trustworthy and more compliant than a system that always completes smoothly.

The most underrated feature in high-risk AI governance may be a system that can say:

I know what to do, but I am no longer sure I am allowed to do it.

That is not failure.

That is runtime governance.

15.2 The Absence of a Pause Condition

Many AI governance failures may not come from the absence of policies.

They may come from the absence of pause conditions.

A pause condition is a predefined state or trigger that prevents an AI-supported workflow from automatically continuing when required control conditions are missing, uncertain or degraded.

Pause conditions are important because modern workflows are designed for momentum.

Once a process begins, systems usually try to complete it.

Automation rewards continuity.

Interfaces often make approval easier than challenge.

Users are encouraged to resolve tasks quickly.

Dashboards often measure throughput.

Organizations often treat interruption as inefficiency.

AI systems add another layer of pressure because their outputs may appear confident, fluent and structured.

A system that produces a clean recommendation may make continuation feel natural.

A system that produces a well-written explanation may make review feel unnecessary.

A system that generates a score may make uncertainty look precise.

A system that summarizes evidence may reduce the human desire to inspect the source.

If there is no pause condition, the workflow may continue by default.

This is dangerous in high-risk environments.

The absence of a pause condition means that degraded conditions may not become visible as governance events.

Stale evidence may not pause the workflow.

Missing human oversight may not pause the workflow.

Use outside intended purpose may not pause the workflow.

Authority uncertainty may not pause the workflow.

A model fallback may not pause the workflow.

A downstream automation change may not pause the workflow.

Incomplete logs may not pause the workflow.

The organization may later discover that the system continued through conditions that should have triggered review.

This is why pause conditions are essential.

They transform uncertainty into an operational state.

Without a pause condition, uncertainty is often absorbed into continuation.

With a pause condition, uncertainty becomes visible.

The workflow can hold.

A human can review.

Evidence can be refreshed.

Authority can be checked.

Provider instructions can be consulted.

Risk can be assessed.

Logs can be preserved.

Escalation can occur.

A pause condition therefore protects the organization from silent continuation.

Examples of pause conditions include:

- required input data missing;
- evidence freshness threshold exceeded;
- source reliability unknown;
- human oversight unavailable;
- reviewer lacks required authority;
- workflow use outside approved scope;
- output consequence higher than approved consequence level;
- model fallback occurred without authority reassessment;
- policy version mismatch;
- provider instruction uncertainty;
- unresolved monitoring alert;
- data representativeness concern;
- log integrity failure;
- downstream automation detected;
- affected-person notification state incomplete;
- serious risk signal detected.

Pause conditions should be risk-based.

Not every warning should stop every workflow.

But high-consequence workflows should not treat missing control conditions as harmless.

The deployer should define which conditions require:

- warning;
- review;
- HOLD;
- escalation;
- refusal;
- suspension.

This turns governance into operating logic.

The absence of a pause condition is dangerous because it allows the system to confuse technical capability with permission.

The system can continue.

Therefore it continues.

But high-risk AI governance requires a different rule:

The system may continue only if current control conditions support continuation.

This is the principle that pause conditions enforce.

15.3 HOLD, ESCALATE and REFUSE as Governance States

High-risk AI systems need more than a binary pass/fail model.

Many real governance situations are not clear failures.

They are uncertainty states.

The system may have enough evidence for a draft, but not enough evidence for a decision.

The model may be available, but not the approved model.

The human reviewer may be present, but not authorized for this consequence level.

The data may be complete enough for triage, but not for final action.

The output may be appropriate for internal review, but not for external communication.

The workflow may be within intended purpose at the start, but drift into a different use later.

A binary model cannot handle these situations well.

If the only options are continue or fail, organizations may continue too often or stop too harshly.

A more useful runtime governance model includes four states:

ALLOW.

HOLD.

ESCALATE.

REFUSE.

ALLOW means the workflow may continue because current control conditions support continuation.

HOLD means the workflow pauses because required control conditions are missing, incomplete, uncertain or degraded, but may be restored.

ESCALATE means the workflow requires review by a higher authority, human overseer, compliance function, risk owner, provider or other responsible party.

REFUSE means the workflow should not proceed under current conditions.

These states are not merely labels.

They are governance actions.

ALLOW should be logged with the reason continuation was permitted.

HOLD should be logged with the condition that prevented continuation.

ESCALATE should be logged with the escalation route, recipient, reason and outcome.

REFUSE should be logged with the boundary, risk or missing condition that prevented use.

This creates evidence of control.

For example:

ALLOW: input data fresh, human review completed, authority valid, use within intended purpose, logs complete.

HOLD: supplier risk data stale; refresh required before recommendation may influence approval.

ESCALATE: AI output conflicts with policy threshold; compliance review required.

REFUSE: system is being used for a purpose outside provider instructions and internal approval.

These states help the deployer avoid a common failure:

treating uncertainty as permission.

HOLD is especially important.

HOLD is not a negative outcome.

It is a state of controlled uncertainty.

It allows the organization to pause without declaring a failure.

It gives the deployer time to restore evidence, verify authority, involve a human, consult the provider, review the policy or reassess scope.

ESCALATE is also essential.

Not every issue can be resolved by the user or the system.

Some issues require a more accountable authority.

For example:

- legal uncertainty;
- serious risk signal;
- use outside intended purpose;
- affected-person challenge;
- policy conflict;
- model substitution;
- high-consequence override;
- unresolved monitoring alert;
- evidence integrity failure.

REFUSE is the strongest state.

It means the workflow is not admissible under current conditions.

This may be required when:

- the use is prohibited or outside allowed scope;
- required human oversight is unavailable;
- evidence cannot be restored;
- input data is unsuitable;
- authority is absent;
- serious risk is unresolved;
- logs are insufficient for required accountability;
- downstream consequences cannot be controlled.

Refusal is not system failure.

It is a governance feature.

A high-risk AI system that can refuse unsafe continuation is more mature than one that always produces an output.

This is why runtime governance should include refusal logic.

A deployer should not fear refusal as a sign of weakness.

It should treat refusal as proof that control boundaries exist.

15.4 Human Review Before Consequence Binds

Human review is most valuable before consequence binds.

A review after consequence may still be useful for audit, appeal, correction or learning.

But it is not the same as preventive control.

The timing of human review is therefore critical.

For Article 26 purposes, deployers should identify the point at which an AI-supported workflow becomes consequential.

This may be when:

- a candidate is shortlisted or rejected;
- a worker is evaluated;
- a supplier is approved;
- a credit risk decision is made;
- a clinical pathway is influenced;
- a public benefit case is routed;
- a customer receives a final decision;
- an enforcement action is triggered;
- a system record is updated;
- an external communication is sent;
- a downstream automation begins;
- an access permission changes.

Before that point, human review can prevent harm.

After that point, human review may only correct harm.

This distinction matters for high-risk AI systems.

A workflow may include a human review step, but if the review occurs too late, it may not function as meaningful oversight.

For example:

A human reviewer may review a report after it has already influenced a decision.

A manager may review an AI-generated performance score after the score has already shaped employee treatment.

A clinician may see an AI-generated summary after it has already influenced triage routing.

A compliance officer may review an AI-generated case assessment after customer action has already been taken.

A public-sector reviewer may examine the AI-supported recommendation after a citizen has already received the decision.

In these situations, human review may support accountability, but not control.

The deployer should therefore distinguish between:

- pre-consequence review;
- post-consequence review;
- appeal review;
- audit review;
- incident review.

All may matter.

But they do not serve the same function.

Pre-consequence review is a control.

Post-consequence review is correction.

Appeal review is redress.

Audit review is assurance.

Incident review is learning and response.

For high-risk AI deployment, the most important question is:

Does human review occur before the AI-supported output becomes operationally binding, influential or consequential?

The word “binding” should be understood broadly.

An AI output may not be legally binding but may still be practically binding.

A recommendation may be so influential that it is rarely challenged.

A ranking may determine who receives attention.

A risk flag may lead to increased scrutiny.

A summary may shape the evidence considered.

A draft decision may become final with minimal change.

A route may determine a person's pathway through an institution.

This is why human review should occur before practical consequence, not merely formal final decision.

A mature deployer should identify consequence points and define review requirements.

For example:

Low-consequence output may require no individual human review.

Moderate-consequence output may require spot review or review on exception.

High-consequence output may require documented human review.

Critical-consequence output may require human review plus second review, escalation or control-gate approval.

Human review should also be supported by evidence.

The reviewer should see:

- the AI output;
- the input data condition;
- the evidence sources;
- freshness status;
- warnings;
- system limitations;
- policy version;
- authority state;
- previous escalations;
- downstream consequence;
- available actions.

Without this information, review may be superficial.

The reviewer should also have real options.

Approve.

Modify.

Reject.

Request more evidence.

Escalate.

Hold.

Refuse continuation.

If the workflow gives the human only an approve button, human review is weak.

If the workflow makes challenge difficult, human review is biased toward continuation.

If the organization treats override as failure, human review is culturally constrained.

Human review before consequence binds must therefore be designed as a practical control point.

Not a symbolic step.

Not a final signature.

Not a box to tick.

A real opportunity to change the outcome.

15.5 Designing Refusal as a Safety Feature

Refusal is often misunderstood.

In product design, refusal may be seen as friction.

In business process design, refusal may be seen as inefficiency.

In automation design, refusal may be seen as failure.

But in high-risk AI governance, refusal can be a safety feature.

A system that refuses to continue when control conditions are invalid is doing something valuable.

It is preserving the boundary between capability and permission.

The AI system may be capable of generating an output.

The workflow may be capable of completing.

The user may want a result.

The organization may prefer speed.

But capability is not permission.

This distinction is central to responsible deployment.

A high-risk AI system should not be designed only to answer, recommend, score, rank, classify, summarize, route or act.

It should also be able to recognize when it should not do those things.

Refusal may be appropriate when:

- the use case is outside intended purpose;
- required input data is missing;
- evidence freshness cannot be established;
- human oversight is required but unavailable;
- authority is absent or invalid;
- the output would affect a person without required notice;
- the requested action exceeds system permission;
- the model used does not support the required authority level;
- the workflow would create uncontrolled downstream consequence;
- logs cannot preserve required evidence;
- a serious risk signal is unresolved;
- provider instructions prohibit the use;
- the system cannot produce an output without unsupported assumptions.

This refusal should be explainable.

Not necessarily in technical detail for every user, but in governance terms.

A refusal should state, where appropriate:

what condition failed;
 why continuation is not allowed;
 what evidence is missing;
 who may review;
 what action may restore continuation;
 whether escalation is available;
 whether the refusal is final or temporary.

For example:

Refused: required human oversight is unavailable for this high-consequence workflow.

Held: supplier risk data is stale and must be refreshed before approval.

Escalated: current use appears outside approved intended purpose.

Refused: the system is not authorized to generate final employment decisions.

Held: evidence freshness cannot be verified.

Escalated: output conflicts with active policy version.

These responses are much stronger than silent failure or generic error messages.

They show that the system understands the governance boundary.

Refusal should also be logged.

The log should capture:

refusal reason;

control condition involved;

user or workflow;

system version;

input state;

authority state;

escalation path;

final outcome;

whether the issue was resolved;

whether repeated refusal indicates systemic problem.

Refusal logs are valuable.

They show that control boundaries are operating.

They also reveal where workflows may need improvement.

If many refusals occur because data is stale, the organization may need better data integration.

If many refusals occur because human oversight is unavailable, the organization may need staffing or workflow redesign.

If many refusals occur because use cases are outside scope, the organization may need clearer access controls and training.

If many refusals occur because logs are incomplete, the organization may need stronger evidence architecture.

Refusal is therefore not only a protective action.

It is a diagnostic signal.

A mature deployer should not hide refusal rates.

It should study them.

They may reveal where governance assumptions are failing.

The ultimate goal is not to make AI systems refuse unnecessarily.

The goal is to ensure that refusal is available when continuation is not justified.

A system without refusal logic may be efficient.

But in high-risk contexts, it may be unsafe.

A system with refusal logic may be slightly slower.

But it may be far more trustworthy.

This is the future of high-risk AI control:

not systems that always answer,

but systems that know when an answer should not become action.

Chapter 15 Summary

Successful completion is not always success.

A high-risk AI workflow may complete technically while failing governance conditions.

The AI system may generate a plausible output, the workflow may move forward, the human may approve and the log may record completion. But if input data is stale, authority is invalid, oversight is missing, scope has drifted or evidence is insufficient, the workflow may have continued when it should have paused.

This chapter argues that deployers need pause, escalation and refusal conditions.

The absence of a pause condition is one of the most dangerous weaknesses in AI governance.

Without pause conditions, uncertainty is often absorbed into continuation.

With pause conditions, uncertainty becomes visible and manageable.

The chapter proposes four runtime governance states:

ALLOW.

HOLD.

ESCALATE.

REFUSE.

ALLOW means current control conditions support continuation.

HOLD means required conditions are missing, incomplete, uncertain or degraded, but may be restored.

ESCALATE means higher-level review is required.

REFUSE means the workflow should not proceed under current conditions.

These states allow deployers to move beyond binary pass/fail logic.

They also create evidence of control.

The chapter emphasizes that human review is most valuable before consequence binds. Post-consequence review may support audit, appeal or correction, but pre-consequence review is the stronger control point.

Finally, the chapter argues that refusal should be treated as a safety feature.

A high-risk AI system should not only be capable of producing outputs.

It should also be capable of refusing continuation when control conditions do not support use.

The central governance principle is simple:

Capability is not permission.

A system may be able to continue.

But it should continue only when current conditions justify continuation.

This is how Article 26 becomes operational control.

Chapter 16

Financial Services and Capital Markets

Legal anchor and framework boundary. [Article 26](#). Financial-services law may deem some monitoring duties fulfilled through sector governance; sector-specific obligations must be checked separately.

16.1 AI Governance in Regulated Finance

Financial services and capital markets are among the most important environments for practical AI governance.

They are highly regulated.

They are evidence-intensive.

They depend on trust.

They operate through delegated authority.

They rely on records, controls, approvals, oversight, auditability, accountability and regulatory confidence.

This makes the sector especially relevant for Article 26.

In financial services, AI systems may be used across a wide range of activities, including:

- client onboarding;
- know-your-customer processes;
- anti-money laundering monitoring;
- fraud detection;
- transaction monitoring;
- credit risk;
- market surveillance;
- investment research;
- portfolio analytics;
- compliance monitoring;
- regulatory reporting;
- customer support;
- complaint handling;
- fund administration;
- operational risk;

- cyber risk;
- internal audit;
- workflow automation;
- agentic process support.

Not all of these uses will necessarily fall within the same risk category under the EU AI Act.

But many can become high-consequence because they affect regulated processes, customer treatment, market integrity, investor protection, financial exposure, operational resilience or supervisory evidence.

This is why financial services deployers should not treat AI governance as a separate technology exercise.

It must be connected to existing control disciplines.

Financial institutions already understand many of the concepts that AI governance requires:

- named accountability;
- audit trails;
- segregation of duties;
- approval limits;
- model risk management;
- data governance;
- outsourcing risk;
- operational resilience;
- incident management;
- compliance monitoring;
- record retention;
- regulatory reporting;
- board oversight;
- three lines of defence.

The challenge is that AI changes how these controls need to operate.

AI systems may not simply execute a predefined rule.

They may rank, score, classify, summarize, generate, recommend, reason, route or interact with tools.

Their outputs may influence humans indirectly.

Their recommendations may become workflow defaults.

Their summaries may shape what evidence is reviewed.

Their alerts may prioritize cases.

Their scores may influence treatment of customers, clients, suppliers, employees or counterparties.

Their agentic workflows may perform multi-step operations before a human sees the result.

This creates a new governance question for financial services deployers:

Can the organization prove that AI-supported financial workflows remained within valid control conditions when they affected a regulated process?

This question is directly aligned with the central thesis of this study.

For financial services, Article 26 should not be interpreted only as a legal compliance obligation.

It should be interpreted as part of operational control.

A financial institution using a high-risk AI system should be able to show:

- what system was used;
- for which regulated process;
- under whose authority;
- with what input data;
- under which policy version;
- with what human oversight;
- with what monitoring;
- with what logs;
- with what escalation route;
- with what downstream consequence;
- and why continuation was allowed.

This is evidence of control.

The financial sector is already familiar with the idea that decisions must be reconstructable.

AI makes reconstructability harder, but also more important.

A transaction monitoring decision, risk assessment, client classification, suitability process, complaint response or regulatory report may involve multiple systems, data sources, human reviewers and control steps.

If AI enters that chain, the deployer must be able to identify its role.

Was the AI system advisory?

Did it shape prioritization?

Did it summarize evidence?

Did it recommend action?

Did it draft reasoning?

Did it trigger a workflow?

Did it alter the decision environment?

Did a human reviewer meaningfully examine the output?

Did the human have authority to override?

Was the output logged?

Was the input data fresh?

Was the use within intended purpose?

Was the downstream consequence controlled?

In regulated finance, these questions are not theoretical.

They go to the heart of supervisory confidence.

A regulator may not be satisfied with a statement that “a human made the final decision” if the AI system shaped the evidence, prioritization or recommendation that the human relied upon.

A board may not be satisfied with an AI inventory if it cannot see whether AI-supported workflows are controlled.

An auditor may not be satisfied with technical logs if they do not reconstruct the decision trail.

A customer or investor may not be satisfied with a final outcome if the AI influence behind it is opaque.

This is why financial services is a strong test case for Article 26.

The sector already has mature expectations around control.

The AI governance challenge is to make those expectations work in AI-supported, hybrid and agentic workflows.

16.2 Fund Administrators and Evidence Chains

Fund administrators may become a critical operational layer in AI governance.

They sit close to records, processes, reconciliations, reporting, investor services, compliance workflows, fund operations and client evidence needs.

This position matters because AI governance is not only about the entity that builds or procures the AI system.

It is also about the operational chain in which AI-supported outputs become part of regulated processes.

Fund administrators may use AI internally.

They may also support fund managers, asset managers or other regulated clients who use AI-supported processes.

This creates a dual role.

First, the administrator may need to govern its own AI use.

Second, the administrator may need to preserve, structure or provide evidence that clients depend on for their own governance, compliance or audit obligations.

This does not mean that the administrator automatically becomes responsible for every AI use by a client.

Responsibilities will depend on the specific service, contract, role, system, process and regulatory context.

But operationally, fund administrators may become part of the evidence infrastructure.

For example, a fund administrator may hold or process records relevant to:

- investor onboarding;
- transaction records;
- fund accounting;
- net asset value processes;
- reconciliations;
- investor communications;
- regulatory reporting;
- anti-money laundering workflows;
- transfer agency operations;

- subscription and redemption processes;
- client reporting;
- audit support;
- operational controls.

If AI is used to support any of these workflows, the evidence chain becomes important.

The organization may need to show:

Which system was used?

What data was supplied?

Which records were relied upon?

What output was produced?

Who reviewed it?

Was the output used internally or externally?

Did it affect a regulated process?

Was the workflow within approved scope?

Were logs retained?

Was escalation possible?

Was the provider involved?

Was the client informed where necessary?

This is where Article 26 and financial services governance meet.

A deployer using high-risk AI must be able to prove appropriate use, oversight, monitoring, logging and escalation.

If the relevant evidence sits partly with a fund administrator, the administrator's operational controls become important to the client's AI governance posture.

This creates practical questions for fund administrators.

Do service agreements define AI-related responsibilities?

Do clients know where AI is used in administrator workflows?

Can the administrator identify AI-supported processes?

Can the administrator provide evidence of human review?

Can the administrator preserve logs relevant to regulated workflows?

Can the administrator distinguish AI-generated output from human-authored records?

Can the administrator support incident reconstruction?

Can the administrator detect when AI affects a client-facing or investor-facing process?

Can the administrator respond to client audit or regulator requests?

Can the administrator show that AI use remained within intended purpose?

This is not only a legal question.

It is an operational readiness question.

Fund administrators may need to develop AI evidence-chain capabilities.

These capabilities may include:

- AI system inventory by client service line;
- workflow mapping for AI-supported processes;
- data lineage records;
- human review records;
- output classification;
- log retention controls;
- escalation pathways;
- client notification procedures;
- audit support packs;
- third-party AI provider risk records;
- change-control records;
- incident response coordination.

For example, if an AI system supports AML case summarization, the administrator should know whether the output was used only for drafting, whether a human analyst reviewed source evidence, whether the summary entered the case record, whether the reviewer modified it, and whether the case outcome relied on it.

If an AI system supports investor communication drafting, the administrator should know whether the communication was reviewed, whether the source records were current, whether the output was sent externally, and whether it created investor-facing consequence.

If an AI system supports regulatory reporting checks, the administrator should know whether AI output was advisory, whether exceptions were reviewed, and whether final reporting remained under human control.

The evidence chain must be clear.

This is especially important because AI may blur responsibility.

A fund manager may rely on an administrator.

An administrator may rely on a third-party technology provider.

A technology provider may rely on an AI model provider.

A human operator may rely on an AI-generated summary.

A final report may rely on a workflow that includes AI assistance.

If something goes wrong, the question will not only be:

Who used the AI system?

The question will be:

Where did AI influence enter the evidence chain, and who controlled it?

Fund administrators that can answer this question will be more valuable in an AI-governed financial ecosystem.

They will not merely provide services.

They will provide evidence confidence.

16.3 AI-Supported Compliance Workflows

Compliance is one of the most natural areas for AI adoption in financial services.

It is also one of the most sensitive.

Compliance teams face large volumes of information, complex regulation, repetitive review tasks, changing guidance, increasing reporting obligations and pressure to operate efficiently.

AI systems may help with:

- transaction monitoring;
- alert triage;
- suspicious activity review;

- sanctions screening support;
- policy mapping;
- regulatory change monitoring;
- complaint analysis;
- case summarization;
- control testing;
- internal audit support;
- evidence retrieval;
- report drafting;
- workflow prioritization;
- risk classification;
- document review;
- employee communications;
- compliance training support.

These uses may improve efficiency.

They may help human teams process more information.

They may identify patterns.

They may reduce manual burden.

They may support consistency.

But they also introduce new governance risks.

A compliance AI system can shape what receives attention.

It can prioritize cases.

It can summarize evidence.

It can suggest closure.

It can draft reasoning.

It can classify risk.

It can recommend escalation.

It can influence whether a human reviewer sees a case as urgent, low risk, suspicious, explainable or resolved.

This means AI-supported compliance is not only administrative support.

It can affect the control environment.

The key risk is that AI may become part of compliance judgment without the organization recognizing the authority attached to its output.

For example:

An AI system summarizes a suspicious transaction case. The summary omits a relevant fact. The human reviewer relies on the summary and closes the case.

An AI system ranks compliance alerts. Lower-ranked alerts receive less attention. Over time, alert ranking becomes a practical resource-allocation mechanism.

An AI system tracks regulatory change and summarizes obligations. A misinterpretation enters internal policy drafting.

An AI system drafts customer complaint responses. The response appears compliant but fails to reflect case-specific evidence.

An AI agent retrieves policy documents and recommends whether a control applies. The recommendation is accepted without checking the source.

In each example, the AI system may not make the final decision.

But it shapes compliance work.

This is why financial services deployers need strong evidence of control around AI-supported compliance workflows.

For each compliance use case, the organization should define:

- the role of the AI system;
- the decision or process affected;
- the authority attached to output;
- the required human review;
- the data sources used;
- the policy version relied upon;
- the logging requirements;
- the escalation triggers;
- the boundary between assistance and decision-making;
- the evidence retained.

Compliance workflows should be especially sensitive to evidence freshness.

Regulatory rules change.

Internal policies change.

Customer status changes.

Transaction data changes.

Risk thresholds change.

Sanctions lists change.

Case records change.

A compliance AI system using stale evidence can create false comfort.

For high-consequence compliance use cases, the workflow should know whether the evidence is current.

If not, the output should not proceed with full authority.

The correct state may be HOLD or ESCALATE.

Human oversight in compliance workflows must also be meaningful.

A human reviewer should not merely approve an AI-generated summary.

The reviewer should understand:

- what source material was used;
- what was omitted;
- what uncertainty exists;
- whether the output conflicts with policy;
- whether escalation is required;
- whether the AI system is within intended use;
- whether the output can be used in regulated records.

This is especially important where AI outputs may become part of audit evidence or regulatory reporting.

A regulator may later examine the record.

If the record contains AI-generated content, the organization should be able to explain how it was reviewed, validated and approved.

The organization should also preserve evidence that the AI system did not replace accountable compliance judgment where human judgment was required.

This may include:

reviewer identity;

reviewer role;

evidence reviewed;

AI output;

modifications;
 source documents;
 policy version;
 escalation decision;
 final outcome;
 decision trail.

AI-supported compliance should therefore be governed through a simple principle:

AI may assist compliance work, but it should not silently redefine compliance authority.

If an AI system changes what compliance teams see, how they prioritize, what they record or how they decide, then governance must make that influence visible.

16.4 Audit Readiness and Regulator-Facing Proof

Financial services organizations are accustomed to audit.

They are accustomed to regulatory inquiry.

They are accustomed to producing records.

But AI changes what those records may need to show.

A traditional audit trail may show that a process was completed.

An AI governance decision trail must show how AI influenced the process, what controls operated and why the workflow was allowed to continue.

This is especially important for Article 26 deployers.

A deployer may need to show not only that an AI system was registered or approved, but that it was used under valid control conditions.

In a financial services context, regulator-facing proof may need to answer:

Which AI systems are used in regulated processes?

Which systems are high-risk or high-consequence?

What is each system's intended purpose?

What is its actual use?

Who owns the system?

Who has accountability?

Which third parties are involved?

What data is used?

What human oversight exists?

How is output validated?

How are logs retained?

How are incidents escalated?

How is outsourcing or third-party risk managed?

How is model substitution handled?

How are affected persons or clients informed where required?

How does the organization know that use remains within approved boundaries?

These questions require more than policies.

They require connected evidence.

A regulator may ask for system-level evidence.

This includes system identity, provider, risk classification, intended purpose, version, deployment date, approved use cases and internal ownership.

A regulator may ask for workflow-level evidence.

This includes where the system is used, what process it affects, what output it produces, who reviews it and what consequence follows.

A regulator may ask for control-level evidence.

This includes human oversight, data controls, access controls, monitoring, escalation, incident handling and log retention.

A regulator may ask for case-level evidence.

This includes a specific AI-supported workflow, the input data used, the output generated, the human review performed, the decision made and the evidence supporting continuation.

A regulator may ask for change evidence.

This includes system updates, model changes, provider updates, policy changes, workflow changes, data source changes, permission changes and reassessment records.

The organization should be prepared for all levels.

This is audit readiness.

Audit readiness should not be created after an incident.

It should be designed into the AI governance architecture.

For financial services deployers, an audit-ready AI evidence pack may include:

- AI system register;
- high-risk classification rationale;
- intended purpose and actual use mapping;
- provider instructions;
- internal use procedures;
- system ownership and accountability;
- third-party and outsourcing risk assessment;
- data source mapping;
- input data validation controls;
- human oversight model;
- training records;
- monitoring design;
- alert and escalation rules;
- log retention policy;
- decision trail examples;
- incident response procedure;
- evidence of periodic reassessment;
- board or governance committee reporting;
- internal audit findings;
- remediation records.

The strongest audit evidence is not a pile of documents.

It is a coherent chain.

The chain should connect:

system,

workflow,

data,

output,

oversight,
authority,
monitoring,
logs,
escalation,
consequence,
and decision to continue.

This is the evidence chain.

Financial services organizations should also consider the relationship between AI governance and existing controls.

AI governance should not sit in isolation.

It should connect to:

model risk management;
operational risk;
compliance risk;
technology risk;
outsourcing risk;
data governance;
cybersecurity;
conduct risk;
internal audit;
legal and regulatory affairs;
board risk oversight.

This integration matters because AI risks may appear across control domains.

A model issue may become a conduct issue.

A data issue may become a compliance issue.

An outsourcing issue may become an operational resilience issue.

A workflow automation issue may become an accountability issue.

A recordkeeping issue may become a regulatory reporting issue.

A human oversight issue may become a governance failure.

The organization's audit readiness should reflect these connections.

The goal is not only to prove that AI governance exists.

The goal is to prove that AI governance is embedded in the institution's control environment.

16.5 Financial Services Implementation Checklist

A financial services deployer implementing Article 26 should translate obligations into a practical checklist.

The purpose of the checklist is not to replace legal advice, regulatory analysis or internal governance frameworks.

Its purpose is to help organizations convert deployer obligations into operational evidence.

1. System Identification

The organization should identify all AI systems used in financial services workflows.

For each system, record:

- system name;
- provider;
- owner;
- business function;
- intended purpose;
- actual use;
- risk classification;
- affected process;
- affected clients, investors, employees or counterparties;
- data sources;
- third-party dependencies;
- downstream systems;
- deployment status.

2. High-Risk and High-Consequence Mapping

The organization should identify whether the system falls within high-risk AI categories or otherwise affects high-consequence regulated processes.

The organization should map:

- regulated process affected;
- consequence level;
- legal or compliance sensitivity;
- client or investor impact;
- market integrity implications;
- financial exposure;
- operational resilience impact;
- affected-person implications;
- supervisory relevance.

3. Intended Purpose and Actual Use

The organization should compare provider-defined intended purpose with actual internal use.

Evidence should show:

- approved use case;
- prohibited uses;
- actual users;
- actual workflow;
- output role;
- downstream consequence;
- scope change history;
- reassessment records.

4. Human Oversight

The organization should define and evidence human oversight.

Records should show:

- oversight owner;
- reviewer roles;
- competence requirements;
- training records;
- review timing;
- review evidence;
- override capability;
- escalation route;
- intervention logs;
- review quality monitoring.

5. Input Data Governance

The organization should map and control input data.

Evidence should show:

- data sources;
- data owners;
- source reliability;
- freshness thresholds;
- completeness checks;
- representativeness where relevant;
- missing-data rules;
- validation logs;
- data lineage;
- exception handling.

6. Monitoring and Runtime Control

The organization should define monitoring signals and governance states.

Monitoring should include:

- performance changes;
- data source failure;
- stale evidence;
- workflow drift;
- human oversight availability;
- authority drift;
- downstream integration change;
- model fallback;
- unresolved alerts;
- repeated exceptions.

Each signal should connect to ALLOW, HOLD, ESCALATE or REFUSE logic.

7. Logging and Decision Trails

The organization should preserve logs that support reconstructability.

Logs should help reconstruct:

- system identity;
- input data condition;

- output;
- human review;
- authority state;
- policy version;
- warnings;
- escalation;
- downstream effect;
- final decision;
- reason continuation was allowed.

8. Third-Party and Outsourcing Risk

The organization should identify third-party dependencies.

This includes:

- AI providers;
- model providers;
- data providers;
- cloud providers;
- workflow tool providers;
- outsourcing partners;
- fund administrators;
- managed service providers;
- external compliance vendors.

Evidence should show contractual controls, service boundaries, incident contact routes, data handling, audit support and change notification.

9. Incident and Escalation Readiness

The organization should define AI incident escalation.

The process should identify:

- what must be reported;
- who triages;
- who preserves evidence;
- who contacts provider;
- who decides suspension;
- who informs regulators where required;
- who communicates internally;
- who reviews affected cases;

who approves return to service.

10. Board and Senior Management Oversight

The organization should provide senior leadership with meaningful AI governance information.

Board or senior management reporting should include:

- high-risk AI inventory;
- major use cases;
- control maturity;
- incidents;
- unresolved risks;
- third-party dependencies;
- monitoring trends;
- human oversight effectiveness;
- audit findings;
- remediation status.

11. Periodic Reassessment

The organization should reassess AI systems periodically and when triggers occur.

Triggers may include:

- use case change;
- provider update;
- model change;
- workflow change;
- new downstream integration;
- incident;
- regulatory change;
- data source change;
- increased reliance;
- authority drift;
- evidence failure.

12. Evidence Pack

For each high-risk or high-consequence AI system, the organization should maintain an evidence pack.

This pack should answer:

What is the system?

What is it used for?

Who owns it?

What process does it affect?

What data does it use?

What oversight exists?

What logs are retained?

What monitoring occurs?

What escalation pathways exist?

What incidents occurred?

How does the organization know the system remains within valid boundaries?

This checklist turns Article 26 into financial services control practice.

It does not make AI governance simple.

But it makes the evidence problem visible.

In regulated finance, that visibility may become one of the most important sources of trust.

Chapter 16 Summary

Financial services and capital markets are natural environments for Article 26 operational governance.

They are regulated, evidence-intensive and dependent on accountability, auditability, controls, approvals and supervisory confidence.

AI systems may support many financial workflows, including compliance, AML, fraud detection, transaction monitoring, investment research, fund administration, customer support, complaint handling, regulatory reporting and internal workflow automation.

The core question is not only whether AI is used.

The core question is whether the organization can prove that AI-supported financial workflows remained within valid control conditions when they affected regulated processes.

This chapter explains why fund administrators may become important evidence-chain actors.

They may not only govern their own AI use, but also support client evidence needs where AI affects fund operations, investor services, reporting, AML, reconciliations or other regulated workflows.

The chapter also examines AI-supported compliance workflows.

AI can assist compliance teams, but it can also shape prioritization, evidence review, case outcomes, policy interpretation and escalation decisions. For this reason, AI influence in compliance workflows must be visible, reviewed and evidenced.

Audit readiness is a central requirement.

Financial services deployers should be prepared to produce system-level, workflow-level, control-level, case-level and change-level evidence.

The strongest evidence is not a pile of documents.

It is a connected chain showing system, workflow, data, output, oversight, authority, monitoring, logs, escalation, consequence and decision to continue.

Finally, the chapter provides a financial services implementation checklist covering system identification, high-risk mapping, intended purpose, human oversight, input data governance, monitoring, logging, third-party risk, escalation readiness, board oversight, periodic reassessment and evidence packs.

The central financial services governance question is:

Can the institution show not only that AI was used, but that it was used under valid, monitored, accountable and reconstructable control conditions?

In regulated finance, that question may become one of the most important tests of AI readiness.

Chapter 17

Human Resources and Employment

Legal anchor and framework boundary. [Article 14](#); [Article 26\(7\)](#) and [\(11\)](#). Employment, data-protection and labour-law duties may apply in addition to the AI Act; examples are illustrative.

17.1 High-Risk HR AI Systems

Human resources and employment may become one of the most important practical areas for Article 26 deployer governance.

The reason is simple.

Work is not only an economic activity.

Work affects livelihood, dignity, opportunity, career development, identity, social status, autonomy and personal security.

When AI systems enter employment workflows, they may influence decisions that affect a person's future.

This is why AI systems used in employment, worker management and access to self-employment deserve serious governance attention.

AI systems may be used in HR and employment for many purposes, including:

- candidate sourcing;
- CV screening;
- applicant ranking;
- interview scheduling;
- interview analysis;
- skills matching;
- promotion support;
- performance evaluation;
- worker monitoring;
- task allocation;
- shift scheduling;
- productivity scoring;
- training recommendations;
- disciplinary risk flagging;
- termination support;
- internal mobility;

- workforce planning;
- contractor selection;
- self-employment platform access.

Some of these uses may appear administrative.

Some may appear supportive.

Some may be framed as efficiency tools.

But in employment, even supportive tools can become consequential.

A ranking can determine who receives attention.

A score can determine who is shortlisted.

A productivity metric can shape management perception.

A monitoring alert can trigger scrutiny.

A scheduling system can affect working conditions.

A performance recommendation can influence promotion.

A task allocation tool can affect workload, opportunity and stress.

A termination-support tool can affect livelihood.

This is why HR AI governance cannot be reduced to whether the system makes the final decision.

The more important question is whether the AI system shapes the decision environment.

An organization may say:

The AI system does not decide.

A human makes the final decision.

That may be formally true.

But the AI system may still influence which candidates are seen, which workers are flagged, which evidence is emphasized, which options are available, which cases are escalated and which outcomes feel normal.

This means human oversight must be meaningful.

It must not simply confirm an AI-shaped pathway.

For Article 26 deployers, the central HR question is:

Can the organization prove that the AI-supported employment workflow was used under valid control conditions before it affected a person's work-related opportunity, treatment or outcome?

This requires evidence of:

- appropriate use;
- intended purpose;
- actual use;
- input data suitability;
- human oversight;
- worker or applicant information;
- monitoring;
- logs;
- escalation;
- authority;
- contestability;
- decision trail.

HR AI is especially sensitive because employment decisions often involve power imbalance.

Applicants may not know how they were assessed.

Workers may fear challenging management systems.

Managers may over-trust algorithmic outputs.

HR teams may be under pressure to process candidates quickly.

Productivity tools may create surveillance pressure.

AI-generated metrics may appear more objective than they are.

This makes the deployer's responsibility more serious.

An employer deploying high-risk AI systems in the workplace must not only ask whether the system is efficient.

It must ask whether the system is governable, explainable enough for its purpose, contestable, monitored and subject to real human control.

The core risk is not only that the system may be wrong.

The core risk is that the system may become authoritative without the organization recognizing that authority has shifted.

A system introduced to assist recruitment may become the practical gatekeeper.

A monitoring tool introduced to support productivity may become a disciplinary signal.

A task allocation tool introduced for efficiency may affect fairness, workload and opportunity.

A performance system introduced for analytics may become a management decision layer.

This is why employment AI should be governed through evidence of control.

The organization should be able to show not only that AI was used, but that its use remained within valid boundaries.

17.2 Recruitment, Scoring and Worker Evaluation

Recruitment is one of the most visible areas of HR AI risk.

AI systems may help source candidates, screen applications, rank CVs, match skills, analyze interviews, generate shortlist recommendations or draft candidate evaluations.

These tools may appear helpful.

They may reduce administrative burden.

They may process large volumes of applications.

They may create consistency.

They may support recruiters.

But they can also shape access to opportunity.

A candidate who is ranked low may never be reviewed carefully.

A candidate whose CV is summarized poorly may be misunderstood.

A candidate whose experience does not match historical patterns may be filtered out.

A candidate who lacks certain keywords may be deprioritized.

A candidate from a non-traditional background may be scored unfairly.

A candidate with career gaps, migration history, disability-related gaps or unusual work history may be misrepresented by automated screening.

The danger is that the AI system may be treated as neutral because it appears technical.

But recruitment is not a neutral process.

It involves interpretation, context, judgment, opportunity and fairness.

For deployers, recruitment AI requires strong controls.

The organization should define:

- what the system is allowed to do;
- what the system is not allowed to do;
- what data it may use;
- whether protected or sensitive characteristics are excluded or controlled;
- whether proxy variables are assessed;
- how human review occurs;
- whether candidates can be reconsidered;
- whether the output is advisory or decisive;
- whether rejected candidates were meaningfully reviewed;
- how ranking or scoring is monitored;
- what evidence is retained.

The same applies to worker evaluation.

AI systems may analyze performance, productivity, communication patterns, task completion, customer feedback, attendance, schedule adherence, quality metrics or behavioural signals.

These outputs can affect promotions, bonuses, warnings, discipline, training, work allocation or termination.

The deployer must therefore ask:

What does the system measure?

Does the metric actually represent performance?

What context is missing?

Does the system treat different roles, teams or working conditions fairly?

Are workers able to explain anomalies?

Are managers trained not to over-rely on the system?

Can the system's output be challenged?

Are decisions based on AI-supported metrics reviewed by humans with real authority?

This is especially important because worker evaluation data can be narrow.

A productivity metric may not reflect difficulty of work.

A customer rating may reflect bias.

A communication metric may not reflect quality.

A task completion rate may not reflect complexity.

An absence metric may not reflect legitimate circumstances.

A performance score may not reflect teamwork, mentoring, reliability or context.

The AI system may transform partial data into a confident evaluation.

This is a governance risk.

For Article 26, deployers should preserve evidence that recruitment and worker evaluation systems were used under valid conditions.

This evidence may include:

- provider instructions;
- approved use case;
- role and job-family mapping;
- data source records;
- input data checks;
- exclusion of prohibited or inappropriate data;
- human review records;
- shortlist review evidence;
- candidate reconsideration procedure;
- worker feedback process;
- override logs;
- escalation records;
- monitoring of outcome patterns;
- bias and impact review;
- communication to workers and representatives;
- decision trail connecting AI output to final outcome.

The key question is not only:

Did a human make the final employment decision?

The stronger question is:

Did the human decision remain meaningfully independent of AI-shaped ranking, scoring or evaluation?

This is where human-in-control becomes essential.

17.3 Human Oversight in Employment Decisions

Human oversight in employment AI is difficult because employment environments contain power, pressure and incentives.

A recruiter may be under pressure to process many applications quickly.

A manager may prefer simple performance scores.

An HR team may rely on system outputs because they appear consistent.

A worker may fear challenging an AI-supported evaluation.

A candidate may never know an AI system influenced their rejection.

A human reviewer may technically be present but practically dependent on the AI output.

This is why human oversight in employment should be designed carefully.

The deployer should not assume that human involvement automatically protects candidates or workers.

The organization must ask whether the human can actually challenge the AI output.

A meaningful human reviewer should have:

- competence to understand the employment process;
- training on the AI system's role and limitations;
- access to relevant evidence;
- ability to view source data;
- authority to override;
- time to review;
- ability to request additional context;
- responsibility to consider non-AI evidence;
- escalation pathway;
- protection against pressure to rubber-stamp system outputs.

Human oversight should occur before the employment consequence becomes difficult to reverse.

For recruitment, this may mean before a candidate is rejected, not only before the final hire.

For promotion, this may mean before a shortlist is finalized.

For performance evaluation, this may mean before the AI-supported assessment affects pay, discipline or career progression.

For termination support, this may mean before any decision pathway is initiated.

For worker monitoring, this may mean before an alert creates disciplinary scrutiny.

The timing matters.

A human review after an applicant has already been filtered out may not provide meaningful control.

A human review after a worker has already been flagged may not erase the reputational effect.

A human review after a score has already shaped management perception may not restore neutrality.

The deployer should therefore identify the real consequence points in employment workflows.

A consequence point may occur earlier than the formal final decision.

Examples include:

- candidate removed from consideration;
- candidate placed below review threshold;
- worker flagged as low performer;
- productivity alert sent to manager;
- promotion shortlist generated;
- shift allocation finalized;
- disciplinary case opened;
- training or development path assigned;
- internal mobility recommendation made;
- contractor access denied.

At each consequence point, the deployer should ask:

Is human oversight required here?

If yes:

Was it performed?

Was it meaningful?

Was it recorded?

Could the human change the outcome?

Did the human have access to context?

Was the affected person able to provide relevant information?

This last question is important.

Employment decisions often require context that AI systems may not capture.

A candidate's unusual career path may have a legitimate explanation.

A worker's productivity data may reflect equipment failure, team changes, health-related accommodation, caregiving needs, training gaps, difficult customers or unusual task complexity.

A scheduling pattern may affect workers differently depending on personal circumstances.

A task allocation system may unintentionally reduce opportunity for certain workers.

Human oversight should allow this context to enter the process.

Otherwise, the AI system may narrow the decision environment before the human acts.

For deployers, evidence of employment oversight should include:

- review before consequence;
- reviewer role and authority;
- information shown to reviewer;
- source data access;
- override capability;
- reason for accepting or rejecting AI output;
- worker or candidate input where relevant;
- escalation records;
- final decision record;
- evidence that AI output was not treated as automatically determinative.

The core test is simple:

Could the human meaningfully disagree with the AI system?

If not, human oversight may be present in form but absent in substance.

17.4 Bias, Input Data and Affected-Person Rights

Employment AI raises serious questions around bias, input data and affected-person rights.

The risk is not only that the model may be biased.

The risk is that the entire employment workflow may contain historical, structural or operational bias, and the AI system may reproduce or amplify it.

Recruitment data may reflect past hiring preferences.

Performance data may reflect unequal opportunity.

Productivity data may reflect role differences rather than effort.

Customer feedback may reflect discriminatory attitudes.

Manager evaluations may reflect subjective bias.

Absence data may reflect health, caregiving or accommodation-related context.

Promotion history may reflect previous inequality.

If an AI system learns from or relies on such data, the output may appear objective while carrying historical patterns.

For deployers, this creates an input data responsibility.

The organization should ask:

What data is used?

Why is it relevant?

Is it sufficiently representative?

Does it contain historical bias?

Does it include proxy variables?

Does it disadvantage non-traditional candidates?

Does it reflect opportunity differences?

Does it reflect manager bias?

Does it reflect customer bias?

Does it reflect protected or sensitive characteristics directly or indirectly?

Does it include context needed to interpret the data fairly?

The system's intended purpose matters.

A dataset suitable for workforce planning may not be suitable for individual performance evaluation.

A dataset suitable for training recommendations may not be suitable for promotion decisions.

A dataset suitable for job matching may not be suitable for candidate rejection.

A dataset suitable for general analytics may not be suitable for disciplinary processes.

The deployer must ensure that data use matches purpose.

This is especially important because employment consequences can be severe.

A biased or poorly contextualized AI system can affect hiring, pay, promotion, working conditions, discipline or termination.

The affected person may not know why the outcome occurred.

This connects to affected-person information and contestability.

Workers and candidates should not be left facing an AI-supported process that is invisible, unexplained and unchallengeable.

Depending on the specific use case and applicable law, the organization may need to provide information about AI involvement and preserve channels for questions, review or challenge.

From a governance perspective, contestability is not only a rights issue.

It is also a quality-control mechanism.

Affected persons may identify missing context, incorrect data, misleading assumptions or unfair outcomes.

A worker may know that the data used to evaluate them is incomplete.

A candidate may know that the system misread their experience.

A manager may know that a productivity metric lacks context.

A union or worker representative may identify systemic concerns.

These signals should not be treated only as complaints.

They are monitoring inputs.

A mature deployer should therefore connect affected-person feedback to AI governance.

This may include:

- candidate review requests;
- worker concerns;
- manager feedback;
- representative feedback;
- data correction requests;
- outcome pattern analysis;
- appeal records;
- investigation outcomes;
- system improvement actions;
- bias review updates.

The organization should preserve evidence that such feedback was considered.

This helps show that the deployer did not treat the AI system as closed and unquestionable.

Bias control should also be monitored over time.

Even if a system is assessed before deployment, actual outcomes may change.

Recruitment pools change.

Job requirements change.

Managers change.

Worker populations change.

Business priorities change.

Data collection changes.

System configurations change.

This means bias and impact review should not be one-time exercises.

The deployer should monitor patterns, especially where AI output affects access, opportunity or treatment.

Useful monitoring signals may include:

selection rates by relevant groups where legally and ethically appropriate;
 override rates;
 appeal outcomes;
 false positive and false negative patterns;
 manager reliance patterns;
 candidate drop-off;
 worker complaint patterns;
 outcome disparities;
 repeated data quality issues;
 system use outside intended purpose.

The goal is not to reduce human beings to metrics.

The goal is to detect whether the AI-supported workflow is producing patterns that require review.

For Article 26, the core point is that input data relevance and representativeness are not abstract technical concerns.

In employment, they directly affect people's livelihoods.

17.5 HR Deployer Readiness Checklist

Organizations deploying AI systems in HR and employment should translate Article 26 obligations into practical readiness checks.

The following checklist provides a governance-oriented structure.

1. System Identification

The organization should identify all AI systems used in HR, recruitment, worker management or access to self-employment.

For each system, record:

system name;
 provider;
 internal owner;
 HR process affected;
 intended purpose;
 actual use;
 user groups;
 affected persons;

- data sources;
- output type;
- decision influence;
- risk classification;
- deployment status.

2. Employment Use-Case Classification

The organization should classify the employment use case.

Examples include:

- candidate sourcing;
- CV screening;
- applicant ranking;
- interview support;
- promotion support;
- performance evaluation;
- monitoring;
- task allocation;
- scheduling;
- training recommendation;
- disciplinary support;
- termination support;
- contractor selection;
- platform access.

For each use case, the organization should identify the consequence level.

3. Intended Purpose and Actual Use

The organization should compare provider intended purpose with actual HR use.

Evidence should show:

- approved use case;
- prohibited uses;
- actual workflow;
- users;
- affected persons;
- output role;
- decision influence;
- scope change history;

reassessment triggers.

4. Worker and Representative Information

Where required, the organization should inform workers' representatives and affected workers before putting the high-risk AI system into service or using it.

Evidence should include:

- notice text;
- date of notice;
- recipients;
- worker representative communication;
- explanation of AI role;
- route for questions;
- updates where use changes;
- acknowledgement where appropriate.

5. Candidate or Affected-Person Information

Where applicable, the organization should inform natural persons when a high-risk AI system is used to make or assist decisions relating to them.

Evidence should include:

- information provided;
- process affected;
- AI role;
- human oversight explanation;
- review or challenge route;
- timing of notice;
- records of responses or concerns.

6. Human Oversight

The organization should define meaningful oversight.

Evidence should show:

- reviewer role;
- competence;
- training;
- authority;
- access to source evidence;
- ability to override;

- review timing;
- escalation route;
- intervention logs;
- review quality checks.

7. Input Data Governance

The organization should assess input data relevance and representativeness.

Evidence should include:

- data source mapping;
- data quality checks;
- missing-data rules;
- proxy variable analysis;
- representativeness review;
- freshness thresholds;
- protected-characteristic safeguards;
- historical bias review;
- purpose-fit assessment.

8. Decision Influence Mapping

The organization should map how AI output influences employment decisions.

Evidence should show whether the AI output:

- screens;
- ranks;
- scores;
- summarizes;
- recommends;
- flags;
- routes;
- drafts;
- triggers review;
- affects final decision.

The organization should avoid vague descriptions such as “decision support” unless the actual influence is clearly mapped.

9. Monitoring and Outcome Review

The organization should monitor AI-supported HR workflows.

Monitoring should include:

- outcome patterns;
- override rates;
- appeal rates;
- complaint patterns;
- group-level impacts where appropriate;
- drift from intended purpose;
- increased reliance;
- human review quality;
- data quality issues;
- escalation trends.

10. Logs and Decision Trails

The organization should retain logs that support reconstructability.

A decision trail should show:

- system used;
- input data condition;
- AI output;
- human review;
- authority state;
- worker or candidate information;
- override or escalation;
- final decision;
- reason continuation was allowed.

11. Contestability and Review

The organization should provide appropriate routes for challenge or review.

This may include:

- candidate reconsideration;
- worker concern process;
- data correction route;
- manager review;
- HR escalation;
- representative engagement;
- appeal outcome records.

12. Reassessment Triggers

The organization should reassess the system when:

- use case changes;
- output becomes more influential;
- new worker groups are affected;
- data sources change;
- provider instructions change;
- monitoring identifies disparity;
- complaints increase;
- human oversight weakens;
- downstream consequences expand;
- regulatory or internal policy expectations change.

13. Evidence Pack

For each HR high-risk AI system, the organization should maintain an evidence pack showing:

- what the system does;
- why it is used;
- who is affected;
- what data it uses;
- what decisions it influences;
- what human oversight exists;
- what information was provided;
- what logs are retained;
- what monitoring occurs;
- how affected persons can raise concerns;
- how the organization knows use remains within valid boundaries.

This checklist helps translate Article 26 into HR operational governance.

It does not eliminate the complexity of employment AI.

But it makes the control problem visible.

Chapter 17 Summary

Human resources and employment are among the most sensitive areas for Article 26 deployer governance.

AI systems used in recruitment, selection, promotion, termination, task allocation, monitoring and worker evaluation can affect career prospects, livelihoods and workers' rights.

The central question is not only whether the AI system makes the final decision.

The central question is whether the AI system shapes the employment decision environment.

A ranking, score, summary, alert or recommendation can influence opportunity, treatment or outcome even when a human signs the final decision.

This chapter argues that employment AI requires strong evidence of control.

Deployers must be able to show appropriate use, intended-purpose alignment, input data suitability, meaningful human oversight, worker or affected-person information, monitoring, logs, escalation and contestability.

Human oversight is especially important in employment because of power imbalance and automation bias.

A human reviewer must have competence, authority, time, evidence access and practical ability to challenge the AI output before employment consequence attaches.

Input data governance is also critical.

Employment data may reflect historical bias, unequal opportunity, subjective judgment, customer bias, manager bias or missing context. A dataset suitable for workforce analytics may not be suitable for individual employment decisions.

The chapter emphasizes that affected-person feedback and contestability should be treated as governance signals, not only complaints.

Workers, candidates and representatives may identify missing data, unfair assumptions or unintended system effects.

Finally, the chapter provides an HR deployer readiness checklist covering system identification, use-case classification, worker information, human oversight, input data, decision influence, monitoring, logs, contestability, reassessment and evidence packs.

The central HR governance question is:

Can the organization prove that AI-supported employment decisions remained within valid, transparent, reviewable and contestable control conditions before affecting a person's opportunity, treatment or livelihood?

That is the practical meaning of Article 26 in employment.

Chapter 18

Procurement and Supplier Risk

Legal anchor and framework boundary. [Article 13](#); [Article 26](#). Procurement controls are operational recommendations for preserving the conditions needed for compliant deployment.

18.1 Procurement Agents and Supplier Recommendations

Procurement is a natural environment for AI adoption.

Organizations need to evaluate suppliers, compare proposals, manage contracts, monitor risk, reduce cost, assess performance, detect conflicts, review sanctions exposure, manage compliance and support decision-making across large volumes of information.

AI systems may help with:

- supplier discovery;
- request-for-proposal analysis;
- bid comparison;
- contract review;
- supplier scoring;
- risk assessment;
- sanctions screening support;
- financial health monitoring;
- ESG and sustainability review;
- conflict-of-interest detection;
- vendor due diligence;
- procurement policy compliance;
- fraud detection;
- supplier performance monitoring;
- negotiation support;
- spend analysis;
- workflow automation;
- approval routing;
- renewal recommendations.

Some of these systems may appear low-risk at first.

They may be described as productivity tools.

They may summarize supplier documents.

They may compare vendor responses.

They may draft procurement recommendations.

They may flag missing information.

They may help prepare internal decision packs.

But procurement can become high-consequence quickly.

Supplier decisions affect money, operational resilience, compliance exposure, cybersecurity, safety, supply-chain continuity, public trust, customer delivery, anti-corruption controls and regulatory risk.

In public-sector procurement, supplier decisions may also affect public accountability, fairness, competition and use of public funds.

In regulated sectors, supplier selection may affect outsourcing obligations, operational resilience, data protection, cybersecurity, financial crime controls and critical third-party risk.

This means procurement AI should not be treated only as a cost-efficiency tool.

It can become part of the organization's control environment.

A procurement AI system may not sign the contract.

But it may shape which suppliers are considered.

It may summarize evidence.

It may rank bids.

It may produce risk scores.

It may recommend exclusion.

It may suggest preferred vendors.

It may generate the reasoning used in approval papers.

It may route the workflow to the next authority level.

It may trigger additional due diligence.

It may create a default decision path.

For deployers, the key question is:

Can the organization prove that AI-supported supplier recommendations remained within valid control conditions before they influenced procurement consequence?

This is not only a technical question.

It is a governance question.

The deployer should know:

What supplier data was used?

Was the data current?

Was the supplier risk profile complete?

Was sanctions or exclusion data checked?

Was financial health data fresh?

Was cybersecurity risk assessed?

Was conflict-of-interest information considered?

Was the AI output advisory or authoritative?

Who reviewed the recommendation?

Did the reviewer have procurement authority?

Was the recommendation used in a regulated or high-value process?

Did the output trigger downstream approval?

Were logs retained?

Could the decision trail be reconstructed?

Procurement AI creates particular risk because supplier recommendations often rely on multiple evidence sources.

A supplier profile may include:

- company registration information;
- financial statements;
- credit scores;
- sanctions or exclusion lists;
- litigation history;
- insurance records;
- security certifications;
- data protection posture;

- service performance;
- delivery history;
- pricing;
- sustainability information;
- subcontractor dependencies;
- geographic risk;
- ownership structure;
- conflict-of-interest declarations;
- previous incidents;
- audit results;
- contractual terms.

If the AI system uses only part of this evidence, or if some evidence is stale, the recommendation may appear complete while being incomplete.

This is especially dangerous when the output is presented in polished language.

A well-written supplier recommendation may create confidence even when the evidence behind it is partial.

For this reason, procurement deployers should not evaluate only the quality of AI-generated text.

They should evaluate the evidence condition behind the recommendation.

The central procurement governance question is:

Does the evidence support the authority attached to the recommendation?

If a supplier recommendation is used only for early internal triage, the evidence threshold may be lower.

If it influences final supplier approval, high-value contract award, regulated outsourcing, critical infrastructure, sensitive data processing or public procurement, the evidence threshold must be higher.

Control should follow consequence.

This principle is essential for procurement AI.

18.2 Approval Validity and Changing Risk Context

Procurement decisions depend on approvals.

Approvals depend on authority.

Authority depends on current conditions.

This creates a key risk for AI-supported procurement workflows:

An approval may remain visible after the conditions that made it valid have changed.

For example:

A supplier may have been approved when financial risk was low.

A risk threshold may change after the approval.

A sanctions list may be updated.

A cybersecurity incident may occur.

A supplier may lose certification.

A subcontractor may change.

A public procurement rule may be updated.

A procurement manager may move to a different role.

A budget threshold may be exceeded.

A contract scope may expand.

A low-risk purchase may become a critical dependency.

The AI-supported workflow may continue to rely on a previous approval state, but the approval may no longer support the current procurement consequence.

This is approval validity risk.

It is closely connected to evidence freshness and authority.

A procurement approval should not be treated as permanently valid unless the conditions supporting it remain valid.

The deployer should ask:

Who approved the supplier?

What was the approval based on?

What authority did the approver have?

What contract value or risk level was approved?

What data was current at the time?

What conditions or limitations applied?

Has the supplier risk profile changed?

Has contract scope changed?

Has regulatory context changed?

Has the supplier's role become more critical?

Has the AI system used the approval for a different purpose?

Procurement AI systems can intensify this risk because they may reuse historical approvals, supplier scores or previous assessment records.

This can be useful.

But it can also create inherited confidence.

A supplier score from a previous procurement may not be valid for a new procurement.

An approval for a low-value service may not be valid for a high-value strategic contract.

An approval for a non-critical supplier may not be valid for a supplier handling sensitive data.

An approval for one jurisdiction may not be valid for another.

An approval before a cybersecurity incident may not be valid after the incident.

An approval before a financial downgrade may not be valid after the downgrade.

This is why procurement workflows should include approval freshness checks.

Approval freshness is the question of whether the approval still applies to the current supplier, contract, scope, value, risk and timing.

A deployer should define approval validity rules.

These may include:

- approval expiry dates;
- contract value thresholds;
- risk category thresholds;
- event-based reassessment triggers;
- supplier status triggers;
- sanctions or exclusion triggers;
- cybersecurity incident triggers;
- financial health triggers;

- ownership change triggers;
- scope expansion triggers;
- subcontractor change triggers;
- regulatory change triggers;
- new data-processing or security requirements;
- public procurement compliance triggers.

If one of these triggers occurs, the AI-supported recommendation should not automatically proceed on the basis of old approval.

The workflow should enter HOLD or ESCALATE.

For example:

HOLD: supplier financial data is older than the required threshold.

HOLD: supplier approval does not cover the increased contract value.

ESCALATE: supplier now processes sensitive personal data and requires data protection review.

ESCALATE: supplier has unresolved cybersecurity incident.

REFUSE: supplier is listed on an exclusion or sanctions list.

ALLOW: approval current, risk profile unchanged, data fresh, human review completed.

This transforms procurement approval from a static record into a runtime control condition.

The key point is that procurement risk is dynamic.

Supplier risk changes.

Contract context changes.

Organizational reliance changes.

Regulatory expectations change.

AI-supported procurement workflows must therefore be designed to detect when old approval no longer supports new consequence.

18.3 Supplier Data Freshness

Supplier risk is highly dependent on data freshness.

A supplier may be acceptable today and unacceptable tomorrow.

The supplier's financial position may deteriorate.

A sanctions listing may appear.

A legal dispute may emerge.

A cybersecurity breach may be disclosed.

A certification may expire.

A subcontractor may change.

Insurance may lapse.

Ownership may change.

Performance may decline.

A public controversy may affect reputational risk.

A regulator may issue a warning.

An internal incident may occur.

This means supplier recommendations require freshness discipline.

A procurement AI system can produce a supplier recommendation only as reliable as the evidence it uses.

The danger is that supplier data often looks stable.

Supplier records may appear complete because they contain many fields.

But completeness is not freshness.

A supplier profile may contain financial data, certifications, legal records, contract history and performance scores. But if key fields are outdated, the recommendation may be misleading.

For high-value or high-risk procurement, deployers should define supplier data freshness thresholds.

These thresholds may vary by data type.

For example:

Sanctions and exclusion checks may need to be current at the point of decision.

Financial health data may require recent update or event-based review.

Cybersecurity posture may require current certification and recent incident checks.

Insurance coverage may require active validity.

Data protection assessments may require review when processing scope changes.

Supplier performance data may require recent operational records.

Ownership and beneficial ownership data may require periodic and event-based review.

Subcontractor information may require validation when supplier delivery model changes.

ESG or sustainability claims may require source verification.

Contract risk may require review when scope, value or jurisdiction changes.

A procurement AI system should not treat all supplier data as equally fresh.

The workflow should show freshness status for each critical evidence category.

For example:

Sanctions check: current.

Financial data: 280 days old; review required.

Cybersecurity certification: expired.

Insurance: valid.

Performance data: current.

Subcontractor data: unknown.

Conflict-of-interest declaration: missing.

In a low-risk early sourcing stage, such warnings may allow continued exploratory review.

In a final approval stage, the same warnings may require HOLD.

This is why freshness must be linked to workflow stage and consequence level.

Procurement deployers should also distinguish between internal and external supplier data.

Internal data may include:

- previous contracts;
- delivery performance;

- incident history;
- payment records;
- internal risk assessments;
- user feedback;
- audit findings;
- service-level performance;
- internal approvals;
- relationship history.

External data may include:

- public registers;
- sanctions lists;
- credit ratings;
- litigation records;
- financial statements;
- certifications;
- cybersecurity ratings;
- media reports;
- ownership records;
- regulatory notices;
- third-party risk intelligence.

Each source has different reliability and update frequency.

The deployer should know which source is authoritative for which purpose.

A current but unreliable source should not override a reliable source without review.

A model-generated summary of supplier risk should not replace source verification where consequence is high.

Supplier data freshness should also be visible to human reviewers.

A procurement manager reviewing an AI-generated recommendation should see:

What evidence was used?

When was it refreshed?

Which sources were missing?

Which fields were stale?

Which warnings were active?

Which assumptions were made?

Which checks passed?

Which checks require review?

Without this visibility, human oversight may be weak.

The reviewer may approve a recommendation without knowing that the evidence environment has degraded.

This is why supplier data freshness is not only a data governance issue.

It is a human oversight issue.

It is also a log-retention issue.

The decision trail should record the supplier data condition at the time of recommendation and approval.

If a supplier decision later comes under review, the organization should be able to show whether key evidence was fresh enough at the point of consequence.

This is the procurement version of evidence of control.

18.4 Authority and Budget Thresholds

Procurement authority is often structured around thresholds.

Different people or committees may have authority depending on:

- contract value;
- supplier risk level;
- category of goods or services;
- strategic importance;
- data sensitivity;
- outsourcing status;
- public procurement rules;
- cybersecurity exposure;
- jurisdiction;
- contract duration;
- criticality;
- operational dependency;
- financial exposure.

AI-supported procurement workflows must respect these thresholds.

A recommendation that is valid for a low-value purchase may not be valid for a high-value contract.

A manager who can approve one category may not approve another.

A buyer who can select routine suppliers may not approve regulated outsourcing.

A procurement officer who can approve local vendors may not approve cross-border data-processing suppliers.

A department head who can approve an internal purchase may not approve a critical infrastructure dependency.

This means procurement AI systems must be aware of authority boundaries.

If the AI system recommends a supplier, routes an approval, drafts a procurement decision or prepares a decision pack, the workflow should know which authority threshold applies.

The central question is:

Does the current approver have authority for this supplier, contract, risk level and consequence?

If yes, the workflow may continue.

If no, escalation is required.

If unclear, HOLD is appropriate.

This is especially important where AI systems generate procurement recommendations that appear ready for approval.

A polished recommendation may hide the fact that the contract exceeds the approver's authority.

A supplier may appear low-risk in the AI summary, but the contract category may require specialist review.

A recommendation may omit that the supplier processes sensitive data, triggering additional review.

A cost-saving recommendation may ignore outsourcing or resilience thresholds.

A supplier selection may fall within budget but outside cybersecurity authority.

The deployer should therefore connect procurement AI to authority matrices.

Authority checks may include:

- budget threshold;
- supplier risk tier;
- contract category;
- data processing category;
- cybersecurity classification;
- outsourcing classification;
- public procurement route;
- approval committee requirement;
- legal review requirement;
- compliance review requirement;
- information security review requirement;
- finance approval;
- senior management approval;
- board approval where relevant.

These checks should be reflected in workflow design.

The AI system should not only recommend the best supplier.

It should identify the required control pathway for the recommendation.

For example:

Recommendation: supplier selected for initial review.

Required controls: financial check, sanctions check, cybersecurity review, procurement manager approval.

Recommendation: preferred supplier for high-value contract.

Required controls: procurement committee, legal review, finance approval, risk review, executive approval.

Recommendation: supplier handles personal data.

Required controls: data protection review, information security assessment, contract clauses, transfer assessment where relevant.

Recommendation: critical outsourced service.

Required controls: operational resilience review, exit plan, service-level controls, senior approval.

This turns procurement AI from a ranking tool into a controlled decision-support workflow.

Authority must also be rechecked when contract conditions change.

A supplier recommendation may begin below a threshold but later exceed it.

Scope may expand.

Contract duration may increase.

Service category may change.

Data sensitivity may increase.

The supplier may become operationally critical.

In such cases, authority must be reassessed.

A deployer should preserve evidence that authority thresholds were checked at the relevant point.

The log should show:

- contract value;
- risk tier;
- category;
- required approval level;
- actual approver;
- authority validation;
- escalation if threshold exceeded;
- final decision.

This prevents AI-supported procurement from bypassing internal authority structures.

In procurement, governance failure often occurs not because no one approved, but because the wrong level of authority approved under changed conditions.

AI systems should not make that easier.

They should help make authority visible.

18.5 Procurement Control Evidence Map

A procurement control evidence map helps the deployer connect AI-supported supplier workflows to evidence of control.

The purpose is to make the decision trail visible.

For each procurement AI use case, the organization should identify:

- the AI system;
- the workflow;
- the supplier;
- the input evidence;
- the output;
- the human reviewer;
- the authority state;
- the monitoring signals;
- the escalation conditions;
- the logs;
- the final consequence.

A practical procurement evidence map may include the following categories.

1. System and Use Case

Evidence should show:

- system name;
- provider;
- intended purpose;
- approved procurement use case;
- actual use;
- system version;
- model version where relevant;
- user group;
- workflow stage;
- procurement category;
- contract type.

This helps show whether the AI system was used within its approved procurement scope.

2. Supplier Identity and Context

Evidence should show:

- supplier name;
- legal entity;
- jurisdiction;
- ownership information where relevant;

- contract scope;
- contract value;
- service category;
- criticality;
- data sensitivity;
- outsourcing status;
- subcontractor dependencies;
- previous relationship;
- performance history.

This helps determine consequence and required controls.

3. Input Evidence Condition

Evidence should show:

- supplier data sources;
- financial data freshness;
- sanctions or exclusion check status;
- cybersecurity assessment status;
- insurance status;
- certification status;
- legal or litigation indicators;
- performance data;
- ESG or sustainability evidence;
- conflict-of-interest information;
- data protection assessment where relevant;
- missing evidence;
- stale evidence warnings.

This helps prove whether the recommendation was based on valid supplier evidence.

4. AI Output

Evidence should show:

- AI-generated summary;
- ranking;
- score;
- recommendation;
- risk category;
- supporting reasons;

- uncertainty indicators;
- warnings;
- omitted or missing evidence where detected;
- limitations displayed to user.

This helps show what the AI system contributed.

5. Human Oversight

Evidence should show:

- reviewer identity or role;
- reviewer competence;
- information shown to reviewer;
- review timestamp;
- questions raised;
- modifications made;
- override decision;
- request for further evidence;
- escalation;
- final reviewer conclusion.

This helps prove that human oversight was meaningful.

6. Authority and Approval

Evidence should show:

- required approval level;
- approver identity or role;
- approval authority;
- contract value threshold;
- risk threshold;
- policy version;
- delegation status;
- approval timestamp;
- approval limitations;
- escalation if authority exceeded.

This helps show whether the workflow was authorized.

7. Monitoring and Escalation

Evidence should show:

active alerts;
stale evidence triggers;
unresolved supplier risks;
conflict indicators;
policy mismatches;
use outside intended purpose;
high-risk supplier status;
escalation route;
escalation outcome;
suspension or refusal where applicable.

This helps show whether monitoring connected to action.

8. Downstream Consequence

Evidence should show:

supplier shortlisted;
supplier approved;
contract awarded;
due diligence triggered;
contract renewal recommended;
external communication sent;
purchase order created;
workflow closed;
supplier rejected;
case escalated.

This helps show where the AI output influenced consequence.

9. Decision Trail

The final decision trail should connect:

AI system,
supplier evidence,
AI output,
human review,
authority check,
risk controls,

approval,

and final procurement action.

This decision trail should answer:

Why was this supplier recommendation allowed to influence this procurement decision?

This is the core evidence question.

A procurement AI system should not only produce a recommendation.

It should support a reconstructable supplier decision.

For high-value, regulated, public-sector or critical procurement, this evidence map becomes especially important.

The deployer should be able to show:

We knew what evidence was used.

We knew whether the evidence was fresh.

We knew what the AI system recommended.

We knew who reviewed it.

We knew who had authority.

We knew what risks were present.

We knew whether escalation was required.

We knew why the workflow continued.

This is procurement evidence of control.

Chapter 18 Summary

Procurement and supplier risk provide a strong practical example of Article 26 deployer governance.

AI systems may support supplier discovery, bid comparison, risk scoring, contract review, sanctions screening, supplier monitoring, approval routing and procurement recommendations.

These uses may improve efficiency, but they can also influence high-consequence decisions involving money, operational resilience, cybersecurity, compliance, public accountability, critical third parties and regulated outsourcing.

The central procurement governance question is:

Can the organization prove that AI-supported supplier recommendations remained within valid control conditions before they influenced procurement consequence?

This chapter explains why procurement AI requires careful control of supplier evidence, approval validity, authority thresholds and decision trails.

Supplier risk changes over time.

A supplier may become financially weaker, lose certification, appear on an exclusion list, suffer a cybersecurity incident, change ownership, alter subcontractors or become more operationally critical.

For this reason, supplier data freshness is essential.

A recommendation based on stale supplier evidence may appear reliable while the decision environment has already degraded.

The chapter also explains approval validity.

A previous supplier approval may not remain valid if contract value, risk category, scope, data sensitivity, outsourcing status or regulatory context changes.

Procurement authority must therefore be checked at runtime.

The organization should know whether the current approver has authority for the supplier, contract, risk level and consequence.

Finally, the chapter provides a procurement control evidence map.

This map connects system identity, supplier context, input evidence, AI output, human oversight, authority, monitoring, escalation, downstream consequence and final decision trail.

The procurement version of evidence of control is simple:

The organization should be able to reconstruct why a supplier recommendation was allowed to influence a procurement decision.

If that cannot be shown, the AI-supported procurement workflow may not be sufficiently governed.

Chapter 19

Healthcare and Clinical Decision Support

Legal anchor and framework boundary. [Article 14](#); [Article 26](#). Healthcare and medical-device contexts require separate sector-specific legal, clinical and safety review.

19.1 Medical AI and Clinical Influence

Healthcare is one of the most sensitive environments for AI deployment.

The reason is simple.

Clinical decisions can affect life, health, safety, dignity, access to care, diagnosis, treatment, triage, prioritization and patient trust.

When AI systems enter healthcare workflows, they do not only process medical data.

They may influence clinical attention.

They may influence urgency.

They may influence interpretation.

They may influence diagnosis support.

They may influence triage.

They may influence treatment options.

They may influence patient communication.

They may influence how clinicians allocate time and concern.

This makes healthcare a critical use case for Article 26 deployer governance.

The central issue is not only whether an AI system is technically accurate.

The central issue is whether the system is used under valid clinical, operational and governance conditions when its output influences care.

A medical AI system may not make a final clinical decision.

But it may still shape the clinical decision environment.

It may summarize patient records.

It may flag risk.

It may prioritize cases.

It may recommend next steps.

It may draft clinical notes.

It may interpret imaging, test results or symptoms.

It may support triage.

It may identify possible diagnoses.

It may suggest care pathways.

It may generate patient-facing explanations.

Each of these outputs may influence what a clinician sees, considers, verifies or acts upon.

In healthcare, influence can matter even when authority formally remains human.

A clinician may remain responsible for the final decision, but the AI output may shape the evidence environment before the clinician acts.

This means healthcare deployers should not rely on the statement:

A human clinician makes the final decision.

That statement may be true, but incomplete.

The stronger governance question is:

Did the clinician remain meaningfully in control of the clinical judgment after AI shaped the information environment?

This question is central to Article 26.

Human oversight in healthcare must be more than human presence.

It must be clinically meaningful.

Input data must be current, relevant and complete enough for the intended purpose.

Monitoring must detect performance issues, data issues, workflow drift and unsafe reliance.

Logs must preserve enough context to reconstruct the AI-supported clinical workflow.

Escalation pathways must exist for malfunction, unsafe output, serious risk or incident.

The system must be used according to provider instructions and intended purpose.

This is especially important because healthcare AI may create false confidence.

A system may produce a fluent summary.

A model may generate a plausible differential.

A triage tool may assign priority.

A risk score may appear precise.

A recommendation may appear evidence-based.

But the output may depend on incomplete records, stale data, poor representativeness, missing clinical context or assumptions that do not fit the patient.

The presence of a clinically formatted output does not prove clinical validity.

This is one of the core governance risks in medical AI.

The deployer must ensure that the workflow does not confuse fluency with clinical reliability.

For Article 26, the deployer's responsibility is not to replace clinical judgment with compliance language.

It is to build an operational environment where clinical judgment can remain informed, independent, supported and capable of intervention.

The central healthcare governance question is:

Can the organization prove that the AI-supported clinical workflow remained under valid clinical and control conditions before the output influenced patient care?

19.2 Validation Context and Operational Authority

Medical AI validation is one of the most important issues in healthcare governance.

A system may perform well in a study, benchmark, validation exercise or controlled environment.

But deployers must ask whether that evidence supports the authority being attached to the system in actual clinical workflow.

This is a governance timing problem.

A research signal may become operationally influential before the validation context is mature enough to support the confidence attached to it.

A benchmark result may influence procurement.

A performance claim may influence clinical adoption.

A pilot result may influence workflow redesign.

A vendor evaluation may influence trust.

A published study may influence teaching, policy or deployment decisions.

But the deployer must ask:

What was actually measured?

Who judged the output?

What patient population was represented?

What data sources were used?

What exclusions applied?

How were unanswered or uncertain cases handled?

What failure cases were counted?

What clinical setting was tested?

What comparison baseline was used?

Was the system tested in the same environment where it will be deployed?

Was the system validated for the intended purpose?

Was it validated for the actual patient population?

Was it validated under real workflow pressure?

Was it validated for the consequence now attached to its output?

This matters because validation evidence does not automatically transfer across contexts.

A model may perform well in one dataset but not another.

A clinical decision-support tool may perform well in a controlled evaluation but not in a busy emergency department.

A triage system may perform differently across patient groups.

A summarization system may work well on complete records but poorly when recent data is missing.

An imaging support system may perform differently across devices, settings or populations.

A diagnostic assistant may produce useful suggestions for specialists but unsafe outputs for non-specialists.

A system tested as advisory may be used in practice as more authoritative.

The deployer must therefore distinguish between validation evidence and operational authority.

Validation evidence says:

The system performed under certain conditions.

Operational authority asks:

Is that evidence strong enough to justify the role the system now plays in this workflow?

This is especially important when AI output begins to shape clinical workflow.

A system used for background support may require one level of validation.

A system used for triage may require a much higher level.

A system used to suggest possible diagnoses may require careful clinical oversight.

A system used to influence treatment pathways may require strong evidence, monitoring and escalation.

A system used in emergency healthcare may require especially strict control conditions because timing, uncertainty and consequence are severe.

In healthcare, evidence maturity should determine authority.

A system should not gain more operational authority than its validation context can support.

For example:

If a system was validated only for summarization, it should not be used as a diagnostic authority.

If a system was validated only in one hospital environment, deployment in another environment should trigger reassessment.

If a system was validated only with complete structured records, use with incomplete or unstructured records should be treated cautiously.

If a system was validated for specialist review, use by non-specialists may require additional controls.

If a system was validated as advisory, downstream automation should not treat its output as decisive.

This is why validation context should travel with the system.

The deployer should preserve:

- validation scope;
- intended purpose;
- tested population;
- tested data sources;
- performance limits;
- known failure modes;
- user competence assumptions;
- human oversight assumptions;
- operational environment assumptions;
- excluded cases;
- uncertainty conditions;
- post-deployment monitoring expectations.

This validation context should inform deployment.

It should also inform runtime control.

A clinical AI output should not inherit authority from a headline, benchmark or vendor claim if the current workflow falls outside the conditions that made the evidence meaningful.

The deployer should therefore treat validation context as part of evidence of control.

The question is not only:

Which AI system performed better?

The stronger question is:

Is the evidence mature enough to justify the authority now being attached to the system in clinical workflow?

19.3 Human Oversight in Clinical Workflows

Human oversight in healthcare must be clinically meaningful.

A clinician cannot oversee an AI system properly if they do not understand the system's role, limitations, evidence base, input condition, output meaning or failure modes.

A clinician also cannot oversee the system properly if the workflow does not give enough time, context or authority to challenge the output.

This is why healthcare deployers must distinguish between clinician presence and clinician control.

A clinician may be present in the workflow.

A clinician may sign the final record.

A clinician may review the AI output.

But the key question is:

Could the clinician meaningfully disagree with the AI system before the output influenced care?

Meaningful clinical oversight requires several conditions.

First, the clinician must understand the intended purpose of the system.

A tool intended for summarization should not be treated as diagnostic.

A tool intended for triage support should not be treated as a final clinical decision-maker.

A tool intended for specialist use should not be deployed to non-specialists without additional safeguards.

Second, the clinician must understand the input data condition.

What records did the system use?

Were recent labs included?

Were imaging results included?

Was medication history current?

Were allergies current?

Was patient status updated?

Were relevant notes omitted?

Were external records available?

Were patient-reported symptoms captured?

If the clinician cannot see the input condition, oversight may be weak.

Third, the clinician must understand uncertainty.

The AI output should not appear more certain than the evidence supports.

A fluent summary may hide missing information.

A risk score may hide weak calibration.

A recommendation may hide poor evidence.

A triage priority may hide population-specific limitations.

The clinician needs visibility into uncertainty, limitations and warnings.

Fourth, the clinician must have practical authority.

They must be able to reject, modify, escalate or ignore the AI output.

They must be able to request additional evidence.

They must be able to slow the workflow when the AI output does not fit clinical judgment.

They must be supported by the organization when they challenge the system.

Fifth, the workflow must not create automation pressure.

If the interface makes AI acceptance easy and challenge difficult, oversight is weakened.

If clinical workload encourages reliance on AI summaries without checking source evidence, oversight is weakened.

If management treats AI recommendations as efficiency targets, oversight is weakened.

If clinicians are expected to approve outputs at high speed, oversight is weakened.

Sixth, the clinical review must occur before consequence attaches.

A review after patient routing, treatment recommendation, discharge communication or clinical documentation may be too late to function as preventive oversight.

The deployer should identify clinical consequence points, such as:

- triage priority assigned;

- patient routed to a care pathway;
- clinical summary entered into record;
- diagnostic suggestion presented;
- medication-related recommendation shown;
- discharge advice generated;
- patient-facing message sent;
- alert escalated;
- case deprioritized;
- treatment pathway influenced.

At these points, human oversight should be designed according to consequence.

Not every AI-assisted clinical output requires the same level of review.

But high-consequence outputs require stronger controls.

A deployer should also monitor clinical overreliance.

Signals may include:

- low override rates;
- rapid approvals;
- repeated acceptance of AI summaries without source review;
- reduced documentation of independent clinical reasoning;
- clinicians reporting time pressure;
- increased use outside intended purpose;
- AI output becoming default pathway;
- escalations declining unexpectedly;
- patient complaints related to AI-supported decisions.

Human oversight is not static.

It can degrade under pressure.

Clinical teams may initially review outputs carefully, then gradually trust the system more.

This can be useful if the system is reliable, but dangerous if trust exceeds evidence.

The deployer should therefore monitor the quality of oversight, not only its existence.

Evidence of meaningful clinical oversight may include:

- reviewer identity or role;
- clinician competence and training;

- output shown;
- input data condition shown;
- warnings shown;
- review action;
- modification or override;
- escalation;
- final clinical decision;
- documentation of independent judgment where relevant;
- evidence that the AI output did not bypass clinical responsibility.

The central clinical oversight question is:

Did the AI system support clinical judgment without replacing or silently narrowing it?

If the answer cannot be shown, the oversight model may be insufficient.

19.4 Benchmark Evidence Versus Deployment Evidence

Benchmarks are useful.

They can compare systems.

They can reveal progress.

They can show performance differences.

They can support procurement and research.

They can help identify promising tools.

But benchmark evidence is not the same as deployment evidence.

This distinction is critical in healthcare.

A benchmark may show that one system performs better than another on a set of tasks.

But deployment evidence asks whether the system works safely and appropriately in a specific clinical environment, with specific users, specific patients, specific data sources, specific workflows and specific consequences.

A benchmark may be controlled.

Deployment is messy.

A benchmark may use curated data.

Deployment uses real patient records.

A benchmark may involve selected tasks.

Deployment includes edge cases, missing data, time pressure and operational complexity.

A benchmark may score outputs after the fact.

Deployment attaches outputs to patient care.

A benchmark may evaluate correctness.

Deployment must also evaluate workflow influence, human oversight, clinical accountability, patient safety, incident response and evidence retention.

This is why deployers should not treat benchmark performance as automatic deployment readiness.

A strong benchmark result may justify further evaluation.

It should not automatically justify clinical authority.

For healthcare deployers, benchmark evidence should be translated into deployment questions.

What does the benchmark measure?

Does it match the intended clinical use?

Does it match the patient population?

Does it match the care setting?

Does it match the input data available locally?

Does it match the user competence assumed locally?

Does it include failure cases?

Does it measure uncertainty?

Does it measure refusal?

Does it measure performance under missing data?

Does it measure performance across patient groups?

Does it measure clinical workflow impact?

Does it measure how humans interact with the output?

Does it measure whether clinicians over-rely on the system?

Does it measure downstream consequence?

If not, the benchmark may still be useful, but it does not fully support deployment authority.

Deployment evidence should include local validation and post-deployment monitoring.

This may include:

- local data evaluation;
- clinician usability review;
- workflow simulation;
- shadow deployment;
- comparison with standard practice;
- error analysis;
- missing-data testing;
- subgroup performance review where appropriate;
- human oversight testing;
- escalation testing;
- incident simulation;
- patient safety review;
- monitoring plan;
- decision-trail design;
- periodic reassessment.

This is especially important for clinical decision support systems.

A system that performs well in a general benchmark may still fail in local practice if the workflow does not match the tested conditions.

For example:

A summary system may omit critical local notes because the local record structure differs from the benchmark data.

A triage system may perform differently because patient flow, staffing, emergency protocols or population characteristics differ.

A diagnostic assistant may be useful for specialists but misleading for general users.

A clinical chatbot may produce safe answers in controlled evaluation but unsafe outputs when patients provide incomplete information.

An imaging AI tool may perform differently across equipment, protocols or image quality.

A medical coding assistant may perform well on standard cases but fail on complex or mixed cases.

This does not mean benchmarks are unimportant.

It means they must be placed in context.

Benchmark evidence is one layer.

Deployment evidence is another.

Runtime evidence is a third.

A mature healthcare deployer should connect all three.

Benchmark evidence helps select and evaluate systems.

Deployment evidence helps assess local suitability.

Runtime evidence helps show that current use remains within valid control conditions.

Article 26 requires attention to the runtime layer.

The deployer must not only know that the system performed somewhere.

It must know whether the system is being used responsibly here, now, under current conditions.

19.5 Healthcare Deployer Readiness Checklist

Healthcare organizations deploying AI systems should translate Article 26 into practical clinical governance checks.

The purpose of this checklist is to support deployer readiness, not to replace clinical, legal, regulatory, medical device or patient safety obligations.

1. System Identification

The organization should identify each AI system used in clinical or healthcare workflows.

For each system, record:

- system name;
- provider;
- intended purpose;
- clinical use case;
- actual use;
- risk classification;
- medical device status where relevant;
- deployment location;
- user group;
- patient group;
- workflow stage;
- system version;
- model version where relevant;
- provider instructions.

2. Clinical Use-Case Mapping

The organization should define the role of the AI system.

Examples include:

- clinical summarization;
- triage support;
- diagnostic support;
- imaging support;
- risk prediction;
- patient prioritization;
- treatment pathway support;
- medication-related support;
- discharge communication support;
- documentation support;
- patient-facing information;
- administrative clinical coding;
- operational scheduling or resource allocation.

For each use case, identify whether the output is advisory, influential, decisive or operationally triggering.

3. Intended Purpose and Actual Use

The organization should compare intended purpose with actual clinical use.

Evidence should show:

- provider-defined intended purpose;
- clinical workflow where used;
- approved users;
- approved patient population;
- approved data sources;
- prohibited uses;
- output role;
- consequence level;
- actual use monitoring;
- reassessment when use changes.

4. Validation Context

The organization should preserve validation context.

Evidence should include:

validation scope;
tested population;
tested setting;
data sources used in validation;
performance limits;
known failure modes;
excluded cases;
user competence assumptions;
human oversight assumptions;
local validation results;
post-deployment monitoring plan.

5. Input Data Governance

The organization should assess whether input data supports the intended clinical use.

Evidence should include:

patient record sources;
data freshness;

- missing-data warnings;
- source reliability;
- current episode relevance;
- medication and allergy status where relevant;
- recent lab or imaging status where relevant;
- external records availability;
- clinician-entered context;
- patient-provided context;
- data validation logs.

6. Human Oversight

The organization should define meaningful clinical oversight.

Evidence should show:

- responsible clinician or role;
- competence and training;
- information shown to reviewer;
- AI output;
- input data condition;
- uncertainty and warnings;
- ability to override;
- escalation route;
- review before consequence;
- final clinical decision;
- modifications made;
- disagreement or override records.

7. Patient Safety and Escalation

The organization should define clinical escalation triggers.

Triggers may include:

- output inconsistent with clinical evidence;
- missing critical data;
- stale patient record;
- use outside intended purpose;
- unexpected recommendation;
- repeated clinician disagreement;
- patient harm or near miss;

- model malfunction;
- unresolved monitoring alert;
- human oversight unavailable;
- system output affecting higher consequence than approved.

Escalation pathways should identify who reviews, who pauses use, who informs provider, who preserves evidence and who manages patient safety response.

8. Logging and Clinical Decision Trail

The organization should retain logs that support reconstructability.

The decision trail should show:

- system used;
- patient case reference where appropriate and lawful;
- input data condition;
- AI output;
- warnings;
- clinician review;
- clinician action;
- escalation;
- final clinical workflow consequence;
- reason continuation was allowed.

9. Monitoring and Post-Deployment Review

The organization should monitor the system after deployment.

Monitoring should include:

- performance indicators;
- error patterns;
- clinician override rates;
- missing-data events;
- subgroup concerns where appropriate;
- patient safety incidents;
- near misses;
- workflow drift;
- use outside intended purpose;
- overreliance indicators;
- escalation trends;
- complaints or patient concerns.

10. Patient and Affected-Person Information

Where applicable, the organization should provide appropriate information to patients or affected persons about AI involvement.

Evidence should include:

- what information was provided;
- when it was provided;
- in what context;
- how AI role was described;
- whether human oversight was explained;
- route for questions or concerns;
- updates when use changes.

11. Third-Party and Medical Device Dependencies

Healthcare deployers should identify relevant third-party dependencies.

This may include:

- AI provider;
- medical device manufacturer;
- cloud provider;
- electronic health record integration;
- data provider;
- imaging system;
- model provider;
- maintenance provider;
- support contact;
- incident reporting route;
- update and change notification process.

12. Evidence Pack

For each healthcare AI system, the organization should maintain an evidence pack showing:

- what the system is used for;
- why it is suitable for that use;
- what validation context supports it;
- who uses it;
- what patients or workflows may be affected;

what data it uses;
 what oversight exists;
 what logs are retained;
 what monitoring occurs;
 what escalation pathways exist;
 how the organization knows use remains within valid clinical and operational boundaries.

This checklist helps translate Article 26 into healthcare deployer governance.

It does not replace clinical responsibility.

It supports it.

The goal is not to make clinicians subordinate to AI governance paperwork.

The goal is to preserve clinical judgment, patient safety and evidence of control in AI-supported healthcare workflows.

Chapter 19 Summary

Healthcare and clinical decision support are among the most sensitive areas for AI deployment.

AI systems may summarize records, support triage, flag risk, suggest diagnoses, recommend pathways, interpret data, draft notes or generate patient-facing communication. Even when they do not make final clinical decisions, they can shape the clinical decision environment.

The central healthcare governance question is:

Can the organization prove that the AI-supported clinical workflow remained under valid clinical and control conditions before the output influenced patient care?

This chapter argues that medical AI governance must distinguish between clinical influence and formal authority.

A clinician may remain formally responsible, but the AI output may still shape what the clinician sees, considers or prioritizes.

Human oversight must therefore be clinically meaningful.

The clinician must understand the system's intended purpose, limitations, input data condition, uncertainty, evidence base and escalation routes. The clinician must also

have the authority, time and support to disagree with the AI output before consequence attaches.

The chapter also distinguishes benchmark evidence from deployment evidence.

A benchmark may show performance under specific test conditions, but deployment evidence must show suitability in the actual clinical environment, with local data, local workflows, local users and real patient consequences.

For healthcare deployers, validation context must travel with the system.

The organization should know what was tested, what was excluded, what population was represented, what failure modes exist and whether local use remains within those assumptions.

Input data freshness is especially important in healthcare.

A clinical AI system may produce a plausible output while missing recent labs, medication changes, imaging results, patient status changes or other critical context.

This creates silent failure.

Finally, the chapter provides a healthcare deployer readiness checklist covering system identification, clinical use-case mapping, intended purpose, validation context, input data, human oversight, patient safety, logging, monitoring, patient information, third-party dependencies and evidence packs.

The practical Article 26 lesson is clear:

Healthcare AI should not be trusted because it produces clinically fluent outputs.

It should be trusted only to the extent that the deployer can show valid clinical evidence, meaningful human oversight, appropriate use, monitoring, escalation and reconstructable decision trails.

In healthcare, evidence of control is not only a compliance concern.

It is part of patient safety.

Chapter 20

Education, Public Sector and Citizen-Facing AI

Legal anchor and framework boundary. [Article 26\(8\) and \(11\)](#); [Article 27](#); [Article 86](#). Public-sector, education and citizen-facing use may trigger additional fundamental-rights, administrative and sector obligations.

20.1 Public-Sector Deployer Obligations

Public-sector AI deployment deserves special attention because public authorities often act in contexts where decisions affect rights, access, eligibility, public trust, public funds, legal standing and institutional treatment.

When AI systems are used by public-sector deployers, the relationship between the organization and the affected person is often asymmetrical.

A citizen, resident, student, applicant, patient, beneficiary or service user may not be able to choose another provider.

They may not understand the internal workflow.

They may not know that AI was used.

They may not be able to challenge the system easily.

They may be affected by a recommendation, score, prioritization or routing decision that happens deep inside an administrative process.

This makes public-sector AI governance particularly sensitive.

The central issue is not only whether the AI system performs a useful function.

The central issue is whether the public authority can prove that the system was used under lawful, appropriate, transparent, reviewable and controlled conditions.

Article 26 deployer obligations are especially important in this context.

A public-sector deployer using a high-risk AI system should be able to show:

- that the system was used according to instructions;
- that the use remained within intended purpose;
- that human oversight was assigned to competent and authorized persons;
- that input data was relevant and sufficiently representative where under the deployer's control;
- that the system was monitored during operation;
- that logs were retained and reconstructable;

- that serious risk or incident pathways existed;
- that affected persons were informed where required;
- that registration duties were considered where applicable;
- that cooperation with authorities could be supported by evidence.

Public-sector AI also raises a distinctive governance issue:

administrative legitimacy.

An AI-supported public-sector process must not only produce efficient outcomes.

It must be explainable enough, contestable enough and accountable enough to support public confidence.

A system may improve speed but weaken trust.

A system may improve consistency but hide context.

A system may improve triage but reduce individual understanding.

A system may reduce workload but make review harder.

A system may appear neutral but reproduce institutional assumptions.

For this reason, public authorities should be careful when describing AI systems as merely administrative.

A routing tool may determine who receives attention first.

A prioritization model may affect access to services.

A fraud risk score may increase scrutiny.

A document classifier may determine how a case is handled.

A summary system may shape what a human reviewer sees.

An eligibility-support tool may influence benefits, permits, grants or other public decisions.

Even when the AI system does not make the final decision, it may shape the administrative pathway.

For deployers, the key question is:

Can the public authority reconstruct how AI influenced the treatment of a person, case or application?

This requires a decision trail.

The decision trail should show:

- what AI system was used;
- what public process it supported;
- what data it used;
- what output it produced;
- what human review occurred;
- what legal or policy framework applied;
- whether the affected person was informed where required;
- whether the output influenced the final outcome;
- whether escalation or review was available;
- why continuation was allowed.

Public-sector deployers should also be careful with automation by stealth.

A system may be introduced as decision support.

Over time, staff may rely on it heavily.

The AI output may become the default pathway.

Manual review may become rare.

The final decision may remain formally human, while practical authority has shifted toward the system.

This pattern is especially dangerous in public administration because affected persons may not see where the shift occurred.

Public-sector AI governance should therefore monitor authority drift.

It should ask:

Has the AI system become more influential than originally intended?

Are human reviewers challenging outputs?

Are decisions increasingly aligned with system recommendations?

Are appeals identifying AI-related issues?

Are certain groups affected differently?

Are staff using AI outputs outside approved scope?

Are logs sufficient to reconstruct case-level influence?

Does the public authority still understand where human judgment enters?

These questions are central to Article 26 in public-sector settings.

The deployer's responsibility is not only to use AI efficiently.

It is to ensure that AI-supported public administration remains legitimate, accountable and capable of review.

20.2 AI Systems Affecting Access and Rights

Citizen-facing AI systems may affect access to rights, services, benefits, education, support, public resources, administrative decisions or institutional treatment.

These systems may operate in areas such as:

- social benefits;
- housing support;
- healthcare access;
- education admissions;
- student assessment;
- vocational training;
- immigration and border processes;
- grant allocation;
- public employment services;
- fraud detection;
- case prioritization;
- complaint handling;
- public-service routing;
- justice administration;
- democratic processes;
- access to essential services.

The governance challenge is that AI influence may occur before the final decision.

A system may not deny a benefit.

But it may prioritize the case lower.

A system may not reject an application.

But it may classify it as lower urgency.

A system may not determine educational access.

But it may rank applicants.

A system may not decide a housing case.

But it may flag risk or missing evidence.

A system may not make a legal determination.

But it may summarize evidence in a way that shapes review.

This means affected-person impact can begin before formal decision-making.

For Article 26 deployers, this matters because governance should focus on influence, not only final authority.

The deployer should ask:

Does the AI system influence access?

Does it influence priority?

Does it influence scrutiny?

Does it influence eligibility assessment?

Does it influence the evidence a human sees?

Does it influence the route through the institution?

Does it influence waiting time?

Does it influence whether a person receives a human review?

Does it influence the quality of explanation?

These questions are important because administrative pathways can create real-world effects even before final decisions.

For example:

A public-service AI system that routes some cases into slower channels may affect access.

A fraud detection model that flags certain applicants may create additional burden.

An education AI system that ranks applicants may affect who receives opportunity.

A vocational training tool that evaluates readiness may shape future livelihood.

A benefit-support tool that summarizes evidence may influence how a case worker understands need.

A chatbot that gives public-service guidance may mislead citizens about available options.

A citizen-facing AI system may also affect vulnerable persons disproportionately.

People with limited language skills, disabilities, low digital literacy, complex circumstances or limited ability to challenge administrative processes may be more affected by opacity or error.

For this reason, public-sector deployers should not treat AI transparency as a purely technical requirement.

It is part of procedural fairness.

Affected persons should know where AI played a meaningful role where information duties apply.

They should have access to human review where required.

They should be able to challenge or correct relevant information.

They should not be left facing an invisible system that shapes their treatment without explanation.

This does not mean every internal use of AI must be disclosed in the same way.

But it does mean that where a high-risk AI system is used to make or assist decisions relating to natural persons, the organization must consider the affected-person information obligation and the broader fairness context.

The deployer should preserve evidence of:

- affected process;
- AI role;
- affected-person group;
- information provided;
- human review route;
- challenge or appeal pathway;
- data correction route;
- case-level decision trail;
- oversight record;
- final decision reasoning;
- escalation records;
- outcome monitoring.

The key question is:

Can the person, organization or authority reconstruct how AI influenced the pathway to the outcome?

If the answer is no, the AI-supported process may be difficult to defend.

In public-sector and citizen-facing contexts, opacity is not merely a technical weakness.

It can become an accountability failure.

20.3 Registration, Transparency and Accountability

Public-sector deployers may face additional responsibilities around registration, transparency and accountability depending on the system, role and legal context.

The purpose of registration and transparency duties is not only administrative.

They make high-risk AI use visible.

Visibility matters because public-sector AI systems can affect large numbers of people.

Without visibility, there may be no meaningful public accountability.

A public authority may deploy an AI-supported system inside a service process.

The affected persons may never know.

Oversight bodies may not see the system.

Civil society may not be aware of the change.

Internal governance may know the system exists, but external accountability may remain weak.

Registration and transparency help address this.

They create a record that certain high-risk AI systems are being used.

But registration should not be treated as the end of accountability.

A registry entry may show that a system exists.

It does not prove that the system is used properly.

It does not prove that human oversight is meaningful.

It does not prove that input data is valid.

It does not prove that logs can reconstruct decision trails.

It does not prove that affected persons understand the system's role.

It does not prove that the system remains within intended purpose.

Therefore, public-sector deployers should connect registration to evidence of control.

A mature public-sector AI governance model should include:

- AI system register;
- public-facing registry obligations where applicable;
- internal ownership;
- intended purpose;
- actual use;
- affected public service;
- legal basis or administrative basis;
- human oversight model;
- data source map;
- monitoring plan;
- log retention plan;
- affected-person information process;
- escalation pathway;
- review and appeal process;
- periodic reassessment.

The public-sector system register should not be a static inventory.

It should be a living governance map.

It should help the organization answer:

Which AI systems affect public-facing processes?

Which are high-risk?

Which affect natural persons?

Which support decisions?

Which require information to affected persons?

Which require human oversight?

Which have unresolved monitoring issues?

Which have changed since deployment?

Which have generated incidents or complaints?

Which systems are due for review?

Transparency also requires clear language.

Public-sector AI explanations should not be written only for technical or legal specialists.

Where information is provided to citizens, students, applicants, beneficiaries or service users, it should be understandable.

People should be able to understand:

- that AI is involved;
- what process it supports;
- what it generally does;
- whether it affects decisions or prioritization;
- whether a human reviews the output;
- how to ask questions;
- how to challenge or correct relevant information;
- who is responsible.

This is especially important in education and public services.

A student, parent, applicant or benefit claimant may not understand terms such as model inference, automated triage, risk scoring or decision support.

The deployer should translate those terms into practical meaning.

For example:

“This system helps prioritize cases for human review.”

“This system helps summarize documents for the case worker.”

“This system supports assessment of eligibility but does not make the final decision.”

“This system flags applications that require additional review.”

“This system assists staff in matching students to available support pathways.”

The information should not overstate or understate the AI system’s role.

If the system is advisory, say so.

If the output is usually followed, the organization should not hide that practical influence.

If the system affects prioritization, say so.

If a human can override the output, explain how.

If a person can challenge the decision, explain where.

Accountability requires this level of clarity.

A vague statement that “AI may be used to improve services” is not enough for meaningful transparency in high-risk contexts.

The deployer must explain the role of AI in the relevant process.

Transparency should also be updated when use changes.

If the system expands from internal summarization to decision support, information should be revisited.

If the system begins affecting new groups, information should be revisited.

If the system’s output becomes more influential, information should be revisited.

Transparency, like other controls, has continued validity.

The question is not only:

Was information provided?

The question is:

Does the information still describe the system’s real role under current conditions?

20.4 Human Oversight in Public Services

Human oversight in public-sector AI is not only a technical control.

It is part of administrative accountability.

Public services often involve decisions that affect people’s rights, access, entitlements, obligations or treatment by the state.

This makes meaningful human oversight essential.

A human reviewer in a public-service workflow must not merely confirm an AI output.

They must be able to understand, question, correct, override or escalate it where necessary.

This requires:

- competence in the public-service domain;
- understanding of the AI system’s role;
- access to relevant case evidence;

- awareness of system limitations;
- authority to disagree;
- time to review;
- route to escalate;
- ability to consider individual circumstances;
- ability to preserve reasoning;
- ability to ensure affected-person rights are respected.

Public-sector human oversight may be weakened by several pressures.

High case volume.

Resource constraints.

Administrative backlog.

Trust in automated systems.

Pressure for consistency.

Lack of technical understanding.

Limited time for individualized review.

Interface designs that nudge approval.

Institutional reluctance to challenge system outputs.

These pressures can turn human oversight into a formality.

A case worker may see an AI-generated summary and rely on it.

A reviewer may accept a risk score without checking source evidence.

A public-service employee may follow automated prioritization because it appears objective.

A supervisor may approve system outputs because disagreement requires more work.

In such settings, the human remains present but control weakens.

For Article 26, this is a serious issue.

Human oversight is meaningful only if it can affect the outcome.

A public-sector deployer should therefore test oversight in real workflows.

It should ask:

Can the human reviewer see the source evidence?

Can the reviewer see what data the AI system used?

Can the reviewer see whether data is missing or stale?

Can the reviewer understand the system's limitations?

Can the reviewer override the output?

Can the reviewer request more information?

Can the reviewer escalate uncertainty?

Can the reviewer consider individual circumstances?

Can the reviewer record disagreement?

Can the workflow continue without review?

Is review performed before consequence attaches?

These questions are especially important in education and public services.

In education, AI-supported decisions may affect admission, assessment, allocation, support, monitoring or future opportunity.

In public services, AI-supported decisions may affect access to benefits, housing, social support, permits, healthcare pathways or administrative treatment.

In justice-related or democratic-process contexts, the consequence may be even more sensitive.

Human oversight should therefore be proportionate to consequence.

A low-impact administrative suggestion may require lighter review.

A decision affecting rights, access or opportunity requires stronger review.

A decision with legal or significant effect may require robust human control and clear reasoning.

The reviewer should not only have the authority to approve.

They should have the authority to pause.

They should be able to say:

The evidence is incomplete.

The AI output is not supported.

The case requires additional context.

The use is outside intended purpose.

The affected person must be informed.

The workflow should escalate.

The system should not be relied upon in this case.

This is human-in-control.

Public-sector deployers should preserve evidence of human oversight, including:

- reviewer role;
- competence and training;
- case evidence shown;
- AI output;
- warnings or limitations;
- review decision;
- modifications;
- override;
- escalation;
- final decision;
- explanation;
- affected-person challenge or appeal outcome.

The central public-service oversight question is:

Could a competent human meaningfully prevent an AI-supported workflow from producing an unfair, unlawful or unsupported outcome?

If not, oversight may be insufficient.

20.5 Public-Sector Control Evidence Map

Public-sector and citizen-facing AI systems require strong evidence mapping.

The goal is to ensure that the organization can reconstruct how AI influenced a public-facing process and whether valid control conditions existed.

A public-sector control evidence map may include the following categories.

1. System Identity

Evidence should show:

- system name;
- provider;
- system version;
- intended purpose;
- high-risk classification;
- public-service domain;
- deployment date;
- internal owner;
- provider instructions;
- approved use cases.

This establishes what system was used and why.

2. Public Process Affected

Evidence should show:

- public service or administrative process;
- case type;
- legal or policy framework;
- affected persons or groups;
- decision pathway;
- consequence level;
- appeal or review route;
- public-facing impact.

This connects the AI system to the public function it affects.

3. AI Role and Influence

Evidence should show whether the AI system:

- classifies;
- scores;
- ranks;
- prioritizes;
- summarizes;
- recommends;
- routes;

- flags;
- drafts;
- supports decision-making;
- triggers further review;
- influences final outcome.

This avoids vague descriptions such as “AI assistance” where the real influence may be more specific.

4. Input Data and Evidence Condition

Evidence should show:

- data sources;
- data freshness;
- completeness;
- missing information;
- source reliability;
- user-entered context;
- case documents;
- relevant legal or policy version;
- data warnings;
- evidence validation.

This helps prove that the AI output was based on valid input conditions.

5. Human Oversight

Evidence should show:

- reviewer role;
- competence;
- training;
- authority;
- evidence shown;
- review timestamp;
- review action;
- override or modification;
- escalation;
- final decision;
- explanation where relevant.

This helps prove that human oversight was meaningful.

6. Affected-Person Information

Evidence should show:

- whether affected-person information was required;
- what information was provided;
- when it was provided;
- how AI involvement was described;
- route for questions;
- route for challenge;
- updates where use changed.

This supports transparency and procedural fairness.

7. Logs and Decision Trail

Evidence should show:

- AI output;
- system state;
- input condition;
- user action;
- review action;
- authority state;
- monitoring alerts;
- escalation;
- downstream consequence;
- final outcome.

This supports reconstructability.

8. Monitoring and Risk Signals

Evidence should show:

- performance monitoring;
- outcome patterns;
- complaints;
- appeals;
- data quality issues;
- human override rates;
- workflow drift;
- use outside intended purpose;

affected-group concerns;
incident or near-miss records.

This supports continued validity.

9. Reassessment Records

Evidence should show reassessment when:

system use changes;
policy changes;
affected groups change;
consequence level increases;
monitoring identifies concerns;
appeals reveal problems;
provider updates instructions;
system is connected to new workflows;
human oversight capacity changes.

This helps show that governance remains current.

10. Accountability Record

Evidence should show:

responsible owner;
governance committee review;
public authority decision owner;
escalation owner;
incident owner;
review cycle;
corrective actions;
external authority cooperation where required.

This supports institutional accountability.

The evidence map should answer the central public-sector question:

How did AI influence this person's pathway through the public system, and why was that influence allowed under current conditions?

If the deployer cannot answer that question, the AI-supported public process may not be sufficiently governable.

This does not mean public-sector AI should not be used.

It means it must be used with visible control.

Public-sector AI should improve public service without weakening public accountability.

That is the Article 26 challenge.

Chapter 20 Summary

Education, public sector and citizen-facing AI are among the most important contexts for Article 26 deployer governance.

AI systems used in education may affect access, admission, assignment, learning outcomes, progression, monitoring or future opportunity.

AI systems used in public services may affect access to benefits, housing, healthcare, public support, immigration processes, justice-related workflows, case prioritization or administrative treatment.

The central issue is not only whether an AI system makes the final decision.

The central issue is whether AI influences the pathway toward a decision or public-service outcome.

This chapter argues that public-sector AI governance must focus on legitimacy, transparency, reviewability and evidence of control.

Public authorities should be able to show that AI systems are used according to instructions, within intended purpose, under meaningful human oversight, with valid input data, active monitoring, reconstructable logs and escalation pathways.

The chapter emphasizes that AI influence may occur before final decision-making.

A system may rank, prioritize, route, summarize, flag or recommend in ways that affect access, scrutiny, waiting time, opportunity or treatment.

This influence should be visible and governable.

The chapter also distinguishes registration and transparency from full accountability.

A registry entry may show that a system exists, but it does not prove that the system is controlled. Public-sector deployers need connected evidence showing system identity, process affected, AI role, input data, human oversight, affected-person information, logs, monitoring, reassessment and accountability.

Human oversight in public services must also be meaningful.

A competent human should be able to understand, challenge, override, escalate or pause AI-supported workflows before rights, access or public-service outcomes are affected.

Finally, the chapter provides a public-sector control evidence map.

The central public-sector governance question is:

How did AI influence this person's pathway through the public system, and why was that influence allowed under current conditions?

If that question cannot be answered, the public authority may have AI deployment, but not evidence of control.

Article 26 therefore points toward a simple principle:

Public-sector AI should not only be efficient.

It should be accountable, reviewable and governable.

Chapter 21

Agentic AI and Article 26

Legal anchor and framework boundary. [Articles 13-14](#); [Article 26](#). Agentic AI treatment is a forward-looking operational application of existing deployer duties, not a separate statutory regime.

21.1 Why Agentic AI Changes Deployer Responsibility

Agentic AI changes the practical meaning of deployer responsibility.

Traditional AI systems often produce outputs.

Agentic AI systems may produce actions, action plans, tool calls, workflow transitions, system updates, messages, recommendations, escalations or multi-step operational chains.

This difference matters.

A classification model may classify.

A scoring system may score.

A summarization tool may summarize.

A recommendation system may recommend.

But an agentic AI system may plan, retrieve, decide which tool to use, call external systems, update records, trigger workflows, ask for approval, summarize evidence, generate communication, route tasks and continue through multiple steps with reduced human involvement.

This changes the control problem.

The deployer is no longer governing only an output.

The deployer may be governing a sequence of transitions.

Each transition may change the operational state of the workflow.

Each tool call may change the system's influence.

Each retrieved context may change the evidence base.

Each generated recommendation may shape human attention.

Each downstream integration may convert a suggestion into action.

Each approval request may shift authority.

Each fallback model may alter the reliability and authority condition.

Each step may create new risk.

This makes Article 26 more complex.

The deployer must still ensure appropriate use, human oversight, input data relevance, monitoring, log retention and escalation.

But in an agentic workflow, these obligations must operate across a chain, not only at a single output point.

For example:

A procurement agent may retrieve supplier data, summarize risk, compare bids, identify a preferred supplier, draft an approval memo, route it to a manager and update a procurement system after approval.

A HR agent may screen candidates, summarize CVs, rank applicants, schedule interviews, draft rejection messages and update applicant tracking records.

A compliance agent may retrieve policies, analyze alerts, summarize case evidence, recommend closure, draft a regulatory note and route exceptions.

A customer support agent may review account history, classify the issue, draft a response, apply a refund policy, update the case and send a message.

A clinical workflow assistant may summarize records, identify relevant lab results, generate risk flags and suggest escalation to a clinician.

In each case, the AI system is not merely answering a question.

It is participating in workflow formation.

That participation may create operational consequence before the organization notices that authority has shifted.

This is why agentic AI changes deployer responsibility.

The deployer must know not only what the AI system can output.

The deployer must know what the AI system can do.

That includes:

- what tools it can access;
- what data it can retrieve;
- what systems it can write to;
- what messages it can send;

- what workflows it can trigger;
- what approvals it can request;
- what recommendations it can escalate;
- what records it can modify;
- what downstream systems may act on its output;
- what human review is required before consequence.

Agentic AI therefore requires deployers to map external actions, connected systems, data flows, affected persons and workflow consequences. This is consistent with emerging regulatory analysis of AI agents under EU law, which highlights that agentic systems can autonomously plan, invoke external tools and execute multi-step chains, creating compliance challenges around traceability, human oversight, cybersecurity and runtime behavioral drift.

The practical Article 26 question becomes:

Can the deployer prove that each relevant agentic transition remained within valid control conditions before consequence attached?

This is a much stronger question than:

Did the agent complete the task?

Completion is not enough.

The agent may complete a task by using stale data, exceeding intended purpose, relying on a fallback model, bypassing meaningful oversight, triggering an unintended downstream workflow or acting under unclear authority.

A mature deployer must therefore govern the chain.

Not only the final output.

21.2 From Output to Workflow Execution

The shift from output to workflow execution is the defining feature of agentic AI governance.

In ordinary AI governance, the organization often asks:

Is the output accurate?

Is the output fair?

Is the output explainable?

Is the output logged?

Was the output reviewed?

These questions remain important.

But agentic AI adds new questions:

What did the system do before producing the output?

What tools did it call?

What evidence did it retrieve?

What context did it assemble?

What system state did it change?

What workflow did it trigger?

What permissions were active?

What model or fallback model was used?

What downstream consequence followed?

Did a human review occur before the consequential step?

Was the agent still operating within intended purpose?

This shift matters because the output may not be the main source of risk.

The risk may occur in the path to the output.

A tool call may retrieve the wrong record.

A context assembly step may include stale policy.

A routing step may send the case to the wrong workflow.

A summary step may omit conflicting evidence.

A model fallback may reduce reliability.

A downstream integration may treat a recommendation as an action.

A system update may occur before human review.

A notification may be sent externally.

This means agentic AI requires transition-level governance.

A transition is a movement from one state to another.

For example:

- from prompt to plan;
- from plan to tool call;
- from tool call to retrieved evidence;
- from evidence to output;
- from output to recommendation;
- from recommendation to approval request;
- from approval to system update;
- from system update to external consequence.

Each transition should be governed according to risk.

Not every transition requires manual review.

But each relevant transition should be visible, logged and subject to control conditions.

For low-risk transitions, automated checks may be enough.

For high-consequence transitions, human review or escalation may be required.

For prohibited or unsupported transitions, refusal should occur.

This creates a practical governance architecture.

The deployer should define:

- allowed transitions;
- restricted transitions;
- prohibited transitions;
- transitions requiring human review;
- transitions requiring escalation;
- transitions requiring evidence freshness checks;
- transitions requiring authority validation;
- transitions requiring additional logging;
- transitions requiring provider or governance review.

For example:

A customer support agent may be allowed to draft a refund explanation, but not issue a refund above a threshold without human approval.

A procurement agent may be allowed to summarize supplier data, but not submit a final supplier recommendation unless sanctions and financial checks are current.

A HR agent may be allowed to summarize applications, but not reject candidates or generate final ranking without human review.

A compliance agent may be allowed to draft case notes, but not close a regulatory case.

A clinical assistant may be allowed to summarize records for clinician review, but not send patient-facing advice without clinical approval.

This is the boundary between assistance and execution.

Agentic AI makes that boundary harder to see because the system may operate across multiple steps.

The deployer must therefore design control gates.

A control gate is a point in the workflow where the system checks whether continuation is allowed.

Control gates may sit before:

- tool execution;
- external system write;
- downstream workflow trigger;
- human-facing recommendation;
- affected-person communication;
- high-consequence decision support;
- model fallback continuation;
- case closure;
- automated approval;
- escalation decision.

At each control gate, the system should evaluate current control conditions.

Is the use within intended purpose?

Is the input evidence fresh?

Is the tool call permitted?

Is the output consequence within authorized scope?

Is human oversight required?

Is the reviewer available?

Are logs complete?

Are there unresolved alerts?

Should the workflow continue, hold, escalate or refuse?

This is how Article 26 becomes agentic workflow governance.

The deployer cannot rely on a final review only.

The agent's intermediate steps may create risk before final review occurs.

The governance system must therefore understand the execution path.

21.3 Tool Use, Permissions and Downstream Effects

Tool use is one of the most important risk surfaces in agentic AI.

A language model without tools may produce text.

A language model with tools can interact with systems.

This difference changes governance.

Tools may allow the agent to:

- retrieve records;
- search databases;
- access documents;
- query customer accounts;
- read HR systems;
- inspect supplier profiles;
- check policies;
- generate tickets;
- update case status;
- send messages;
- create tasks;
- trigger workflows;
- call APIs;
- write to operational systems.

Each tool expands the agent's practical capability.

Each capability may require a permission boundary.

A deployer should therefore treat agent permissions as governance controls.

The organization should know:

What tools can the agent access?

What data can it read?

What systems can it write to?

What actions can it trigger?

What human approval is required?

What logs are generated?

What happens if the tool returns stale or incomplete data?

What happens if the tool is unavailable?

What happens if the agent selects the wrong tool?

What happens if a downstream system treats the agent's output as authoritative?

Permissions should be granular.

Read permission is different from write permission.

Drafting is different from sending.

Recommending is different from approving.

Retrieving evidence is different from modifying records.

Creating a task is different from closing a case.

Internal communication is different from external communication.

A mature deployer should define permission tiers.

For example:

Tier 1: Read-only retrieval

The agent may retrieve information but cannot alter records or trigger workflows.

Tier 2: Drafting and preparation

The agent may generate drafts, summaries or recommendations for human review.

Tier 3: Internal workflow assistance

The agent may create internal tasks or route items within approved boundaries.

Tier 4: Controlled execution

The agent may perform limited actions after rule-based checks or human approval.

Tier 5: High-consequence execution

The agent may affect external persons, regulated processes or operational systems only under strict control gates and human oversight.

Many organizations may describe an agent as read-only because the agent itself does not directly write to a database.

But this can be misleading.

A read-only agent can still create consequence if its output is used by a downstream system or human as an operational trigger.

For example:

A read-only procurement agent recommends a supplier. A procurement system treats the recommendation as the default approval path.

A read-only compliance agent summarizes a case. A human closes the case based on the summary.

A read-only HR agent ranks candidates. Recruiters review only the top-ranked candidates.

A read-only clinical assistant summarizes patient history. A clinician relies on the summary without checking source records.

A read-only public-service agent flags cases. Flagged cases receive additional scrutiny.

In each example, the agent does not write to a system directly.

But it changes the decision environment.

This is why downstream effects matter.

The deployer should not govern only direct permissions.

It should govern practical influence.

A downstream effect map should identify:

- where the agent output goes;
- who sees it;
- which system receives it;
- whether it becomes structured input;
- whether it triggers automation;
- whether it affects prioritization;
- whether it enters official records;

- whether it reaches affected persons;
- whether it influences a regulated process;
- whether it can be challenged or corrected.

This map is essential for Article 26 evidence of control.

If the organization cannot show where agent outputs travel, it cannot fully govern agentic consequence.

Tool use also creates monitoring obligations.

The deployer should monitor:

- tool calls;
- failed tool calls;
- unexpected tools used;
- permission changes;
- new tool integrations;
- high-risk tool calls;
- tool output freshness;
- tool output conflicts;
- repeated tool errors;
- downstream actions triggered by tool outputs;
- human approval before tool execution where required.

Tool use should be logged as part of the decision trail.

The log should show:

- tool called;
- reason for tool call where available;
- data retrieved;
- timestamp;
- source status;
- permission state;
- user or agent identity;
- output used;
- downstream transition;
- human review;
- control gate result.

Without this, the organization may have a final output but not a reconstructable agentic pathway.

That is not enough for mature deployer governance.

21.4 When Recommendations Become Executable Signals

One of the most important risks in agentic AI is the conversion of recommendations into executable signals.

A recommendation may begin as advice.

It may later become a trigger.

This conversion can happen silently.

At first, the agent recommends action to a human.

Then the recommendation is inserted into a workflow tool.

Then the workflow tool creates a task.

Then the task is automatically assigned.

Then the assigned task changes a case status.

Then the case status triggers a notification.

Then the notification affects a customer, worker, supplier, patient, student or citizen.

At each step, the recommendation becomes more operational.

The organization may still describe the agent as advisory.

But the workflow may treat its output as executable.

This is hidden execution.

It is one of the most important future risks in deployer governance.

Recommendations can become executable signals through:

- workflow integration;
- structured output formats;
- API connections;
- automation rules;
- task management systems;
- case-management systems;
- robotic process automation;
- human habit;

dashboard defaults;
approval shortcuts;
performance targets;
downstream system interpretation.

A recommendation becomes dangerous when it crosses from suggestion to consequence without control.

For example:

A supplier recommendation becomes an approval workflow.

A fraud risk recommendation becomes account restriction.

A HR ranking becomes interview selection.

A clinical risk flag becomes triage priority.

A public-service classification becomes eligibility route.

A customer support recommendation becomes automatic response.

A compliance recommendation becomes case closure.

In each case, the agent may not formally “decide.”

But its output becomes a signal that another system or person acts upon.

For Article 26, the deployer must know when this happens.

The organization should ask:

Is the output still advisory?

Is it consumed by another system?

Is it structured in a way that triggers automation?

Is the human expected to review it?

Can the human modify it?

Can the workflow proceed without human approval?

Does the recommendation alter case status?

Does it affect a person?

Does it trigger communication?

Does it create official records?

Does it affect regulated processes?

If the answer is yes, the recommendation may require stronger governance.

The deployer should define recommendation-to-action boundaries.

These boundaries should state when an AI-generated recommendation can be used for:

internal exploration;

draft preparation;

human review;

prioritization;

escalation;

decision support;

workflow trigger;

external action;

final decision.

Each level should have different control requirements.

For example:

Internal exploration may require basic logging.

Draft preparation may require human review before use.

Prioritization may require monitoring for bias or unfair impact.

Decision support may require input data validity and human oversight.

Workflow trigger may require authority validation and control gate approval.

External action may require strict review, logging and escalation pathways.

Final decision may require human-in-control governance and complete decision trail.

This is especially important for agentic systems that generate structured output.

Structured output can be easily consumed by other systems.

A JSON field labeled “recommended_action” may become a trigger.

A risk score may become a threshold.

A status recommendation may become a case update.

A generated email may become customer communication.

A proposed next step may become a task.

The deployer should therefore review not only the agent's natural-language outputs, but the machine-readable outputs that downstream systems consume.

This is where AI governance meets system architecture.

The risk is not only what the agent says.

It is what other systems do with what the agent says.

A mature deployer should therefore implement control gates at recommendation-to-action boundaries.

Before a recommendation becomes executable, the system should check:

- intended purpose;
- input evidence freshness;
- authority;
- consequence level;
- human oversight requirement;
- monitoring alerts;
- policy version;
- output confidence or uncertainty;
- log completeness;
- downstream effect.

If conditions are satisfied, ALLOW.

If evidence is missing, HOLD.

If higher authority is needed, ESCALATE.

If the recommendation exceeds scope, REFUSE.

This is the practical control model for agentic AI.

21.5 Agentic AI Deployer Risk Model

Agentic AI requires a deployer risk model that focuses on actions, transitions and consequences.

The model should not ask only:

What is the AI system?

It should ask:

What can the AI system cause to happen?

A practical agentic AI deployer risk model should include at least ten dimensions.

1. Autonomy Level

How much can the agent do without human intervention?

Can it plan?

Can it choose tools?

Can it execute multiple steps?

Can it continue after intermediate results?

Can it retry?

Can it adapt its path?

Can it act without approval?

The higher the autonomy, the stronger the control requirements.

2. Tool Access

What tools can the agent use?

Can it read?

Can it write?

Can it send?

Can it update?

Can it trigger?

Can it delete?

Can it call external APIs?

Tool access defines practical capability.

3. Data Sensitivity

What data can the agent access?

Personal data?

Health data?

Employment data?

Financial data?

Customer data?

Public-sector case data?

Trade secrets?

Security information?

Sensitive data increases governance requirements.

4. Workflow Consequence

What can the agent's output affect?

Internal draft?

Human review?

Customer communication?

Employment decision?

Financial process?

Clinical pathway?

Public-service outcome?

System configuration?

Regulated reporting?

Control should follow consequence.

5. Human Oversight

Where is human oversight required?

Before output?

Before recommendation?

Before tool execution?

Before external action?

Before case closure?

Before affected-person impact?

Oversight must occur before consequence, not only after.

6. Authority State

Who or what has authority to move the workflow forward?

Is authority role-based?

Policy-based?

Threshold-based?

Delegated?

Automated?

Conditional?

Authority should be current and inspectable.

7. Evidence Freshness

What evidence does the agent use?

Is it current?

Is it source-verified?

Is it complete?

Is it representative?

Is it retrieved at the right moment?

Freshness should be checked at critical transition points.

8. Boundary Conditions

What is the agent not allowed to do?

What use cases are prohibited?

What tools are restricted?

What outputs cannot become action?

What conditions require refusal?

Boundaries should be explicit.

9. Logging and Reconstructability

Can the organization reconstruct the agent's path?

What prompt was used?

What context was retrieved?

What tools were called?

What outputs were generated?

What human reviews occurred?

What action followed?

Logs must support decision trails, not only technical debugging.

10. Escalation and Refusal

When does the agent stop?

When does it hold?

When does it escalate?

When does it refuse?

What happens when evidence is stale, authority is unclear, human oversight is missing or use is outside intended purpose?

An agent without refusal logic may be efficient, but not sufficiently governed.

This risk model should be applied before deployment and during operation.

Agentic AI risk can change over time.

New tools may be added.

Permissions may expand.

Users may create new workflows.

The agent may be connected to downstream systems.

Human review may be reduced.

Structured outputs may become triggers.

A fallback model may be introduced.

Business teams may rely more heavily on the agent.

This means agentic AI requires reassessment.

A deployer should reassess when:

a new tool is added;

write permissions are enabled;

downstream integrations change;
output becomes more influential;
the agent begins handling new data;
the agent is used by new teams;
human oversight changes;
model routing changes;
context management changes;
incidents or near misses occur;
logs are incomplete;
affected persons change;
consequence level increases.

The risk model should produce an operational decision:

ALLOW.

HOLD.

ESCALATE.

REFUSE.

For example:

ALLOW: agent drafts internal summary from current evidence for trained reviewer.

HOLD: agent cannot verify evidence freshness before recommendation.

ESCALATE: agent is asked to use output in a higher-consequence workflow than approved.

REFUSE: agent is asked to send external decision communication without required human approval.

This is the practical meaning of agentic deployer governance.

The deployer must govern not only the model.

Not only the output.

Not only the prompt.

The deployer must govern the chain from instruction to consequence.

Chapter 21 Summary

Agentic AI changes deployer responsibility because it moves AI systems from output generation toward workflow participation and execution.

An agent may plan, retrieve evidence, call tools, generate recommendations, route tasks, update records, trigger workflows and interact with downstream systems. This means the deployer must govern not only what the system says, but what it can cause to happen.

This chapter explains why Article 26 becomes more complex in agentic workflows.

The obligations around use according to instructions, human oversight, input data, monitoring, logs and escalation must operate across a chain of transitions, not only at the final output point.

The chapter introduces transition-level governance.

Agentic workflows should identify critical transition points such as tool calls, evidence retrieval, model selection, recommendation generation, approval requests, system writes, external communication and downstream action.

At each critical transition, the deployer should ask whether continuation is allowed under current control conditions.

Tool use is a major risk surface.

A model with tools can read, write, trigger, update, send and act. Tool permissions should therefore be granular, monitored and logged.

The chapter also explains hidden execution: recommendations can become executable signals when downstream systems or humans treat them as triggers. A system may remain formally advisory while its output becomes operationally constructive in practice.

Finally, the chapter proposes an agentic AI deployer risk model based on autonomy, tool access, data sensitivity, workflow consequence, human oversight, authority, evidence freshness, boundaries, logging and escalation.

The central agentic AI governance question is:

Can the deployer prove that each relevant agentic transition remained within valid control conditions before consequence attached?

If yes, the workflow may continue.

If no, it should refuse.

If uncertain, it should hold or escalate.

That is how Article 26 becomes agentic runtime governance.

Chapter 22

Model Substitution, Fallback and Authority Continuity

Legal anchor and framework boundary. [Articles 13-14](#); [Article 26](#). Fallback and authority continuity are framework concepts for preserving valid control when models or components change.

22.1 Model Dependency as a Governance Issue

Model dependency is often treated as an infrastructure issue.

A production system depends on a model.

The model becomes unavailable.

The system needs a fallback.

The engineering question becomes:

Can another model answer?

That question is important, but incomplete.

For high-risk AI systems and high-consequence workflows, model dependency is not only an availability problem.

It is an authority continuity problem.

The stronger question is:

Does the replacement model support the same level of authority, confidence, evidence quality, policy scope and operational consequence as the original model?

This distinction matters because not all models are equivalent.

Two models may both generate fluent answers.

Two models may both pass general benchmarks.

Two models may both support the same API shape.

Two models may both be capable of producing recommendations, summaries, classifications or decisions.

But they may not support the same governance conditions.

They may differ in:

- validation evidence;
- intended use;
- benchmark coverage;
- clinical, financial, HR or public-sector suitability;
- explainability;
- refusal behaviour;
- hallucination profile;
- calibration;
- tool-use behaviour;
- latency;
- context handling;
- data retention terms;
- regional availability;
- provider controls;
- logging support;
- security posture;
- regulatory evidence;
- risk classification;
- human oversight assumptions.

A fallback model may therefore restore technical continuity while weakening governance continuity.

The system may keep running.

The workflow may complete.

The output may look similar.

But the basis for trust may have changed.

This is why deployers should not treat model substitution as a neutral technical recovery event.

A model substitution may change the control condition.

If the original model was approved for a specific workflow, with specific validation evidence, under specific provider instructions and specific monitoring assumptions, then replacing it with another model may require reassessment.

The replacement model may be suitable.

It may even be better.

But suitability should not be inherited silently.

For high-risk AI systems, inherited authority is dangerous.

A fallback model should not automatically receive the same authority as the primary model unless the deployer can show that the fallback model is approved for the same use, consequence level and control environment.

This becomes especially important in agentic AI.

An agent may depend on a specific model for planning, reasoning, tool selection, summarization, classification or recommendation.

If the model changes, the agent's behaviour may change.

The agent may call different tools.

It may interpret instructions differently.

It may produce different confidence.

It may refuse less often.

It may summarize differently.

It may handle uncertainty differently.

It may be more or less reliable for the workflow.

The deployer must therefore know when model dependency changes.

A mature deployer should maintain a model dependency register.

This register should identify:

- primary model;
- fallback model;
- model provider;
- approved use cases;
- approved consequence level;
- validation evidence;
- authority level;
- allowed workflows;
- prohibited workflows;
- human oversight requirements;
- logging capability;
- monitoring signals;
- fallback conditions;
- authority reduction rules;

- escalation rules.

This transforms model dependency from hidden infrastructure into visible governance.

The key question is not only:

Which model is currently available?

The key question is:

Which model is currently carrying authority inside this workflow?

That is the deployer governance issue.

22.2 Fallback Is Not Neutral

Fallback is often described as resilience.

If the primary model is unavailable, route to a secondary model.

If the expensive model is unnecessary, route to a cheaper model.

If latency is high, use a faster model.

If one provider fails, use another.

If one region is unavailable, switch region.

These patterns can be useful.

They improve availability.

They reduce cost.

They support operational continuity.

But in high-risk AI governance, fallback is not neutral.

Fallback changes the conditions under which the workflow operates.

A fallback may change:

- model capability;
- model validation status;
- output quality;
- confidence level;
- explainability;
- refusal behaviour;

- context handling;
- safety settings;
- tool-use behaviour;
- provider terms;
- logging support;
- data processing location;
- security posture;
- operational latency;
- monitoring visibility;
- evidence quality;
- human oversight need;
- authority level.

This means fallback should be treated as a governed transition.

A fallback event should not be invisible.

It should not be hidden inside infrastructure.

It should not preserve full authority automatically.

It should not allow the workflow to continue as if nothing changed.

For example:

A medical AI assistant uses a validated primary model for clinical summarization. The primary model becomes unavailable. The system falls back to a general-purpose model that has not been validated for the same clinical workflow. The output may still be fluent, but clinical authority should not remain the same.

A financial compliance agent uses an approved model for policy-based case summarization. The system falls back to a cheaper model that handles long context less reliably. The case summary may still look complete, but the evidence basis has changed.

A HR screening tool uses a model that was tested for a specific recruitment workflow. The fallback model was approved only for internal drafting. Candidate ranking should not continue with the same authority.

A procurement agent uses a primary model approved for supplier risk summaries. The fallback model cannot reliably preserve citations or source references. Supplier recommendations should not proceed unchanged.

A public-service assistant uses a model with strict refusal behaviour for eligibility support. The fallback model produces answers more readily. The workflow may become more risky.

In each case, fallback may restore function.

But it may reduce governance confidence.

The deployer should therefore define fallback classes.

A fallback model may be:

Equivalent fallback

The replacement model has been approved for the same workflow, same consequence level, same evidence requirements and same authority level.

Reduced-authority fallback

The replacement model may support the task, but only with reduced authority, additional human review or stronger evidence checks.

Draft-only fallback

The replacement model may generate internal drafts or summaries, but outputs cannot influence decisions without full human review.

Escalation-only fallback

The replacement model may identify that a workflow needs human review, but cannot produce substantive recommendations.

No-use fallback

The replacement model is not approved for the workflow, and the system must refuse continuation.

This classification prevents the organization from treating all fallback as equivalent.

A mature deployer should define fallback rules before failure occurs.

The system should know:

When may fallback occur?

Which fallback model may be used?

For which workflow?

At what consequence level?

With what authority reduction?

With what human review?

With what logging?

With what notification?

With what escalation?

With what refusal conditions?

Fallback without these rules can create silent governance failure.

The system continues smoothly, but the basis for control has changed.

This is why fallback must be governed.

Resilience without governance can preserve operation while degrading trust.

22.3 Authority Reduction After Substitution

Authority reduction is a practical way to govern model substitution.

The idea is simple:

When a model changes, the workflow should not automatically preserve the same authority.

The replacement model may continue the task, but with reduced ability to influence consequence until the control environment is restored or revalidated.

Authority reduction recognizes that model substitution changes the control condition.

The workflow may still proceed in some form.

But the output may not carry the same decision weight.

For example:

A primary model output may be allowed to support a recommendation.

A fallback model output may be allowed only to produce a draft.

A primary model output may be allowed to support automated prioritization.

A fallback model output may require human review before prioritization.

A primary model output may be allowed to feed a workflow gate.

A fallback model output may require escalation.

A primary model output may support high-consequence decision support.

A fallback model output may be limited to low-consequence internal assistance.

This is authority reduction.

It is not only a technical setting.

It is a governance control.

Authority reduction may be expressed through levels.

Authority Level 1: Internal drafting only

The model may assist with text, summary or preparation, but cannot influence a decision or workflow transition without human review.

Authority Level 2: Advisory support

The model may produce recommendations for human consideration, but the human must review source evidence and make an independent decision.

Authority Level 3: Controlled decision support

The model may support structured decisions within approved workflows, with required oversight, logging and monitoring.

Authority Level 4: Operational trigger support

The model output may trigger workflow actions only when strict control gates are satisfied.

Authority Level 5: No autonomous authority

The model cannot be used for the workflow under current conditions.

A fallback event may reduce the workflow from Level 3 to Level 2, or from Level 2 to Level 1.

In high-consequence cases, it may reduce authority to Level 5.

The reduction should depend on:

- validation gap;
- workflow consequence;
- model difference;
- evidence quality;

- human oversight availability;
- output type;
- downstream effect;
- affected persons;
- regulatory sensitivity;
- ability to reconstruct the output;
- provider instructions;
- internal approval status.

For example:

If the fallback model is approved for the same use and consequence level, no authority reduction may be necessary.

If the fallback model is approved only for drafting, authority should be reduced.

If the fallback model lacks required validation evidence, authority should be reduced or the workflow should hold.

If the fallback model cannot preserve citations or evidence links, authority should be reduced.

If the workflow is high-consequence and human oversight is unavailable, the fallback should refuse continuation.

This prevents technical recovery from becoming governance bypass.

Authority reduction should be visible to users.

A human reviewer should know:

The primary model was unavailable.

A fallback model was used.

The fallback model has reduced authority.

Additional evidence review is required.

The output cannot be used for final decision without approval.

This prevents false confidence.

The output may look similar, but the governance status is different.

Authority reduction should also be logged.

The log should show:

- primary model requested;
- reason fallback occurred;
- fallback model used;
- authority before fallback;
- authority after fallback;
- workflow affected;
- consequence level;
- human review requirement;
- evidence refresh requirement;
- control gate decision;
- final outcome.

This is evidence that fallback was governed.

Without this evidence, the organization may not be able to prove that model substitution was controlled.

22.4 New Admissibility States

Model substitution should trigger a new admissibility state.

This is one of the most important design principles for high-risk AI governance.

Admissibility means whether an output, recommendation, decision or workflow transition is allowed to continue under current control conditions.

When the model changes, the control conditions change.

Therefore, admissibility should be reassessed.

The workflow should not assume that the previous admissibility state still applies.

Before substitution, the workflow may have been admissible.

After substitution, it may no longer be admissible.

Or it may be admissible only with reduced authority, fresh evidence, human review or escalation.

This creates the need for model-substitution states.

A practical model-substitution admissibility model may include:

MODEL_AVAILABLE

Primary model available and approved for current workflow.

MODEL_UNAVAILABLE

Primary model unavailable; workflow cannot continue without fallback or hold.

FALLBACK_AVAILABLE

Approved fallback model available.

FALLBACK_EQUIVALENT

Fallback model approved for same workflow and authority level.

FALLBACK_REDUCED_AUTHORITY

Fallback model available, but output authority must be reduced.

FALLBACK_REQUIRES_HUMAN_REVIEW

Fallback output may be used only after human review.

FALLBACK_REQUIRES_FRESH_EVIDENCE

Fallback output requires evidence refresh before use.

FALLBACK_REQUIRES_ESCALATION

Fallback use requires governance, compliance, clinical, financial, HR or domain-specific escalation.

FALLBACK_NOT_APPROVED

Fallback model not approved for this workflow.

REFUSE_CONTINUATION

Workflow must not proceed under current model conditions.

These states help the system avoid silent continuation.

They also make governance inspectable.

For example:

A procurement agent requests the approved primary model.

The primary model is unavailable.

The system checks the fallback registry.

The fallback model is approved for supplier summaries but not supplier recommendations.

The workflow enters FALLBACK_REDUCED_AUTHORITY.

The system allows draft summary generation but blocks final recommendation until human review and supplier evidence refresh.

This is a governed transition.

Another example:

A clinical summarization tool requests the primary model.

The primary model is unavailable.

The fallback model has not been validated for clinical use.

The workflow enters FALLBACK_NOT_APPROVED.

The system refuses clinical output generation and escalates to manual process.

This is not failure.

It is safe refusal.

Another example:

A customer support agent falls back from a frontier model to a smaller approved model.

The workflow is low consequence and internal.

The fallback model is approved for drafting.

The workflow continues, but the system logs fallback and marks output as draft-only.

This is proportional governance.

The point is not to stop every workflow whenever fallback occurs.

The point is to stop automatic authority inheritance.

Model substitution should be treated as a change in runtime evidence.

The system should ask:

Is this model approved for this task?

Is it approved for this consequence level?

Does it support the same evidence quality?

Does it support the same human oversight assumptions?

Does it preserve the same logging and traceability?

Does it change data-processing or provider conditions?

Does it change refusal behaviour?

Does it change tool-use behaviour?

Does it change authority?

If yes, the workflow may continue under defined conditions.

If no, it should hold, escalate or refuse.

This is how model substitution becomes admissibility governance.

22.5 Evidence That Fallback Was Governed

Fallback governance must produce evidence.

Without evidence, the organization may not be able to prove that model substitution was controlled.

A deployer should preserve a fallback evidence record.

This record should answer:

What happened?

Why did fallback occur?

Which model was unavailable?

Which model replaced it?

Was the replacement approved?

What authority changed?

What human review was required?

What evidence was refreshed?

What workflow was affected?

What consequence followed?

Why was continuation allowed?

A fallback evidence record may include:

- timestamp;

- system or agent identity;
- workflow identity;
- primary model requested;
- primary model status;
- fallback trigger;
- fallback model used;
- fallback approval status;
- fallback classification;
- authority level before substitution;
- authority level after substitution;
- task type;
- consequence level;
- input data condition;
- evidence freshness state;
- human oversight requirement;
- reviewer identity or role;
- control gate decision;
- ALLOW, HOLD, ESCALATE or REFUSE state;
- downstream action;
- final outcome;
- escalation record where relevant.

This evidence supports reconstructability.

If an auditor, regulator, board member, affected person or internal reviewer asks why a fallback model was used, the organization should be able to explain.

It should not discover after the fact that model substitution occurred invisibly.

The deployer should also monitor fallback patterns.

Frequent fallback may indicate operational risk.

For example:

The primary model may be unreliable.

The fallback model may be used more often than intended.

Users may route to cheaper models to reduce cost.

System routing may send high-consequence workflows to lower-authority models.

Fallback may occur during peak periods.

Fallback may correlate with increased human overrides.

Fallback may correlate with lower output quality.

Fallback may bypass monitoring.

Fallback may increase unresolved escalations.

These patterns matter.

A fallback that was designed as an exception may become normal operation.

If that happens, the control model should be reassessed.

The deployer should ask:

Is the fallback still exceptional?

Has the fallback become routine?

Is the fallback model now effectively part of the production control environment?

Has authority reduction been preserved?

Are users aware of fallback status?

Are logs complete?

Are incidents increasing?

Does the fallback require formal approval as a primary or secondary model?

This is continued validity.

Fallback governance should not only operate during the event.

It should also support ongoing review.

A mature deployer should produce fallback reports, including:

- number of fallback events;
- affected workflows;
- models involved;
- reason for fallback;
- authority reductions applied;
- human reviews triggered;
- escalations;
- refusals;
- incidents;
- unresolved issues;

- remediation actions.

This helps governance teams understand model dependency as part of runtime risk.

Fallback evidence also supports human oversight.

A human reviewer should know when fallback occurred.

They should know whether the output came from the approved primary model or a reduced-authority replacement.

They should know whether additional evidence review is required.

They should know whether the output can influence consequence.

This prevents human oversight from being misled by invisible substitution.

The strongest fallback governance model therefore includes:

- approved model registry;
- fallback registry;
- authority rules;
- workflow consequence mapping;
- runtime fallback detection;
- user-visible fallback status;
- automatic authority reduction;
- evidence refresh where required;
- human review where required;
- escalation where required;
- refusal where required;
- fallback event logging;
- periodic fallback review.

This turns model substitution into evidence of control.

The organization can show that fallback did not silently preserve authority.

It can show that fallback was a governed transition.

Chapter 22 Summary

Model substitution is not only an infrastructure issue.

It is an authority continuity issue.

A fallback model may restore availability, but it may not preserve the same validation evidence, confidence, refusal behaviour, context handling, logging capability, provider conditions, human oversight assumptions or operational authority.

This chapter argues that fallback is not neutral.

Changing the model changes the control condition.

For high-risk AI systems and high-consequence workflows, fallback should therefore be treated as a governed transition.

The deployer should maintain a model dependency register and fallback registry, identifying primary models, approved fallback models, approved workflows, consequence levels, authority levels, human oversight requirements, evidence requirements and refusal conditions.

The chapter introduces authority reduction after substitution.

A fallback model may be allowed to continue the workflow, but with reduced authority. It may be limited to drafting, advisory support, human-reviewed output, escalation-only use or no use at all.

The chapter also proposes new admissibility states for model substitution, including `MODEL_UNAVAILABLE`, `FALLBACK_AVAILABLE`, `FALLBACK_EQUIVALENT`, `FALLBACK_REDUCED_AUTHORITY`, `FALLBACK_REQUIRES_HUMAN_REVIEW`, `FALLBACK_NOT_APPROVED` and `REFUSE_CONTINUATION`.

These states prevent silent continuation.

Finally, the chapter defines evidence that fallback was governed.

A fallback evidence record should show which model was unavailable, which model replaced it, why substitution occurred, what authority changed, what human review was required, what evidence was refreshed, what control gate decision was made and why continuation was allowed.

The central governance principle is simple:

Fallback should not preserve authority silently.

A system may continue technically.

But if the model changes, the deployer must reassess whether the workflow is still admissible under current control conditions.

This is how model availability becomes runtime governance.

Chapter 23

Context Management as Control Management

Legal anchor and framework boundary. [Article 13](#); [Article 26\(4\)-\(6\)](#). Context management as control management is an operational interpretation for data, monitoring and evidence continuity.

23.1 Context as Operating Boundary

Context management is often discussed as an efficiency problem.

Too much context increases cost.

Too much history increases latency.

Too many irrelevant documents increase noise.

Too much duplicate information wastes tokens.

Poor context retrieval reduces performance.

These issues matter.

But in high-risk AI governance, context management is not only a cost problem.

It is a control problem.

The context an AI system receives shapes what it can reason about, what it ignores, what it repeats, what it treats as relevant, what it treats as current and what it treats as authoritative.

This means context is part of the AI system's operating boundary.

For a human decision-maker, context determines judgment.

For an AI-supported workflow, context determines the evidence environment.

If the system receives the wrong context, stale context, incomplete context, irrelevant context or unauthorized context, the output may still appear fluent and structured, but the governance condition behind the output may be weak.

This is especially important for large language models and agentic AI systems.

The model may not know the organization's current policy unless that policy is provided.

The agent may not know that a supplier record is stale unless freshness state is included.

The system may not know that human review is required unless the workflow context says so.

The model may not know that the user lacks authority unless role context is supplied.

The agent may not know that a recommendation cannot trigger action unless the boundary condition is present.

The output may therefore reflect not only the model's capability, but the quality of the context environment.

This is why context should be treated as a control surface.

A deployer should ask:

What context is supplied?

Where does it come from?

Who selected it?

Is it current?

Is it complete?

Is it relevant?

Is it authorized?

Is it within intended purpose?

Does it include applicable policy?

Does it include evidence limitations?

Does it include human oversight requirements?

Does it include downstream consequence?

Does it include refusal conditions?

Does it include what the system should not do?

These questions matter because context shapes authority.

For example:

A procurement agent given supplier performance data but not current sanctions information may recommend a supplier without seeing a critical risk.

A HR assistant given candidate CVs but not role-specific criteria may rank candidates using weak assumptions.

A clinical summarization tool given partial patient records may omit recent changes.

A compliance assistant given an outdated policy may produce a plausible but obsolete interpretation.

A customer support agent given old account status may recommend a response that no longer fits the customer's current risk profile.

A public-service assistant given an incomplete case record may summarize eligibility incorrectly.

In each case, the problem may not be that the model is incapable.

The problem may be that the context does not support the consequence attached to the output.

The deployer must therefore govern context as part of Article 26 operational control.

This includes:

- context source selection;
- context retrieval rules;
- freshness checks;
- relevance filters;
- authorization controls;
- policy versioning;
- evidence completeness indicators;
- context compaction rules;
- context logging;
- context review;
- context-based pause conditions.

The central governance question is:

Did the system receive the right context for the authority attached to the output?

If yes, the workflow may continue.

If no, the workflow should hold, escalate or refuse.

If unknown, the workflow should not assume validity.

Context is not just what the model sees.

Context is what the organization allows the model to rely on.

23.2 Stale Facts and Degraded Decision Environments

Stale context is one of the most dangerous forms of AI governance risk.

A stale fact may remain visible long after the conditions that made it true have changed.

This is dangerous because AI systems often treat supplied context as relevant unless instructed otherwise.

If a stale fact enters the context window, the model may use it.

If a cached summary contains outdated assumptions, the model may repeat them.

If a policy excerpt is obsolete, the system may reason from it.

If an approval state is old, the agent may treat authority as current.

If a supplier profile is outdated, the recommendation may inherit false confidence.

If a human oversight assignment has changed, the workflow may believe review is available when it is not.

This is how stale facts degrade decision environments.

The AI system may produce a strong answer from weak context.

The output may not reveal the weakness.

A human reviewer may assume that the system saw current information.

A downstream system may act on the output.

The log may record normal execution.

But the decision environment has already degraded.

This is a specific form of silent failure.

The system does not fail visibly.

It continues with stale context.

Examples include:

A compliance assistant uses a policy summary created before the latest internal policy update.

A procurement agent uses supplier risk context retrieved before a financial downgrade.

A HR system uses role requirements that changed after the job description was updated.

A clinical assistant summarizes patient history from a snapshot that excludes recent lab results.

A financial services agent uses a previous risk appetite statement.

A public-service assistant uses eligibility rules that have been superseded.

A customer support agent uses account context retrieved before a new fraud flag was added.

In all of these cases, the system may complete the task.

But completion is not proof of decision validity.

The deployer should therefore treat context freshness as a runtime control condition.

Context should include freshness metadata.

For example:

- retrieved at time;
- source last updated;
- policy version;
- approval status timestamp;
- data source health;
- case record completeness;
- known missing data;
- source reliability;
- expiry time;
- refresh requirement.

Without freshness metadata, the model may not distinguish current evidence from old evidence.

The human reviewer may not distinguish either.

This is why context should not be passed as raw content only.

Context should be passed with condition.

A mature context object should answer:

What is this information?

Where did it come from?

When was it retrieved?

When was it last updated?

Is it authoritative?

Is it complete?

Is it approved for this use?

Is it still valid?

What limitations apply?

When must it be refreshed?

This turns context into governance evidence.

The system should not only consume context.

It should evaluate context condition.

If the context is stale, the workflow may continue only under reduced authority, human review or refreshed evidence.

If the context is critically stale, the workflow should hold.

If the context is outside intended purpose, the workflow should refuse or escalate.

This is how context management becomes control management.

The goal is not to eliminate all stale information.

Some older information may remain useful as background.

The goal is to prevent stale information from carrying current authority.

A historical fact may be acceptable for background explanation.

It may not be acceptable for high-consequence decision support.

The deployer should therefore distinguish between:

- background context;
- supporting evidence;
- authoritative evidence;
- decision-critical evidence;
- expired evidence;
- prohibited context.

This classification helps the system know how much weight context should carry.

Not every piece of context deserves the same authority.

23.3 Context Hygiene and Governance Risk

Context hygiene usually refers to keeping the context window clean.

Remove duplicates.

Remove irrelevant history.

Remove outdated content.

Avoid unnecessary token usage.

Keep the model focused.

This is useful engineering practice.

But in deployer governance, context hygiene also reduces risk.

Poor context hygiene can create governance failure.

Duplicate information may cause the model to overweight certain assumptions.

Irrelevant history may distract from current facts.

Old summaries may preserve outdated decisions.

Conflicting policy versions may confuse the output.

Unverified documents may appear authoritative.

User-provided context may override official records.

Previous conversation history may introduce assumptions that do not belong in the current workflow.

Compressed summaries may omit limitations.

Retrieval systems may select semantically similar but legally or operationally irrelevant documents.

In low-risk use cases, these issues may cause inconvenience.

In high-risk workflows, they may affect consequence.

For example:

A legal or compliance assistant may retrieve an old policy because it is semantically similar to the current one.

A HR assistant may include previous candidate notes that should not influence a new decision.

A procurement system may include supplier marketing claims without distinguishing them from verified due diligence.

A healthcare assistant may include a previous episode summary without marking it as historical.

A public-sector AI system may include irrelevant prior case notes that bias the current assessment.

A financial services assistant may retrieve a draft procedure rather than the approved procedure.

These are context hygiene failures.

They are also governance failures because they allow the AI system to reason from an ungoverned evidence environment.

A deployer should therefore define context hygiene rules for high-risk AI workflows.

These rules may include:

- use only approved sources for decision-critical context;
- include policy version metadata;
- exclude superseded policies unless marked historical;
- mark draft documents as draft;
- separate user-provided claims from verified records;
- remove duplicate records;
- remove irrelevant prior case history;
- include limitations and missing evidence;
- preserve source references;
- flag conflicting context;
- require refresh before high-consequence use;
- block prohibited context categories;
- log retrieved context identifiers.

Context hygiene should also account for retrieval quality.

Retrieval-augmented generation can improve provenance, but provenance is not the same as correctness.

A citation shows where information came from.

It does not prove that the information is current, relevant, complete or correctly applied.

Therefore, deployers should not treat retrieved context as automatically valid.

The governance question is:

Was the retrieved context fit for the decision purpose?

This requires controls around:

- source authority;
- relevance ranking;
- freshness;
- completeness;
- conflict detection;
- policy hierarchy;
- intended purpose;
- data access permissions;
- human review.

For high-risk AI systems, context hygiene should be monitored.

Signals of poor context hygiene may include:

- repeated retrieval of outdated documents;
- conflicting policy versions in context;
- missing source references;
- excessive reliance on old summaries;
- user overrides based on unverified information;
- output citing irrelevant evidence;
- human reviewers frequently correcting context;
- downstream errors caused by context gaps;
- model hallucination when context is insufficient;
- escalation due to evidence conflict.

These signals should not be treated only as quality issues.

They should be treated as control signals.

If context hygiene is poor, the workflow's outputs may not be admissible for high-consequence use.

The correct response may be HOLD until context is refreshed or corrected.

A mature deployer should therefore include context hygiene in Article 26 monitoring.

It is part of ensuring appropriate use, valid input conditions, meaningful human oversight and reconstructable evidence.

23.4 Context Compaction and Loss of Evidence

Context compaction is often necessary.

Long workflows create long histories.

Agents generate intermediate steps.

Users provide multiple instructions.

Documents are retrieved.

Tools return results.

Human reviewers add comments.

The system may not be able to carry everything forward.

To manage cost and performance, systems often compress context into summaries.

This can be useful.

But context compaction creates governance risk.

A summary is not the same as the source evidence.

A compressed context may preserve conclusions while losing uncertainty.

It may preserve the main point while losing exceptions.

It may preserve a recommendation while losing the limitation.

It may preserve an approval while losing the condition attached to it.

It may preserve a policy statement while losing the version.

It may preserve the answer while losing the evidence trail.

This is dangerous in high-risk AI workflows.

The system may continue with a compacted summary that no longer contains enough evidence to support consequence.

A human reviewer may see the summary and assume it represents the full record.

A downstream agent may treat the summary as authoritative.

The original evidence may remain stored somewhere, but the active context no longer carries the conditions needed for safe use.

This is how compaction can cause loss of control.

Examples include:

A procurement agent compacts supplier due diligence into “supplier acceptable” but drops the condition that cybersecurity review is pending.

A HR assistant summarizes a candidate as “strong fit” but drops the uncertainty about missing qualification evidence.

A clinical assistant summarizes a patient as “stable” but omits that the latest lab result was not included.

A compliance agent summarizes a policy as allowing an action but omits the threshold condition.

A public-service assistant summarizes a case as complete but omits a pending document submission.

A financial agent summarizes risk as low but omits that the market data was stale.

In each case, the compacted context may become more dangerous than raw missing data because it creates confidence without preserving limitations.

For deployers, context compaction should therefore be governed.

The organization should define what must never be lost in compaction.

Critical context elements may include:

- source identifiers;
- timestamps;
- policy versions;
- input data freshness;
- missing evidence;
- uncertainty;
- warnings;
- limitations;
- human review status;
- authority state;
- escalation state;
- pending checks;
- prohibited uses;

- downstream consequence;
- refusal conditions.

A compressed summary should not be allowed to remove these elements when the workflow is high-consequence.

This may require structured summaries.

Instead of a free-form summary, the system may maintain a governance context block.

For example:

Summary: supplier appears suitable for initial review.

Evidence state: financial data current; sanctions check current; cybersecurity review pending.

Authority state: recommendation may not proceed to final approval.

Required action: HOLD until cybersecurity review completed.

This is much stronger than:

Supplier appears suitable.

The governance context block preserves the control condition.

A mature context compaction strategy should separate:

- narrative summary;
- evidence references;
- freshness state;
- uncertainty state;
- authority state;
- pending actions;
- escalation requirements;
- refusal conditions.

This allows the model to operate efficiently without losing governance-critical information.

Compaction should also be logged.

The log should show:

what source context was compacted;
when compaction occurred;

what summary was produced;
what evidence references were retained;
what limitations were retained;
what warnings were retained;
whether any control-critical information was removed;
whether human review was required.

This supports reconstructability.

If an incident occurs, the organization should be able to determine whether context compaction removed evidence that mattered.

Context compaction is not bad.

Uncontrolled context compaction is dangerous.

For high-risk AI systems, compaction must preserve the conditions required for valid continuation.

23.5 Inspectable Context for High-Risk Workflows

High-risk AI workflows require inspectable context.

Inspectable context means that the deployer can identify, review and reconstruct the information supplied to the AI system and the condition of that information at the relevant point in the workflow.

This is essential for human oversight.

A human reviewer cannot meaningfully oversee an AI output if they cannot inspect the context that shaped it.

This is also essential for audit.

An auditor cannot reconstruct the decision trail if the organization cannot show what context the model saw.

It is essential for incident response.

An investigator cannot determine why an AI-supported workflow failed if the active context is unknown.

It is essential for affected-person review.

A person cannot meaningfully challenge an AI-supported outcome if the organization cannot identify what evidence influenced the pathway.

Inspectable context should include:

- context sources;
- source identifiers;
- retrieval timestamps;
- source last-updated timestamps;
- policy versions;
- data freshness;
- input completeness;
- user-entered information;
- system-generated summaries;
- compaction history;
- tool outputs;
- excluded context where relevant;
- warnings;
- uncertainty;
- limitations;
- authority state;
- human review status.

The goal is not to expose sensitive information unnecessarily.

The goal is to preserve enough internal evidence to reconstruct the AI-supported workflow.

Data protection, confidentiality, professional secrecy and security obligations must still be respected.

Inspectable does not mean public.

Inspectable means governable.

The deployer should be able to inspect context through appropriate access controls.

Different users may need different levels of visibility.

A front-line user may need to see freshness warnings.

A human overseer may need to see source evidence.

A compliance officer may need to see policy version and decision trail.

An auditor may need to see control records.

A technical team may need to see retrieval and tool-call logs.

A regulator may need selected evidence where lawful and required.

Inspectable context also supports refusal logic.

If context cannot be inspected, the workflow may not be admissible for high-consequence use.

For example:

If the system cannot identify which policy version it used, a compliance recommendation should not proceed with full authority.

If the system cannot show whether supplier data was current, a procurement approval should hold.

If the system cannot show which patient records were included, a clinical summary should not be treated as complete.

If the system cannot show whether a candidate ranking used approved criteria, an employment decision should not rely on it.

If the system cannot show what case evidence informed a public-service recommendation, the decision trail is weak.

Inspectable context should therefore become part of evidence of control.

A deployer should preserve a context record for high-risk workflows.

The context record may include:

- workflow identity;
- system identity;
- user role;
- prompt or instruction type;
- context bundle ID;
- source documents;
- retrieval method;
- timestamps;
- freshness status;
- compaction status;
- missing evidence warnings;
- policy version;
- authority state;
- output generated;
- human review;

final decision state.

This record supports the central Article 26 question:

Can the organization prove that the AI system was used under valid control conditions?

For context-heavy AI systems, that question cannot be answered unless the organization knows what context was used.

The future of AI governance will therefore require a shift.

Context should not be treated as temporary prompt material.

In high-risk workflows, context is part of the decision trail.

It must be governed, inspected, logged and preserved at a level proportionate to risk.

Chapter 23 Summary

Context management is not only an efficiency discipline.

For high-risk AI systems, context management is control management.

The context supplied to an AI system shapes what it can reason about, what it treats as current, what it treats as relevant and what authority it appears to support.

This chapter argues that context is part of the AI system's operating boundary.

Poor context management can create governance risk through stale facts, irrelevant history, duplicate information, outdated policy, missing limitations, unverified sources, poor retrieval, uncontrolled compaction and hidden loss of evidence.

The chapter explains stale context as a form of silent failure.

A workflow may complete successfully while relying on old facts, old approvals, old policies, stale supplier records, incomplete patient data or superseded case information.

The system may not fail visibly, but the decision environment has degraded.

The chapter also explains context hygiene as a governance control.

Good context hygiene should remove noise, mark drafts, preserve source references, exclude superseded material, separate verified from unverified information, identify conflicting evidence and maintain policy version integrity.

Context compaction receives special attention.

Summarization and compression are often necessary, but they can remove uncertainty, warnings, conditions, timestamps and evidence references. In high-risk workflows, compaction must preserve governance-critical context such as source, freshness, limitations, authority, pending checks and refusal conditions.

Finally, the chapter introduces inspectable context.

A deployer should be able to reconstruct what context the AI system received, where it came from, whether it was current, whether it was complete and whether it supported the authority attached to the output.

The central governance question is:

Did the system receive the right context for the consequence attached to its output?

If yes, continuation may be allowed.

If no, the workflow should hold, escalate or refuse.

If unknown, the workflow should not assume validity.

This is how context management becomes runtime governance.

Chapter 24

Reasoning Layers, Traceability and Hybrid AI

Legal anchor and framework boundary. [Articles 12-14](#); [Article 26](#). Reasoning-layer and hybrid-AI controls are operational proposals; the Regulation does not prescribe a single reasoning architecture.

24.1 Why Fluent Output Is Not Enough

Fluent output is not the same as trustworthy reasoning.

This distinction is becoming increasingly important in high-risk AI governance.

Modern AI systems can produce fluent explanations, structured summaries, confident recommendations, persuasive arguments and professional-looking reports. In many contexts, this fluency is useful. It can help users understand information, draft documents, compare options and work more efficiently.

But fluency can also create false confidence.

An output may be well written and still unsupported.

An explanation may sound logical but omit critical evidence.

A recommendation may be plausible but based on stale data.

A summary may be elegant but incomplete.

A decision-support output may appear reasoned but hide uncertainty.

A model may generate a convincing answer without exposing the facts, rules, constraints or assumptions that shaped it.

For high-risk AI systems, this is not enough.

The deployer's responsibility is not only to receive a useful output.

The deployer must be able to understand whether the output was generated under valid control conditions.

This requires more than language fluency.

It requires traceability.

It requires inspectable reasoning.

It requires a way to connect the output to:

- structured facts;

- source evidence;
- applicable rules;
- constraints;
- policy conditions;
- input data state;
- uncertainty;
- contradictions;
- human oversight;
- authority;
- decision boundaries.

This is where reasoning layers and hybrid AI become important.

A reasoning layer can help separate fluent generation from structured inference.

A language model may produce text.

A reasoning layer may help determine whether the text is supported by known facts, rules and constraints.

A knowledge graph may preserve relationships.

A rules engine may enforce policy boundaries.

A symbolic layer may identify contradictions.

A validation layer may check whether required evidence exists.

A control layer may determine whether continuation is allowed.

Together, these components can help move AI systems from plausible output toward accountable decision support.

This does not mean that language models become irrelevant.

Language models are powerful because they can interpret, summarize, translate, generate and interact naturally.

But in high-consequence environments, they should not carry the full burden of truth, reasoning and control alone.

A fluent answer should not be trusted simply because it is fluent.

The deployer should ask:

What facts support this output?

What rules apply?

What evidence was used?

What constraints were checked?

What contradictions were detected?

What assumptions were made?

What uncertainty remains?

Is the output allowed to influence consequence?

If the system cannot answer these questions, the output may not be suitable for high-risk use without further review.

For Article 26 deployers, this matters because appropriate use, human oversight, monitoring, logs and escalation all depend on understanding the system's output and its control conditions.

A human overseer cannot meaningfully review an output if the reasoning structure is hidden.

A monitor cannot detect failure if the system does not expose relevant signals.

A log cannot reconstruct control if it records only the final text.

An escalation pathway cannot operate if contradictions or missing evidence are invisible.

This is why fluent output is not enough.

High-risk AI governance requires outputs that are not only readable, but inspectable.

24.2 Reasoning Must Become Inspectable

Inspectable reasoning means that the organization can understand how an AI-supported output was shaped.

This does not necessarily mean that every internal model weight or hidden computation must be exposed.

It means that the deployer can inspect enough of the reasoning environment to determine whether the output is supported, bounded and governed.

For practical governance, inspectable reasoning should help answer:

What did the system know?

What did it infer?

Which evidence supported the inference?

Which rule or policy applied?

Which constraint limited the output?

Which contradiction was detected?

Which uncertainty remained?

Which control condition allowed continuation?

Which condition would have required hold, escalation or refusal?

This is especially important for human oversight.

A human reviewer cannot effectively supervise a black-box recommendation if they cannot see why the system produced it or what evidence it used.

The reviewer may accept the output because it sounds credible.

They may reject it based on intuition.

They may spend too much time reconstructing the evidence manually.

Or they may fail to notice that the system omitted a critical condition.

Inspectable reasoning helps the human reviewer move from passive approval to active control.

For example:

In procurement, the system should show which supplier evidence supported a recommendation, which checks passed, which checks are pending and which risk conditions remain unresolved.

In HR, the system should show which job criteria, candidate evidence and exclusion rules shaped a recommendation, without relying on inappropriate or prohibited data.

In healthcare, the system should show which patient records, clinical facts and missing-data warnings shaped a summary or risk flag.

In financial services, the system should show which policy version, transaction evidence and risk thresholds supported a compliance recommendation.

In public services, the system should show which case evidence, eligibility rule and human review point supported an administrative recommendation.

This does not require every AI system to become fully symbolic.

But it does require deployers to separate output from support.

A high-risk output should not stand alone.

It should be accompanied by enough reasoning evidence to support oversight and reconstruction.

Inspectable reasoning may include:

- cited sources;
- structured facts;
- rule references;
- policy version;
- data freshness state;
- confidence or uncertainty indicator;
- missing evidence;
- contradiction flags;
- human review requirement;
- refusal reason;
- escalation trigger;
- decision boundary.

The important point is that reasoning should be operationally useful.

A long natural-language explanation may not be enough if it does not expose the control-relevant structure.

For governance, the system should not merely explain itself after the fact.

It should expose the reasoning conditions before the output is trusted.

That is the shift.

Reasoning should not be something the organization infers retrospectively.

It should be something the system can make inspectable at the point of use.

This is where reasoning layers can support Article 26.

They can make human oversight more meaningful.

They can improve monitoring.

They can support logs and decision trails.

They can identify contradictions.

They can support refusal when evidence does not support continuation.

They can help preserve evidence of control.

The goal is not only explainability.

The goal is governability.

24.3 Structured Facts, Rules and Constraints

A reasoning layer becomes valuable when it helps the system distinguish between facts, rules, constraints, assumptions and generated language.

This distinction is important because high-risk AI governance depends on knowing what kind of information the system is using.

A fact is a statement about the world or a record.

For example:

A supplier certification expires on a specific date.

A candidate has a particular qualification.

A patient has a recorded allergy.

A policy version became effective on a specific date.

A customer account has a fraud flag.

A public-service applicant submitted a document.

A rule defines what should happen under certain conditions.

For example:

A supplier above a certain risk level requires additional review.

A hiring recommendation cannot be based on prohibited characteristics.

A clinical alert requires clinician review before patient communication.

A refund above a threshold requires manager approval.

A public-service eligibility decision requires current case evidence.

A constraint defines a boundary.

For example:

The system may summarize but not decide.

The system may draft but not send.

The system may recommend but not approve.

The system may use approved data sources only.

The system may not proceed if evidence freshness is unknown.

An assumption is something the system treats as true or useful, but that may require verification.

For example:

The supplier profile is current.

The reviewer has authority.

The policy version is active.

The applicant record is complete.

The patient record contains all relevant recent information.

The user's request is within intended purpose.

Generated language is the model's expression of output.

It may combine facts, rules, constraints and assumptions into a fluent response.

The governance problem occurs when these categories collapse into one another.

A generated sentence may sound like a fact.

An assumption may be treated like evidence.

A summary may be treated like a rule.

A recommendation may be treated like authority.

A constraint may be omitted from the output.

A missing fact may be hidden by fluent language.

Reasoning layers can help prevent this collapse.

They can preserve structure.

They can show:

This is a verified fact.

This is an applicable rule.

This is a constraint.

This is an unresolved assumption.

This is generated explanation.

This is missing evidence.

This is a contradiction.

This is a refusal condition.

This structure is important for deployers.

It helps them evaluate whether the output is admissible.

For example, a procurement recommendation may say:

“Supplier A is recommended.”

A structured reasoning layer should show:

- Supplier A passed financial review.
- Sanctions check is current.
- Cybersecurity review is pending.
- Contract value exceeds standard approval threshold.
- Recommendation is allowed only for preliminary review.
- Final approval requires escalation.

Without that structure, the recommendation may be misleading.

In HR, an AI system may say:

“Candidate B is a strong match.”

A structured reasoning layer should show:

- Match is based on required skills.
- Missing qualification evidence remains unresolved.
- No prohibited criteria were used.
- Human review required before shortlisting.
- Candidate ranking should not be used as automatic rejection.

In healthcare, an AI assistant may say:

“Patient appears stable.”

A structured reasoning layer should show:

- Based on records available up to a specific timestamp.

- Latest lab results not included.
- Medication history current.
- Clinician review required.
- Output is not a diagnosis.

The point is not to make every output longer.

The point is to make the reasoning support inspectable.

Structured reasoning allows the deployer to preserve the difference between what is known, what is inferred, what is required and what remains uncertain.

This is essential for evidence of control.

24.4 Refusal When the World Model Does Not Support Continuation

One of the most important functions of a reasoning layer is refusal.

A system should not always answer.

A system should not always recommend.

A system should not always proceed.

In high-risk AI governance, refusal is a control feature.

A reasoning layer can help determine when the world model does not support continuation.

The “world model” in this context does not need to mean a complete representation of reality.

It means the structured representation of relevant facts, rules, relationships, constraints and current conditions needed for the workflow.

If the world model is incomplete, stale, contradictory or outside intended purpose, the system should not behave as if continuation is fully justified.

For example:

A procurement system should refuse to recommend final approval if supplier risk data is stale or required checks are pending.

A HR system should refuse to generate rejection decisions if the candidate ranking lacks required human review or uses unsupported criteria.

A clinical assistant should refuse to provide patient-facing advice if critical clinical context is missing.

A compliance assistant should refuse to apply an outdated policy version.

A public-service AI system should refuse to support eligibility action if required case evidence is missing.

An agentic workflow should refuse to execute a tool call if authority is absent or downstream consequence is uncontrolled.

This is where reasoning layers become governance layers.

They can help the system detect:

- missing facts;
- expired evidence;
- conflicting rules;
- unsupported assumptions;
- invalid authority;
- scope mismatch;
- prohibited use;
- unresolved risk;
- insufficient human oversight;
- downstream consequence beyond permission.

When these conditions appear, the system should not hide them inside a fluent output.

It should expose them.

The output may be:

HOLD: required evidence is missing.

ESCALATE: rule conflict detected.

REFUSE: use is outside intended purpose.

REDUCE AUTHORITY: fallback model used.

HUMAN REVIEW REQUIRED: consequence level exceeds automated support.

This connects reasoning layers directly to the ALLOW, HOLD, ESCALATE and REFUSE model developed earlier in this study.

A reasoning layer can help determine which state applies.

This is especially important because language models may be optimized to answer.

In high-risk AI systems, answering is not always appropriate.

A system that always produces an answer may be less safe than a system that knows when the evidence does not support an answer.

Refusal should also be explainable.

A refusal should not merely say:

Cannot proceed.

It should say, where appropriate:

- what condition failed;
- what evidence is missing;
- which rule or constraint applies;
- what action may restore continuation;
- who should review;
- whether escalation is required.

For example:

“Cannot recommend final supplier approval because cybersecurity review is pending and contract value exceeds the approval threshold.”

“Cannot support candidate rejection because the ranking has not been reviewed by an authorized human reviewer.”

“Cannot summarize this clinical case as complete because the latest lab results are not included.”

“Cannot apply this policy because the retrieved version is superseded.”

“Cannot execute this workflow because the agent does not have authority to write to the downstream system.”

These refusals preserve control.

They also create evidence.

A refusal log shows that the system did not simply continue under invalid conditions.

It recognized a boundary.

It preserved the boundary.

It created an opportunity for human review, evidence refresh or escalation.

This is one of the strongest forms of runtime governance.

24.5 Hybrid AI as Governance Infrastructure

Hybrid AI combines different AI and reasoning approaches.

This may include language models, knowledge graphs, rule engines, symbolic reasoning, retrieval systems, validation layers, constraint solvers, policy engines, monitoring tools and human review workflows.

The point is not to choose one approach against another.

The point is to use different components for what they are good at.

Language models are strong at language, summarization, interaction, drafting, interpretation and flexible reasoning.

Knowledge graphs are strong at representing relationships.

Rule engines are strong at applying explicit conditions.

Retrieval systems are strong at bringing evidence into context.

Validation layers are strong at checking required fields, freshness and constraints.

Monitoring systems are strong at detecting change.

Human reviewers are essential for judgment, accountability, context and intervention.

A high-risk AI governance architecture may need all of these.

This is why hybrid AI should be understood not only as a technical design choice, but as governance infrastructure.

Hybrid AI can help answer:

What facts are known?

Which rules apply?

What evidence supports the output?

What constraints limit the workflow?

What context is current?

What contradictions exist?

What authority applies?

What human review is required?

What should be logged?

Should the workflow continue?

This supports Article 26 deployer obligations.

Use according to instructions can be supported by rule-based intended-purpose boundaries.

Human oversight can be supported by structured reasoning evidence.

Input data relevance can be supported by validation layers and freshness checks.

Monitoring can be supported by runtime signals and drift detection.

Logs can be supported by decision trails.

Escalation can be supported by explicit trigger logic.

Affected-person information can be supported by traceable system role descriptions.

Hybrid AI does not eliminate risk.

It can introduce its own complexity.

A knowledge graph may be incomplete.

A rule engine may be outdated.

A retrieval system may select poor sources.

A reasoning layer may encode wrong assumptions.

A validation layer may miss context.

A human reviewer may over-rely on structured outputs.

Therefore, hybrid AI must also be governed.

But hybrid AI offers an important advantage:

It can separate roles.

The language model does not need to be the sole source of truth.

The rule engine can enforce boundaries.

The knowledge graph can preserve relationships.

The retrieval layer can identify sources.

The validation layer can check freshness.

The monitoring layer can detect drift.

The human can judge and intervene.

This separation makes control more visible.

It also makes failure easier to locate.

If the model output is wrong, was the context wrong?

Was the rule outdated?

Was the evidence stale?

Was the graph incomplete?

Was the human review missing?

Was authority invalid?

Was downstream automation uncontrolled?

A hybrid architecture can support this investigation better than a single opaque output.

For deployers, the governance goal is not to build the most complex architecture.

The goal is to build an architecture that matches consequence.

Low-risk workflows may not require a full reasoning layer.

High-risk workflows may.

Critical workflows may require structured facts, explicit rules, evidence freshness, human control gates, refusal logic and reconstructable decision trails.

The central principle is proportionality.

The more consequential the AI-supported workflow, the more important traceable reasoning becomes.

In this sense, hybrid AI may become one of the most important foundations for operational AI governance.

Not because it makes AI perfect.

But because it can make AI more inspectable, bounded and governable.

Chapter 24 Summary

Fluent output is not enough for high-risk AI governance.

An AI system may produce a clear, confident and professional-looking answer while hiding missing evidence, stale context, unsupported assumptions, contradictions or invalid authority.

This chapter argues that reasoning layers, traceability and hybrid AI can help move AI systems from fluent generation toward accountable decision support.

Inspectable reasoning is central.

A deployer should be able to understand what facts, rules, constraints, assumptions and evidence shaped an output. This does not require exposing every internal model computation, but it does require enough structure to support human oversight, monitoring, logging, escalation and reconstruction.

The chapter distinguishes facts, rules, constraints, assumptions and generated language.

This distinction matters because governance failures occur when these categories collapse. An assumption may be treated as fact. A generated summary may be treated as evidence. A recommendation may be treated as authority.

Reasoning layers can help preserve these differences.

The chapter also explains refusal as a key governance feature.

When the world model does not support continuation because evidence is missing, stale, contradictory or outside intended purpose, the system should not simply produce a fluent answer. It should hold, escalate or refuse.

Finally, the chapter presents hybrid AI as governance infrastructure.

Language models, knowledge graphs, rule engines, retrieval systems, validation layers, monitoring systems and human review workflows can work together to create more inspectable and bounded AI-supported processes.

The central governance question is:

Can the deployer inspect enough of the reasoning structure to know whether the output is supported, bounded and allowed to influence consequence?

If yes, the output may support governed decision-making.

If no, fluency alone should not be trusted.

For Article 26 deployers, this is the deeper lesson:

The future of trustworthy AI will not be built only on better answers.

It will be built on systems that can show why an answer was allowed to matter.

Chapter 25

Article 26 Deployer Readiness Matrix

Legal anchor and framework boundary. [Article 26](#). The readiness matrix is a practical implementation aid and does not replace a legal applicability or conformity assessment.

25.1 Purpose of the Readiness Matrix

The purpose of the Article 26 Deployer Readiness Matrix is to help organizations translate deployer obligations into practical governance evidence.

Article 26 is not only a legal checklist.

It is an operational readiness test.

A deployer of a high-risk AI system must be able to show that the system is used under appropriate technical and organisational conditions, according to instructions, with meaningful human oversight, suitable input data, active monitoring, retained logs, escalation pathways and relevant information duties where required.

The readiness matrix converts these obligations into a structured assessment.

It helps the organization answer:

Are we ready to deploy?

Are we ready to operate?

Are we ready to monitor?

Are we ready to intervene?

Are we ready to explain?

Are we ready to prove control?

This matters because AI governance maturity is often uneven.

An organization may have strong policies but weak logs.

It may have a system register but no decision trail.

It may have human oversight on paper but no evidence that humans can intervene.

It may have monitoring dashboards but no escalation logic.

It may have provider documentation but no internal translation into workflow controls.

It may have incident procedures but no tested AI escalation pathway.

It may have transparency notices but no evidence that those notices still match actual use.

The readiness matrix is designed to expose these gaps.

It should not be used only once.

It should be used:

- before deployment;
- during procurement;
- during system integration;
- before expanding use;
- after major provider updates;
- after model substitution;
- after workflow change;
- after incidents or near misses;
- during periodic reassessment;
- during audit preparation;
- before board or regulator reporting.

The matrix should be applied proportionately.

Not every AI system requires the same depth of evidence.

A low-risk internal drafting assistant may require lighter controls.

A high-risk AI system affecting employment, healthcare, finance, public services, education, critical infrastructure or regulated processes requires stronger evidence.

The matrix is therefore not a rigid template.

It is a governance tool.

Its purpose is to help the deployer identify whether the system is controlled enough for the consequence attached to its output.

The central question is:

Can the organization prove that this AI system is used under valid control conditions?

If yes, deployment may be ready.

If no, deployment may require remediation.

If uncertain, the system should not be treated as fully ready.

25.2 Readiness Domains

The readiness matrix is organized around twelve domains.

Each domain corresponds to an operational control area required for mature Article 26 deployer governance.

The twelve domains are:

- System Identification and Ownership
- Intended Purpose and Actual Use
- Instructions for Use
- Human Oversight
- Input Data Governance
- Runtime Monitoring
- Logs and Decision Trails
- Authority, Scope and Boundaries
- Risk, Incident and Escalation Pathways
- Workplace and Affected-Person Information
- Third-Party and Provider Dependency
- Evidence of Control and Reassessment

Each domain should be assessed separately.

This is important because readiness is not binary.

An organization may be mature in one domain and weak in another.

For example:

The system may be well identified, but actual use may have drifted.

Provider instructions may be stored, but not translated into internal controls.

Human oversight may be assigned, but reviewers may lack time or authority.

Input data may be available, but freshness thresholds may be undefined.

Monitoring may exist, but escalation triggers may be missing.

Logs may be retained, but decision trails may not be reconstructable.

Affected persons may be informed, but notices may not reflect current use.

Third-party contracts may exist, but provider incident channels may not be tested.

The matrix should make these differences visible.

A single “ready” label is not enough.

The deployer needs to know where readiness is strong, where it is partial, and where deployment should hold.

A practical scoring model may use five levels:

Level 0 — Not Established

No clear control exists.

The organization cannot show ownership, process, evidence or operational implementation.

Level 1 — Documented Intention

A policy, plan, document or owner exists, but the control is not yet operationally embedded.

Level 2 — Implemented Process

A process exists and is used, but evidence may be incomplete, manual, inconsistent or not connected to runtime control.

Level 3 — Evidenced Control

The control operates in the workflow and produces evidence that supports reconstruction.

Level 4 — Runtime-Assured Control

The control is monitored during operation, connected to ALLOW, HOLD, ESCALATE or REFUSE states, and subject to periodic reassessment.

This maturity scale allows the deployer to distinguish between paperwork and operational control.

The goal is not always to reach Level 4 in every domain immediately.

The goal is to match maturity to risk.

For a high-risk AI system with significant consequence, several domains should reach Level 3 or Level 4 before deployment or continued use.

For an agentic, high-consequence or regulated workflow, runtime-assured controls may be necessary in key areas such as authority, input data, human oversight, monitoring, logs and escalation.

The readiness matrix should therefore produce both a score and a decision.

Possible decisions include:

- Ready for limited pilot;
- Ready for controlled deployment;
- Ready for production use;
- Ready only with reduced authority;
- Hold pending remediation;
- Escalate for governance approval;
- Refuse deployment under current conditions.

This makes the matrix actionable.

It does not only describe maturity.

It supports governance decisions.

25.3 Domain 1 — System Identification and Ownership

The first readiness domain is system identification and ownership.

An organization cannot govern an AI system it cannot identify.

The deployer should know what AI systems are being used, where they are used, who owns them and what processes they affect.

Minimum evidence should include:

- system name;
- provider;
- internal owner;

- business owner;
- technical owner;
- deployment environment;
- high-risk status;
- intended purpose;
- actual use;
- affected process;
- affected persons or groups;
- user groups;
- data sources;
- downstream systems;
- deployment date;
- version history.

Readiness questions include:

Is the AI system recorded in an internal register?

Is the provider identified?

Is ownership assigned?

Is the system's use case described clearly?

Is the risk classification recorded?

Is the affected workflow mapped?

Are affected persons or groups identified?

Are downstream systems identified?

Is there a review owner?

Is there a retirement or decommissioning owner?

A Level 0 organization may have no central record of the system.

A Level 1 organization may know the system exists but have incomplete ownership.

A Level 2 organization may have a register, but it may not link to workflows or evidence.

A Level 3 organization has system records connected to controls, logs, oversight and monitoring.

A Level 4 organization has a live system register connected to runtime state, review dates, incident history and reassessment triggers.

The key evidence question is:

Can the organization identify the AI system and the accountable owners for its use?

If the answer is no, the system is not ready for high-risk deployment.

25.4 Domain 2 — Intended Purpose and Actual Use

The second readiness domain is intended purpose and actual use.

The provider defines the intended purpose.

The deployer controls actual use.

The risk appears when actual use drifts away from intended purpose.

Minimum evidence should include:

- provider-defined intended purpose;
- internal approved use case;
- actual workflow use;
- prohibited uses;
- user groups;
- data sources;
- output role;
- consequence level;
- downstream integrations;
- use-case change history;
- periodic reassessment records.

Readiness questions include:

What does the provider say the system is for?

What does the organization actually use it for?

Does actual use match intended purpose?

Are prohibited uses defined?

Has use expanded since deployment?

Has the output become more influential over time?

Has the system been connected to new workflows?

Has the consequence level changed?

Does the current notice or policy still match actual use?

A Level 0 organization cannot clearly describe intended purpose or actual use.

A Level 1 organization has provider documentation but no internal use mapping.

A Level 2 organization has mapped intended purpose to initial use, but does not actively monitor drift.

A Level 3 organization can show actual use remains within approved scope.

A Level 4 organization monitors use drift and triggers HOLD or reassessment when actual use changes.

The key evidence question is:

Can the organization prove that actual use remains within the conditions that made deployment appropriate?

If not, the system may require reassessment before continued use.

25.5 Domain 3 — Instructions for Use

The third readiness domain is instructions for use.

Provider instructions should not remain in a document repository.

They must be translated into internal controls.

Minimum evidence should include:

- provider instructions;
- instruction version history;
- internal review of instructions;
- mapping from instructions to internal procedures;
- user guidance;
- training materials;
- system configuration records;
- limitations communicated to users;
- monitoring linked to instruction requirements;
- records of instruction updates and internal changes.

Readiness questions include:

Have provider instructions been received and stored?

Has the organization reviewed them?

Have they been translated into internal procedures?

Do users understand what the system is and is not for?

Are limitations visible to users?

Are human oversight requirements implemented?

Are data requirements implemented?

Are logging and monitoring requirements implemented?

Are provider updates tracked?

Does the organization reassess use when instructions change?

A Level 0 organization does not have the instructions.

A Level 1 organization has stored the instructions but not operationalized them.

A Level 2 organization has internal procedures, but evidence of use may be incomplete.

A Level 3 organization can show instructions are reflected in workflow controls.

A Level 4 organization monitors instruction compliance and triggers reassessment when actual use departs from instructions.

The key evidence question is:

Can the organization show how provider instructions became operational controls?

If the answer is no, Article 26 readiness is weak.

25.6 Domain 4 — Human Oversight

The fourth readiness domain is human oversight.

Human oversight is not merely human presence.

It requires competence, training, authority and support.

Minimum evidence should include:

- assigned oversight roles;
- reviewer competence criteria;
- training records;
- authority records;
- access to evidence;
- review procedure;
- override capability;
- escalation route;
- intervention logs;
- review quality monitoring;
- records of disagreement, modification or refusal.

Readiness questions include:

Who is assigned to human oversight?

Do they have the right competence?

Have they received system-specific training?

Do they understand the system's limitations?

Can they access source evidence?

Can they override the AI output?

Can they stop or delay the workflow?

Do they have authority to escalate?

Is review performed before consequence attaches?

Is intervention recorded?

Is automation bias monitored?

A Level 0 organization has no defined oversight.

A Level 1 organization names a human role but does not define competence, authority or workflow action.

A Level 2 organization has review procedures, but review quality is not evidenced.

A Level 3 organization can show meaningful human review, override and escalation evidence.

A Level 4 organization monitors oversight effectiveness, override rates, review timing and automation bias, and triggers reassessment when oversight weakens.

The key evidence question is:

Could the human meaningfully change the outcome before consequence attached?

If not, human oversight may be symbolic.

25.7 Domain 5 — Input Data Governance

The fifth readiness domain is input data governance.

Where the deployer controls input data, it must ensure that input data is relevant and sufficiently representative in view of the intended purpose.

Minimum evidence should include:

- input data source map;
- data owner;
- source reliability;
- freshness thresholds;
- completeness checks;
- missing-data rules;
- representativeness assessment where relevant;
- data validation logs;
- user-entered context controls;
- warning states;
- data quality monitoring.

Readiness questions include:

What data does the system receive?

Who controls the data?

Is the data relevant to the intended purpose?

Is it sufficiently representative?

Is it fresh enough?

Are required fields defined?

Are missing fields flagged?

Are source systems reliable?

Are user-entered inputs distinguished from verified records?

Are stale evidence conditions visible to users?

Can the workflow hold when data is insufficient?

A Level 0 organization has no clear map of input data.

A Level 1 organization knows the main sources but has no suitability controls.

A Level 2 organization has data checks, but they are manual or incomplete.

A Level 3 organization can show data relevance, freshness and completeness evidence for workflow use.

A Level 4 organization monitors input conditions at runtime and triggers HOLD, ESCALATE or REFUSE when data conditions fail.

The key evidence question is:

Was the AI system given input data that could support a valid output for the intended purpose?

If not, output confidence should not be trusted.

25.8 Domain 6 — Runtime Monitoring

The sixth readiness domain is runtime monitoring.

Monitoring is not only technical observability.

It must detect whether the system remains under valid control conditions.

Minimum evidence should include:

- monitoring plan;
- monitored signals;
- technical performance metrics;
- input data monitoring;
- workflow drift monitoring;
- human oversight monitoring;
- authority drift monitoring;
- downstream effect monitoring;
- alert thresholds;
- response procedures;
- monitoring review records.

Readiness questions include:

What is monitored?

Does monitoring include more than uptime?

Are input data conditions monitored?

Is actual use monitored?

Is human oversight availability monitored?

Are override and escalation patterns monitored?

Are downstream integrations monitored?

Are authority drift signals monitored?

Are alerts connected to action?

Who reviews monitoring outputs?

Can monitoring trigger HOLD, ESCALATE or REFUSE?

A Level 0 organization has no monitoring.

A Level 1 organization monitors only technical availability.

A Level 2 organization monitors performance but not governance conditions.

A Level 3 organization monitors relevant control conditions and keeps review records.

A Level 4 organization connects monitoring signals to runtime governance states and reassessment triggers.

The key evidence question is:

Can the organization detect when continued use is no longer justified?

If not, monitoring is incomplete.

25.9 Domain 7 — Logs and Decision Trails

The seventh readiness domain is logs and decision trails.

Logs should support reconstructability.

They should not only show activity.

They should help prove control.

Minimum evidence should include:

- logging capability;
- log retention policy;
- automatically generated logs under deployer control;
- system identity in logs;
- input condition records;
- output records;
- human review records;
- authority state;
- escalation records;
- downstream consequence;
- integrity controls;
- retention period;
- audit access process.

Readiness questions include:

What logs are generated?

Which logs are under deployer control?

How long are logs retained?

Do logs identify system version?

Do logs preserve input condition?

Do logs preserve output?

Do logs preserve human review?

Do logs preserve authority state?

Do logs show escalation or refusal?

Can logs reconstruct decision trails?

Are logs protected from alteration?

A Level 0 organization has no meaningful log retention.

A Level 1 organization stores technical logs only.

A Level 2 organization retains logs but cannot reconstruct decision context.

A Level 3 organization can reconstruct system, input, output, human review and final decision.

A Level 4 organization maintains decision trails connected to runtime state, authority, escalation and evidence freshness.

The key evidence question is:

Can the organization reconstruct why the AI-supported workflow was allowed to continue?

If not, logs may not be sufficient for evidence of control.

25.10 Domain 8 — Authority, Scope and Boundaries

The eighth readiness domain is authority, scope and boundaries.

The deployer must know who or what is allowed to move an AI-supported workflow toward consequence.

Minimum evidence should include:

- authority matrix;
- approval thresholds;
- user permissions;
- system permissions;
- tool permissions;
- approved output role;
- prohibited actions;
- boundary conditions;
- downstream integration map;
- authority validation records;
- scope-change records.

Readiness questions include:

Who has authority to approve?

What is the AI system allowed to do?

Is the output advisory, influential or executable?

Can the system trigger downstream actions?

Are permission boundaries clear?

Are budget, risk or consequence thresholds mapped?

Can authority expire or change?

Are authority changes monitored?

Are model fallback authority rules defined?

Can the workflow refuse when authority is absent?

A Level 0 organization has no mapped authority model.

A Level 1 organization has formal approval roles but no AI-specific authority mapping.

A Level 2 organization maps authority but does not monitor drift.

A Level 3 organization can show authority validation before consequence.

A Level 4 organization monitors authority drift, scope expansion and downstream effects at runtime.

The key evidence question is:

Was the workflow still authorized to continue under current conditions?

If not, continuation should be questioned.

25.11 Domain 9 — Risk, Incident and Escalation Pathways

The ninth readiness domain is risk, incident and escalation pathways.

Monitoring must connect to action.

Minimum evidence should include:

- AI incident procedure;
- serious risk criteria;
- escalation matrix;
- provider contact route;
- internal owner;
- suspension procedure;
- evidence preservation process;
- affected-case review process;
- serious incident assessment process;
- escalation tests;
- incident logs;
- corrective action records.

Readiness questions include:

What counts as an AI risk signal?

Who can report it?

Who triages?

Who decides whether use continues?

Who contacts the provider?

Who preserves evidence?

Who decides suspension?

Who handles serious incident assessment?

Are near misses logged?

Are escalation pathways tested?

Can the system enter HOLD or REFUSE state?

A Level 0 organization has no AI-specific escalation path.

A Level 1 organization has general incident processes but no AI mapping.

A Level 2 organization has an AI incident process but it is not tested.

A Level 3 organization has tested escalation and evidence preservation.

A Level 4 organization connects runtime signals to escalation, suspension, provider notification and corrective action.

The key evidence question is:

Did the organization know when monitoring required action?

If not, escalation readiness is weak.

25.12 Domain 10 — Workplace and Affected-Person Information

The tenth readiness domain is workplace and affected-person information.

Where required, workers, worker representatives and affected natural persons must receive meaningful information about relevant high-risk AI use.

Minimum evidence should include:

information-duty assessment;

worker notification records;

worker representative communication;
affected-person information records;
notice content;
timing records;
explanation of AI role;
human oversight description;
question or challenge route;
update records when use changes.

Readiness questions include:

Does the system affect workers?

Have workers and representatives been informed where required?

Does the system assist decisions relating to natural persons?

Have affected persons been informed where required?

Does the notice explain the AI system's role?

Does the notice reflect actual use?

Is there a route for questions or concerns?

Are notices updated when use changes?

Can the organization prove notification timing?

A Level 0 organization has no information-duty assessment.

A Level 1 organization has generic AI notices only.

A Level 2 organization has system-specific notices but does not monitor continued accuracy.

A Level 3 organization can show timely and meaningful notification records.

A Level 4 organization monitors actual use against notices and triggers updates when use changes.

The key evidence question is:

Were affected people informed in a way that made the AI system's role visible and understandable?

If not, transparency may be formal but not meaningful.

25.13 Domain 11 — Third-Party and Provider Dependency

The eleventh readiness domain is third-party and provider dependency.

High-risk AI governance depends on provider information, system updates, support channels, logging capabilities, incident communication and contractual controls.

Minimum evidence should include:

- provider identity;
- contract records;
- provider instructions;
- support contact;
- incident contact;
- update notification process;
- model dependency register;
- fallback registry;
- data processing terms;
- subcontractor information where relevant;
- audit support terms;
- service-level expectations;
- provider change records.

Readiness questions include:

Who is the provider?

What information has the provider supplied?

Are instructions complete enough for deployment?

How are provider updates handled?

How are model changes communicated?

What happens if the provider system is unavailable?

Are fallback models approved?

Who is contacted during incidents?

Are logs available to the deployer?

Are third-party dependencies mapped?

Can the organization continue safely if the provider changes?

A Level 0 organization has no structured provider dependency record.

A Level 1 organization has contracts but no operational dependency mapping.

A Level 2 organization maps providers but not fallback, incident or update pathways.

A Level 3 organization has provider dependency controls connected to system governance.

A Level 4 organization monitors provider dependency, model availability, fallback status and authority continuity at runtime.

The key evidence question is:

Can the organization prove that provider and model dependencies are governed rather than hidden?

If not, third-party risk may weaken Article 26 readiness.

25.14 Domain 12 — Evidence of Control and Reassessment

The twelfth readiness domain is evidence of control and reassessment.

This domain asks whether the organization can prove that control conditions remain active, valid and inspectable over time.

Minimum evidence should include:

- evidence map;
- control owners;
- evidence owners;
- decision-trail samples;
- reassessment schedule;
- reassessment triggers;
- review records;
- remediation records;
- board or governance reporting;
- audit readiness pack;
- system retirement criteria.

Readiness questions include:

For each Article 26 obligation, what evidence proves control?

Who owns the evidence?

Where is it stored?

How is it reviewed?

What triggers reassessment?

Can the organization reconstruct a specific case?

Can it show continued validity?

Can it show when conditions changed?

Can it show remediation?

Can it show why the system remains allowed to operate?

A Level 0 organization has no evidence model.

A Level 1 organization has documents but no connected evidence chain.

A Level 2 organization has evidence, but it is fragmented across teams.

A Level 3 organization has connected evidence of system, workflow, data, oversight, logs and escalation.

A Level 4 organization has live evidence of control connected to runtime governance, reassessment and reporting.

The key evidence question is:

Can the organization prove that the system remains under valid control conditions over time?

This is the final readiness test.

25.15 Readiness Matrix Table

The following matrix provides a practical assessment structure.

Domain	Level 0	Level 1	Level 2	Level 3	Level 4
System Identification	Unknown or unmanaged system	Basic system record	Register exists	Register linked to controls	Live register with runtime state
Intended Purpose	No clear purpose	Provider purpose stored	Initial use mapped	Actual use evidenced	Drift monitored and controlled

Domain	Level 0	Level 1	Level 2	Level 3	Level 4
Instructions for Use	Instructions missing	Instructions stored	Internal procedure created	Workflow controls mapped	Instruction compliance monitored
Human Oversight	No oversight	Named human only	Review process exists	Intervention evidenced	Oversight quality monitored
Input Data	Data sources unknown	Sources identified	Data checks exist	Data validity evidenced	Runtime data controls active
Monitoring	No monitoring	Technical monitoring only	Performance monitoring	Governance signals monitored	Monitoring triggers control states
Logs	No useful logs	Technical logs only	Logs retained	Decision trail reconstructable	Runtime evidence trail active
Authority	Authority unclear	Formal roles exist	Authority mapped	Authority validated	Authority drift monitored
Escalation	No AI escalation	General incident path	AI path documented	Tested escalation	Runtime escalation active
Information Duties	Not assessed	Generic notice	System-specific notice	Timely evidence retained	Notices monitored against actual use
Third Parties	Dependencies unknown	Contracts exist	Dependencies mapped	Provider controls integrated	Model/fallback dependency monitored
Evidence of Control	No evidence model	Documents exist	Fragmented evidence	Connected evidence chain	Live evidence and reassessment

This table should not be used mechanically.

A Level 4 score in one domain does not compensate for a Level 0 score in another critical domain.

For example:

Strong logging does not compensate for absent human oversight.

Strong provider documentation does not compensate for unsuitable input data.

Strong monitoring does not compensate for no escalation pathway.

Strong notification does not compensate for use outside intended purpose.

Strong system inventory does not compensate for weak evidence of control.

The matrix should be used to identify domain-specific readiness.

The final deployment decision should consider consequence, risk, regulatory context and control gaps.

25.16 Readiness Decision Logic

After completing the matrix, the deployer should make an explicit readiness decision.

A practical decision model may include five outcomes.

1. Ready for Controlled Pilot

The system may be tested in a limited environment with restricted users, limited consequence, enhanced monitoring and clear rollback.

This may be appropriate where core controls are partially implemented but production consequence is not yet allowed.

2. Ready for Production with Conditions

The system may be deployed, but only with defined conditions.

Conditions may include human review, limited scope, no downstream automation, enhanced logging, or evidence freshness requirements.

3. Ready for Full Approved Use

The system may be used for the approved workflow and consequence level because required controls are operational and evidenced.

This does not mean no further review is needed.

It means current deployment is justified under defined conditions.

4. Hold Pending Remediation

The system should not proceed to the proposed use until specific gaps are remediated.

HOLD may apply where input data, human oversight, logging, authority, notification or escalation controls are incomplete.

5. Refuse Deployment Under Current Conditions

The system should not be deployed or used for the proposed purpose because control conditions cannot support the consequence.

This may apply where use is outside intended purpose, human oversight is not possible, evidence cannot be reconstructed, provider dependency is uncontrolled, input data is unsuitable, or serious risk cannot be mitigated.

The decision should be documented.

The record should include:

- assessment date;
- system assessed;
- assessors;
- domain scores;
- critical gaps;
- decision outcome;
- conditions;
- remediation actions;
- responsible owners;
- due dates;
- next review date;
- escalation decision where required.

This record becomes part of evidence of control.

It shows that the organization did not simply deploy the system because it was available.

It assessed readiness.

It identified gaps.

It made a governance decision.

The readiness decision should also be revisited.

A system that is ready today may not remain ready tomorrow.

Readiness should be reassessed when:

provider instructions change;
model changes;
fallback occurs;
use expands;
data sources change;
workflow changes;
downstream integration changes;
human oversight changes;
authority changes;
incident or near miss occurs;
monitoring signals deteriorate;
affected-person complaints increase;
regulatory or internal policy changes occur.

Readiness is not permanent.

It is a continuing condition.

25.17 Minimum Evidence Pack

The readiness matrix should produce a minimum evidence pack for each high-risk AI system.

This pack should be proportionate to risk, but for high-risk or high-consequence systems it should include:

System record

Provider instructions

Intended purpose and actual use map

Risk classification

Workflow map

Data source map

Input data suitability evidence

Human oversight assignment

Human oversight training records

Authority and approval map

Monitoring plan

Logging and retention plan

Decision trail sample

Escalation procedure

Incident contact records

Workplace or affected-person information assessment

Third-party dependency map

Model dependency and fallback rules where relevant

Reassessment triggers

Readiness decision record

This minimum evidence pack should answer:

What is the system?

What is it used for?

Why is that use allowed?

Who owns it?

Who oversees it?

What data does it use?

What output does it produce?

What consequence can follow?

What controls exist?

What evidence proves those controls?

What happens when conditions fail?

When will the system be reassessed?

This evidence pack should not be created only for audit.

It should be used operationally.

Governance teams should be able to review it.

Human overseers should be able to rely on relevant parts of it.

Risk teams should be able to identify control gaps.

Internal audit should be able to test it.

Senior management should be able to understand the readiness state.

If the evidence pack cannot answer these questions, the organization may have documentation but not readiness.

Chapter 25 Summary

The Article 26 Deployer Readiness Matrix is a practical tool for translating deployer obligations into operational evidence.

It helps organizations assess whether a high-risk AI system is ready for deployment, continued use, expansion, audit or reassessment.

The matrix is organized around twelve domains:

System Identification and Ownership

Intended Purpose and Actual Use

Instructions for Use

Human Oversight

Input Data Governance

Runtime Monitoring

Logs and Decision Trails

Authority, Scope and Boundaries

Risk, Incident and Escalation Pathways

Workplace and Affected-Person Information

Third-Party and Provider Dependency

Evidence of Control and Reassessment

Each domain is assessed across five maturity levels:

Level 0: Not Established

Level 1: Documented Intention

Level 2: Implemented Process

Level 3: Evidenced Control

Level 4: Runtime-Assured Control

The matrix helps distinguish between paperwork and operational governance.

A policy may exist, but the control may not operate.

A log may exist, but the decision trail may not be reconstructable.

A human may be named, but meaningful intervention may not be possible.

A system may be approved, but actual use may have drifted.

The readiness matrix therefore supports deployment decisions such as controlled pilot, production with conditions, full approved use, hold pending remediation or refusal under current conditions.

The central readiness question is:

Can the organization prove that this AI system is used under valid control conditions, and can it detect when those conditions no longer support continuation?

If yes, the deployer is moving toward Article 26 operational readiness.

If no, the system may be deployed technologically, but not governed sufficiently.

The readiness matrix turns Article 26 from a list of obligations into a practical evidence architecture.

Chapter 26

Human Oversight Checklist

Legal anchor and framework boundary. [Article 14](#); [Article 26\(2\)](#). The checklist operationalises oversight capability and should be adapted to the system, risk and context.

26.1 Purpose of the Human Oversight Checklist

The purpose of the Human Oversight Checklist is to help deployers determine whether human oversight is real, operational and evidenced.

Human oversight is one of the most frequently repeated phrases in AI governance.

It is also one of the easiest to weaken in practice.

An organization may say that a human remains responsible.

It may add a human review step.

It may require a person to approve an AI-generated recommendation.

It may describe the workflow as human-in-the-loop.

It may assign a manager, analyst, clinician, recruiter, case worker or compliance officer to review the output.

But none of this automatically proves meaningful oversight.

A human can be present and still not be in control.

A human can review an output and still lack enough information to challenge it.

A human can have formal authority and still lack practical support.

A human can approve a recommendation and still be shaped by automation bias, time pressure, interface design or organizational expectations.

For high-risk AI systems, the central question is not:

Was a human involved?

The stronger question is:

Could the human meaningfully understand, challenge, delay, escalate, override or stop the AI-supported workflow before consequence attached?

This checklist is designed around that question.

It helps the deployer evaluate whether the oversight role has:

- competence;
- training;
- authority;
- support;
- timing;
- evidence access;
- intervention ability;
- escalation route;
- workload capacity;
- independence;
- documentation;
- monitoring.

The checklist should be applied before deployment, during pilot testing, during periodic reassessment, after incidents, after workflow changes and whenever the AI system begins influencing new decisions, people or downstream systems.

Human oversight should not be treated as a one-time role assignment.

It is a continuing control condition.

26.2 Oversight Role Identification

The first step is to identify who performs human oversight.

A deployer should avoid vague statements such as:

“The team reviews the output.”

“Management is responsible.”

“Users remain accountable.”

“Human oversight is included in the process.”

These statements may be too broad to support evidence of control.

A mature oversight model should identify specific roles and responsibilities.

The deployer should define:

- who reviews the AI output;
- who has authority to approve;
- who has authority to override;

- who has authority to escalate;
- who has authority to stop the workflow;
- who monitors oversight quality;
- who owns training;
- who owns periodic review;
- who handles unresolved disagreement;
- who preserves oversight records.

The oversight role may vary by workflow.

For example:

In recruitment, the reviewer may be a trained HR professional or hiring manager.

In procurement, the reviewer may be a procurement officer, risk owner or approval committee.

In healthcare, the reviewer may be a clinician with relevant responsibility.

In financial services, the reviewer may be a compliance analyst, risk officer or regulated function holder.

In public services, the reviewer may be a case worker, supervisor or legally authorized official.

In technical operations, the reviewer may be a system owner, security engineer or change-control approver.

The deployer should avoid assigning oversight to someone who is too distant from the workflow.

A senior person may have authority but lack context.

A technical person may understand the system but lack domain authority.

A domain expert may understand the decision but lack access to system evidence.

A front-line user may see the output but lack power to challenge it.

Good oversight often requires a combination of domain knowledge, system understanding and organizational authority.

The checklist should therefore ask:

- Is the oversight role named clearly?
- Is the role connected to the specific AI workflow?
- Is the reviewer a natural person, not only a team or function?
- Is backup coverage defined?

- Is responsibility clear when the main reviewer is unavailable?
- Is the reviewer independent enough to challenge the output?
- Is the reviewer close enough to the workflow to understand consequence?
- Is there a second-level reviewer for high-consequence cases?
- Is there an escalation owner?
- Is the oversight assignment documented and current?

If the deployer cannot answer these questions, human oversight may be too vague to rely on as a control.

26.3 Competence Checklist

Competence means the human overseer has the ability to understand the system's role, the domain context, the output, the limitations and the consequence of action.

Competence is not generic intelligence.

A person may be skilled and experienced but still not competent for a specific oversight task.

For example:

A manager may understand the business process but not the AI system's limitations.

A technical specialist may understand the model but not the legal or clinical context.

A compliance officer may understand regulation but not the operational workflow.

A clinician may understand medicine but not the system's validation limits.

A recruiter may understand hiring but not how ranking algorithms can shape opportunity.

A case worker may understand the public-service process but not how AI summarization may omit critical evidence.

The deployer should define competence specifically.

The checklist should ask:

- Does the overseer understand the domain?
- Does the overseer understand the workflow affected by the AI system?
- Does the overseer understand the intended purpose of the system?
- Does the overseer understand what the system is not intended to do?
- Does the overseer understand the output type?

- Does the overseer understand known limitations?
- Does the overseer understand possible failure modes?
- Does the overseer understand the input data requirements?
- Does the overseer understand when evidence may be stale or incomplete?
- Does the overseer understand the consequence level of the workflow?
- Does the overseer understand when escalation is required?
- Does the overseer understand how to override or refuse continuation?
- Does the overseer understand automation bias?
- Does the overseer understand affected-person implications?
- Does the overseer understand what must be documented?

Competence should be evidenced.

Possible evidence includes:

- role description;
- qualification record;
- domain experience;
- system-specific training;
- oversight procedure acknowledgement;
- practical assessment;
- supervised review period;
- periodic refresher training;
- review quality audits;
- escalation participation records.

The deployer should not assume competence because someone holds a senior title.

Competence must match the oversight task.

The key evidence question is:

Can the organization show why this person was competent to oversee this AI-supported workflow?

26.4 Training Checklist

Training turns general competence into system-specific readiness.

A person may understand the domain but still need training on the specific AI system.

Training should not be limited to generic AI awareness.

For high-risk AI systems, training should cover the actual workflow, the provider's instructions, the system's limitations, the output interpretation, escalation triggers and the reviewer's authority.

The checklist should ask whether training covers:

- System identity and provider.
- Intended purpose.
- Approved use case.
- Prohibited uses.
- Output type.
- Known limitations.
- Input data requirements.
- Data freshness and missing-data warnings.
- Human oversight role.
- Automation bias.
- Evidence review requirements.
- Override procedure.
- Escalation procedure.
- HOLD and REFUSE conditions.
- Logging requirements.
- Incident reporting.
- Affected-person or worker information duties where relevant.
- Sector-specific risks.
- Examples of correct and incorrect use.
- Practical review exercises.

Training should be role-specific.

A front-line reviewer needs practical workflow guidance.

A supervisor needs escalation and override guidance.

A compliance owner needs evidence and audit guidance.

A technical owner needs monitoring and log-integrity guidance.

A senior manager needs risk, authority and accountability guidance.

Training should also be refreshed when conditions change.

Refresh training may be required when:

- provider instructions change;

- model version changes;
- workflow changes;
- new data sources are added;
- output becomes more influential;
- downstream integrations change;
- incidents occur;
- monitoring reveals weak oversight;
- legal or internal policy expectations change;
- new user groups are introduced;
- model fallback rules change.

Training evidence should include:

- training materials;
- attendance or completion records;
- version of training;
- date completed;
- role covered;
- assessment result where used;
- refresher schedule;
- practical simulation records;
- acknowledgement of responsibilities;
- remediation for failed or incomplete training.

The key evidence question is:

Can the organization show that the human overseer was trained for this system, this workflow and this consequence level?

If not, human oversight may be under-supported.

26.5 Authority Checklist

Authority means the human overseer can act.

A person may be competent and trained, but if they cannot challenge, override, delay, escalate or stop the workflow, oversight is weak.

Authority must be practical, not only formal.

A policy may say that the reviewer can override the AI output.

But if the interface does not allow override, authority is weak.

A policy may say that escalation is available.

But if escalation is slow, unclear or discouraged, authority is weak.

A policy may say that humans remain responsible.

But if the workflow automatically proceeds unless the reviewer acts instantly, authority is weak.

The checklist should ask:

Can the reviewer approve the AI-supported output?

Can the reviewer reject the AI-supported output?

Can the reviewer modify the output?

Can the reviewer request additional evidence?

Can the reviewer delay the workflow?

Can the reviewer escalate uncertainty?

Can the reviewer override the AI recommendation?

Can the reviewer stop the workflow before consequence attaches?

Can the reviewer refuse continuation?

Can the reviewer record disagreement?

Can the reviewer trigger second review?

Can the reviewer challenge input data quality?

Can the reviewer challenge use outside intended purpose?

Can the reviewer block downstream automation?

Can the reviewer contact a responsible owner or provider channel where needed?

Authority should be mapped to consequence.

A reviewer may have authority for low-consequence outputs but not high-consequence decisions.

A procurement officer may review supplier summaries but not approve high-value contracts.

A recruiter may review candidate summaries but not make final hiring decisions.

A compliance analyst may review draft summaries but not close regulated cases.

A clinician may use a summary but must apply independent professional judgment before care decisions.

A public-sector case worker may review a recommendation but require supervisor approval for certain outcomes.

The deployer should define authority thresholds.

These may include:

- financial value thresholds;
- risk category thresholds;
- affected-person impact;
- safety impact;
- legal or regulatory consequence;
- public-sector consequence;
- employment consequence;
- clinical consequence;
- downstream automation effect.

Authority evidence may include:

- approval matrix;
- delegation records;
- role permissions;
- workflow configuration;
- reviewer authority status;
- override logs;
- escalation logs;
- refusal records;
- decision records;
- audit trail.

The key evidence question is:

Could the human reviewer exercise authority at the moment it mattered?

If not, human oversight may not be meaningful.

26.6 Support Checklist

Support means the human overseer has the tools, information, time, interface design and organizational backing needed to perform oversight.

Support is often the missing element.

A person may be competent, trained and formally authorized, but still unable to perform meaningful oversight if they lack support.

The checklist should ask:

Does the reviewer have enough time to review?
 Does the reviewer have access to source evidence?
 Does the reviewer see input data freshness?
 Does the reviewer see missing-data warnings?
 Does the reviewer see system limitations?
 Does the reviewer see policy or rule context?
 Does the reviewer see the AI output and supporting evidence together?
 Does the interface allow rejection, modification, escalation and hold?
 Is approval easier than challenge?
 Is uncertainty visible?
 Are conflicting signals visible?
 Are downstream consequences visible?
 Is the reviewer protected from pressure to rubber-stamp?
 Is escalation supported by management?
 Is there backup support when the reviewer is unavailable?
 Are technical support channels available?
 Are domain experts available where needed?
 Are legal, compliance or safety teams available where needed?
 Does the reviewer receive feedback after escalation?
 Are oversight workloads monitored?

Support should be tested under realistic conditions.

A workflow may look good in a policy document but fail under volume pressure.

The deployer should test:

normal workload;
 peak workload;
 missing-data scenarios;
 stale evidence scenarios;
 model fallback scenarios;
 conflicting evidence scenarios;
 use outside intended purpose;
 human reviewer unavailable;
 escalation required;
 downstream automation risk.

The organization should preserve evidence of support.

Possible evidence includes:

workflow screenshots;
interface design records;
reviewer guidance;
evidence-access logs;
escalation contact list;
support procedures;
workload reports;
time-per-review metrics;
reviewer feedback;
oversight quality audits;
management communications supporting escalation.

The key evidence question is:

Did the organization give the human overseer enough support to exercise control?

If not, oversight may exist on paper but fail in operation.

26.7 Timing Checklist: Before Consequence Attaches

Human oversight is strongest before consequence attaches.

A review after the outcome may support audit, appeal or correction, but it is not the same as preventive control.

The deployer should identify consequence points.

A consequence point is a workflow stage where the AI-supported output begins to affect a person, process, right, opportunity, obligation, operational action or regulated decision.

Examples include:

candidate shortlisted or rejected;
worker flagged for performance concern;
supplier recommended for approval;
clinical pathway influenced;
customer response sent;
public-service case routed;
fraud flag acted upon;
compliance case closed;
access permission changed;

automated task triggered;
 external communication sent;
 official record updated;
 financial approval routed;
 legal or administrative decision drafted.

The checklist should ask:

Has the consequence point been identified?
 Does human review occur before that point?
 Is the reviewer aware of the consequence level?
 Can the reviewer stop the workflow before consequence?
 Can the reviewer request more evidence before consequence?
 Can the reviewer escalate before consequence?
 Can downstream automation proceed without review?
 Can external communication occur before review?
 Can an AI-generated recommendation become a trigger before review?
 Is post-consequence review being mistaken for oversight?

Timing should be evidenced.

The log should show:

AI output timestamp;
 review timestamp;
 reviewer action;
 consequence timestamp;
 whether review occurred before consequence;
 whether escalation occurred before consequence;
 whether output was modified before consequence;
 whether workflow was held or refused.

The key evidence question is:

Did human oversight happen early enough to change what happened?

If not, the oversight may be retrospective rather than preventive.

26.8 Evidence Access Checklist

Human oversight depends on evidence access.

A reviewer cannot meaningfully challenge an AI output if they cannot see the evidence behind it.

The checklist should ask:

Can the reviewer see the source documents or records?

Can the reviewer see which data sources were used?

Can the reviewer see which data sources were missing?

Can the reviewer see freshness status?

Can the reviewer see policy version?

Can the reviewer see model or system version where relevant?

Can the reviewer see warnings and limitations?

Can the reviewer see uncertainty indicators?

Can the reviewer see why the AI system produced the output?

Can the reviewer see what the AI system did not consider?

Can the reviewer access alternative evidence?

Can the reviewer request additional information?

Can the reviewer compare the AI output with source evidence?

Can the reviewer see prior human interventions?

Can the reviewer see downstream effect?

The reviewer should not be forced to approve a black-box output.

For high-risk AI systems, a review screen should ideally show:

AI output;

supporting evidence;

input data condition;

missing evidence;

freshness status;

limitations;

required human action;

available decisions;

escalation route;

downstream consequence.

Evidence access should be proportionate to risk.

A low-risk internal draft may not require deep source review.

A high-risk decision-support output should.

The key evidence question is:

Could the reviewer inspect enough evidence to make an informed judgment?

If not, the reviewer may be approving the appearance of reasoning rather than the reasoning itself.

26.9 Automation Bias Checklist

Automation bias occurs when humans over-rely on automated outputs.

It is one of the most important risks in human oversight.

AI systems can appear objective, consistent and confident.

This can cause reviewers to trust outputs too easily.

Automation bias may increase when:

- outputs are fluent;
- scores appear precise;
- recommendations look professional;
- users are under time pressure;
- the AI system is usually correct;
- management values speed;
- override requires extra work;
- the interface encourages approval;
- reviewers lack source evidence;
- reviewers believe the system is more objective than humans;
- disagreement is culturally discouraged.

The checklist should ask:

- Are reviewers trained on automation bias?
- Are reviewers told that AI outputs may be wrong or incomplete?
- Does the interface show uncertainty?
- Does the interface show limitations?
- Does the interface make challenge practical?
- Are override options visible?
- Are reviewers encouraged to disagree where appropriate?
- Are overrides treated as useful signals rather than failures?
- Are review times monitored?
- Are approval rates monitored?
- Are override rates monitored?

Are escalation rates monitored?
 Are sudden drops in escalation investigated?
 Are reviewers asked for feedback?
 Are cases sampled for review quality?
 Are managers trained not to pressure reviewers into rubber-stamping?
 Are AI outputs prevented from becoming default decisions without review?
 Are high-confidence outputs still reviewable?
 Are human decisions compared with AI recommendations?
 Are signs of overreliance escalated?

Automation bias should be monitored.

Signals may include:

very high approval rates;
 very short review times;
 low override rates;
 low escalation rates;
 repeated acceptance of AI output;
 reviewer comments becoming minimal;
 output becoming official reasoning without modification;
 users skipping source evidence;
 increasing dependence on system ranking or scoring;
 human disagreement disappearing after deployment.

The key evidence question is:

Is the human reviewer exercising judgment, or simply legitimizing the AI output?

If the answer is unclear, oversight quality should be reviewed.

26.10 Workload and Capacity Checklist

Human oversight can fail when reviewers are overloaded.

A person may be competent and willing to review carefully, but volume can make meaningful review impossible.

The checklist should ask:

How many AI-supported outputs does each reviewer handle?

How much time is needed for meaningful review?

How much time is actually available?

Are review queues monitored?

Are high-risk cases prioritized?

Are reviewers pressured by throughput targets?

Are review backlogs visible?

Are reviewers able to pause or escalate when overloaded?

Is backup coverage available?

Are peak periods managed?

Are review times too short?

Are complex cases allocated more time?

Are reviewers trained to refuse superficial review?

Are managers monitoring quality, not only speed?

Are human oversight resources proportionate to deployment scale?

Capacity should match consequence.

If the AI system processes thousands of high-consequence outputs, the organization must ensure that human oversight capacity is realistic.

A token human review step is not enough.

The deployer should preserve evidence of workload and capacity, including:

reviewer caseload;

review time metrics;

backlog reports;

staffing model;

escalation volume;

overtime or pressure indicators;

quality review results;

reviewer feedback;

incidents linked to overload.

The key evidence question is:

Was the human oversight function realistically resourced?

If not, human oversight may degrade under operational pressure.

26.11 Independence and Challenge Checklist

Human oversight requires the ability to challenge.

The reviewer should not be structurally pressured to approve.

The checklist should ask:

Can the reviewer disagree without penalty?

Are reviewers expected to meet approval-speed targets?

Are reviewers rewarded for throughput over quality?

Are overrides investigated fairly?

Are escalations treated as control signals?

Are reviewers protected from retaliation or blame?

Can reviewers raise concerns about the AI system itself?

Can reviewers challenge management expectations?

Is there a route for confidential reporting where needed?

Is second-level review available?

Can domain experts be consulted?

Are reviewer disagreements recorded?

Are repeated disagreements analyzed?

Are human concerns used to improve the system?

Does organizational culture support intervention?

Challenge is especially important in high-pressure sectors.

In employment, recruiters may feel pressure to trust rankings.

In finance, analysts may feel pressure to clear cases quickly.

In healthcare, clinicians may face time pressure.

In public services, case workers may face backlog pressure.

In procurement, reviewers may face business pressure to approve suppliers.

In all such settings, human oversight must be protected from organizational incentives that reduce challenge.

The key evidence question is:

Was the reviewer free and supported to disagree with the AI system?

If not, oversight may be compromised.

26.12 Documentation Checklist

Human oversight must be documented.

Not every review requires a long written explanation.

But the record should be proportionate to consequence and sufficient for reconstruction.

The checklist should ask whether the record captures:

AI system used.

Workflow or case ID.

AI output.

Input data condition.

Reviewer identity or role.

Reviewer authority.

Review timestamp.

Evidence available to reviewer.

Warnings or limitations shown.

Reviewer action.

Approval, modification, rejection, escalation or refusal.

Reason for override or escalation where relevant.

Downstream consequence.

Final decision.

Whether review occurred before consequence.

Whether additional evidence was requested.

Whether the affected person provided context where relevant.

Whether logs were complete.

Whether monitoring alerts were active.

Whether the workflow remained within intended purpose.

The documentation should support a decision trail.

It should help answer:

What did the human reviewer see?

What could they do?

What did they decide?

Why was continuation allowed?

Documentation should avoid two extremes.

The first extreme is no meaningful record.

This weakens accountability.

The second extreme is excessive documentation that makes oversight impractical.

The right level depends on risk.

For high-consequence workflows, stronger documentation is needed.

For lower-risk support tasks, lighter records may be sufficient.

The key evidence question is:

Can the organization reconstruct the human oversight step and its effect on the outcome?

If not, oversight evidence is weak.

26.13 Oversight Quality Monitoring

Human oversight should itself be monitored.

It is not enough to assign a reviewer and assume the control remains effective.

Oversight can degrade.

Reviewers can become overloaded.

Automation bias can increase.

Override rates can fall.

Escalations can disappear.

Use can drift.

The AI output can become more influential.

The workflow can change.

The checklist should ask:

Are review rates monitored?

Are approval rates monitored?

Are override rates monitored?

Are escalation rates monitored?

Are review times monitored?

Are reviewer comments sampled?

Are high-risk cases quality-checked?

Are reviewer disagreements analyzed?

Are near misses reviewed?

Are cases where AI output was wrong reviewed?

Are cases where humans overrode AI reviewed?

Are cases where humans accepted AI incorrectly reviewed?

Are affected-person complaints linked to AI review?

Are reviewer feedback sessions held?

Are training needs updated?

Are interface issues reported?

Are workload issues escalated?

Are oversight failures treated as governance incidents?

Are recurring issues remediated?

Is oversight effectiveness reported to governance bodies?

Useful oversight quality indicators include:

time per review;

percentage of outputs approved without modification;

override frequency;

escalation frequency;

refusal frequency;

reviewer confidence;

reviewer uncertainty reports;

source evidence access frequency;

post-decision appeal outcomes;

incident correlation;

quality audit findings.

The goal is not to punish reviewers.

The goal is to detect whether the oversight function remains real.

The key evidence question is:

Can the organization show that human oversight remained effective over time?

If not, oversight may be degrading silently.

26.14 Human Oversight Decision Table

The following table provides a practical oversight assessment structure.

Oversight Element	Core Question	Evidence Needed	Risk Signal
Role	Who oversees the AI output?	Named role, responsibility record	Oversight assigned only to team
Competence	Can the reviewer understand the task?	Qualifications, domain experience	Reviewer lacks domain knowledge
Training	Has the reviewer been trained for this system?	Training records, materials	Generic AI training only
Authority	Can the reviewer act?	Approval matrix, override logs	Reviewer can only approve
Support	Does the reviewer have tools and time?	Interface records, workload data	High volume, low review time
Evidence Access	Can the reviewer see supporting evidence?	Source access logs, review screen	Final score only
Timing	Is review before consequence?	Timestamps, workflow records	Review after action
Challenge	Can reviewer disagree safely?	Escalation policy, culture evidence	Overrides discouraged
Escalation	Can uncertainty move upward?	Escalation logs, contact matrix	Escalation unclear
Documentation	Is review reconstructable?	Decision trail, review record	Approval click only
Monitoring	Is oversight quality checked?	Review metrics, audits	Override rate collapses
Reassessment	Is oversight updated when use changes?	Review triggers, updates	Use expands without retraining

This table can be used as a quick diagnostic.

If several risk signals appear, human oversight should not be treated as mature.

26.15 Minimum Human Oversight Evidence Pack

For high-risk AI systems, the deployer should maintain a minimum human oversight evidence pack.

This pack should include:

- Oversight role description.
- Named responsible role or person.
- Backup coverage.
- Competence criteria.
- Training materials.
- Training completion records.
- Authority matrix.
- Workflow review procedure.
- Evidence access description.
- Review interface description.
- Override procedure.
- Escalation procedure.
- HOLD and REFUSE conditions.
- Review documentation template.
- Sample decision trail.
- Oversight quality monitoring metrics.
- Reviewer feedback records.
- Automation bias training.
- Reassessment triggers.
- Remediation records.

The evidence pack should answer:

Who oversees?

Why are they competent?

What training did they receive?

What authority do they have?

What support do they receive?

Can they intervene?

Can they stop the workflow?

Is review before consequence?

Is review documented?

Is oversight monitored?

This pack should be updated when:

the system changes;

the workflow changes;

the reviewer changes;

the consequence level changes;

output becomes more influential;

downstream automation is added;

incidents occur;

monitoring shows oversight degradation;

provider instructions change;

model fallback rules change.

Human oversight readiness is not permanent.

It must remain current.

Chapter 26 Summary

Human oversight is not the same as human presence.

A person can be in the loop without being in control.

For Article 26 deployer governance, human oversight must be operational, meaningful and evidenced.

This chapter provides a practical Human Oversight Checklist built around competence, training, authority and support.

The deployer should identify who performs oversight, whether they understand the system and domain, whether they have system-specific training, whether they have authority to act, whether they receive enough support, whether they can access evidence, whether review occurs before consequence, and whether their intervention is documented.

The chapter emphasizes that human oversight is strongest before consequence attaches.

Post-consequence review may support correction, appeal or audit, but pre-consequence review is the stronger control.

The chapter also addresses automation bias, workload, independence, documentation and oversight quality monitoring.

A human reviewer may be competent and authorized, but still become ineffective under time pressure, interface pressure, excessive volume or organizational incentives that discourage challenge.

For this reason, oversight itself must be monitored.

The chapter proposes a Human Oversight Decision Table and a Minimum Human Oversight Evidence Pack.

The central question is:

Could the human meaningfully understand, challenge, delay, escalate, override or stop the AI-supported workflow before consequence attached?

If yes, human oversight may function as a real control.

If no, it may be only a procedural appearance.

In high-risk AI governance, that difference matters.

Chapter 27

Evidence of Control Template

Legal anchor and framework boundary. [Article 12](#); [Article 26\(6\)](#). The evidence template exceeds minimum log-retention language where proportionate; it is a governance aid, not a mandated form.

27.1 Purpose of the Template

The purpose of the Evidence of Control Template is to help deployers document, preserve and reconstruct the control conditions under which a high-risk AI system is used.

This template is designed for practical use.

It does not replace legal advice.

It does not replace sector-specific compliance requirements.

It does not replace provider documentation.

It does not replace internal risk management frameworks.

Its purpose is to help an organization answer one central question:

Can we prove that this AI-supported workflow was allowed to continue under valid control conditions?

This question is the heart of deployer governance.

A deployer may have many documents:

- AI policy;
- system register;
- procurement file;
- vendor contract;
- risk assessment;
- training records;
- incident process;
- data protection documentation;
- audit logs;
- monitoring dashboard;
- human review procedure.

All of these may be useful.

But they may still be fragmented.

The organization may have records, but not a connected evidence chain.

The Evidence of Control Template connects those records into a practical structure.

It helps preserve evidence around:

- system identity;
- intended purpose;
- actual use;
- provider instructions;
- input data condition;
- human oversight;
- authority state;
- monitoring signals;
- logs;
- escalation;
- downstream consequence;
- decision to continue, hold, escalate or refuse.

The template should be used at two levels.

First, the system level.

This records the general governance state of the AI system.

Second, the workflow or case level.

This records the specific evidence of control for a particular AI-supported workflow, output, recommendation, decision or consequence.

Both levels matter.

System-level evidence shows that the organization has governance controls.

Case-level evidence shows that those controls operated when it mattered.

A mature deployer needs both.

27.2 Evidence of Control Header

Every Evidence of Control record should begin with a clear header.

The header should identify the system, workflow and record owner.

Template Section 1 — Record Header

Evidence Record ID:

Unique internal reference.

Date Created:

Date and time the evidence record was created.

Record Owner:

Person or role responsible for maintaining the evidence record.

Business Owner:

Business function responsible for the AI-supported process.

Technical Owner:

Technical function responsible for system operation or integration.

Governance Owner:

Risk, compliance, legal, AI governance or accountable function responsible for oversight.

AI System Name:

Name of the AI system.

Provider:

Provider or vendor of the AI system.

System Version / Model Version:

Version used, where applicable.

Deployment Environment:

Production, pilot, sandbox, internal test, limited deployment or other environment.

Workflow / Process Name:

The business, clinical, employment, procurement, public-service, compliance or operational workflow affected.

Use Case:

Short description of how the AI system is used.

Record Type:

System-level evidence record, workflow-level evidence record, case-level evidence record, incident evidence record or reassessment record.

Risk Classification:

Internal risk classification and high-risk status where applicable.

Consequence Level:

Low, moderate, high or critical consequence.

Affected Persons or Groups:

Workers, customers, patients, candidates, citizens, students, suppliers, clients or other affected groups.

Next Review Date:

Date for reassessment.

This header matters because evidence without identity is weak.

A log, record or review note is useful only if the organization can connect it to the system, workflow, owner and consequence.

The header creates that connection.

27.3 System Identity and Intended Purpose

The next section records what the system is and what it is allowed to do.

Template Section 2 — System Identity and Intended Purpose

System Description:

Describe the AI system in practical terms.

Provider-Defined Intended Purpose:

Summarize the intended purpose as described by the provider.

Internal Approved Use:

Describe how the organization has approved the system for use.

Actual Use:

Describe how the system is actually used in practice.

Prohibited Uses:

List uses that are not allowed.

Approved User Groups:

Identify who may use the system.

Approved Affected Processes:

Identify the workflows where the system may be used.

Output Type:

Classification, score, ranking, summary, recommendation, draft, alert, decision support, tool action, workflow trigger or other output.

Output Authority Level:

Draft-only, advisory, decision support, controlled operational trigger, or not authorized for consequence.

Downstream Systems:

Identify systems that receive or act on the AI output.

Scope Limits:

Describe limits by process, user group, data source, jurisdiction, consequence level or decision type.

Known Limitations:

Record limitations from provider instructions, internal testing, monitoring or incidents.

Reassessment Triggers:

List conditions that require reassessment.

This section helps answer:

Was the system used for something it was allowed to do?

The deployer should compare intended purpose, internal approved use and actual use.

If actual use differs from approved use, the evidence record should show whether reassessment occurred.

This prevents silent scope drift.

27.4 Instructions for Use Mapping

Provider instructions should be translated into deployer controls.

The template should not only store the instructions.

It should show how the instructions were operationalized.

Template Section 3 — Instructions for Use Mapping

Provider Instruction / Limitation	Internal Control	Evidence Produced	Owner	Status
Example: human review required	Workflow requires reviewer	Review timestamp, reviewer ID,	Process owner	Active

Provider Instruction / Limitation	Internal Control	Evidence Produced	Owner	Status
before final decision	approval before final state	decision log		
Example: only use with current source data	Freshness check before output can be used	Data freshness record	Data owner	Active
Example: not for final diagnosis	Output marked decision support only	Interface warning, clinician review record	Clinical owner	Active

The deployer should complete this table for each important provider instruction or limitation.

Important instruction categories may include:

- intended purpose;
- prohibited use;
- required user competence;
- input data requirements;
- limitations;
- human oversight requirements;
- output interpretation;
- monitoring recommendations;
- logging capabilities;
- incident reporting routes;
- update procedures;
- fallback or availability limitations.

This mapping is important because provider instructions do not implement themselves.

The deployer must show how documentation became operational control.

The key evidence question is:

Can we show that the provider's instructions were translated into actual workflow controls?

27.5 Input Data Condition Record

Input data is a central control condition.

The Evidence of Control Template should preserve the state of input data at the relevant point of use.

Template Section 4 — Input Data Condition

Input Data Sources:

List each source used.

Source Owner:

Owner of each source.

Source Type:

Internal system, external database, user-entered data, provider data, third-party data, document repository, real-time signal or other.

Source Reliability:

Authoritative, verified, user-provided, unverified, draft, historical, superseded or unknown.

Last Updated:

Timestamp of source update.

Retrieved At:

Timestamp when the AI system retrieved or received the input.

Freshness Threshold:

Maximum acceptable age or required event-based condition.

Freshness Status:

Current, near threshold, stale, unknown or not applicable.

Completeness Status:

Complete, incomplete, missing required fields, unknown or not applicable.

Representativeness Status:

Sufficient, limited, unknown, not assessed or not applicable.

Known Data Limitations:

Describe missing fields, stale sources, known bias, restricted scope, incomplete records or source conflicts.

User-Entered Context:

Record whether user-provided information was included.

Verified Against Source:

Yes, no, partial or not applicable.

Warnings Displayed:

List warnings shown to user or reviewer.

Input Data Decision:

ALLOW, HOLD, ESCALATE or REFUSE.

Reason:

Explain why input data condition supported or did not support continuation.

This section helps answer:

Was the AI system given data that could support a valid output for this use?

For high-consequence workflows, input data condition should be recorded before consequence attaches.

If the data was stale, incomplete or uncertain, the record should show whether the workflow held, escalated or refused continuation.

27.6 Context Record for LLM and Agentic Systems

For large language model and agentic systems, the active context matters.

The organization should preserve enough context evidence to reconstruct what the system relied on.

Template Section 5 — Context Record

Context Bundle ID:

Unique reference for the context supplied to the system.

Prompt / Instruction Type:

User prompt, system instruction, workflow instruction, agent plan, tool instruction or other.

Context Sources Included:

List documents, records, databases, policies, tool outputs or summaries included.

Context Sources Excluded:

Where relevant, list important sources not included.

Policy Version Included:

Active policy or rule version included in context.

Provider Instruction Included:

Yes, no, partial or not applicable.

Freshness Metadata Included:

Yes, no or partial.

Limitations Included:

Yes, no or partial.

Authority State Included:

Yes, no or partial.

Human Oversight Requirement Included:

Yes, no or partial.

Context Compaction Used:

Yes or no.

Compaction Method:

Summary, memory, retrieval compression, conversation summary or other.

Critical Information Preserved:

Source references, warnings, limitations, uncertainty, freshness state, authority state, pending checks.

Critical Information Lost or Unknown:

Describe any missing or potentially lost information.

Context Decision:

ALLOW, HOLD, ESCALATE or REFUSE.

This section helps answer:

Did the system receive the right context for the consequence attached to its output?

For high-risk workflows, context should not be treated as temporary prompt material.

It is part of the decision trail.

27.7 AI Output Record

The output record preserves what the AI system produced and how it was classified.

Template Section 6 — AI Output Record

Output Timestamp:

Time output was generated.

Output Type:

Summary, score, classification, recommendation, ranking, draft, alert, action plan, tool call, decision support or other.

Output Content:

Store or reference the output.

Confidence / Uncertainty Indicator:

Where available.

Sources Cited:

List source references used or cited.

Known Limitations Displayed:

Warnings, scope limits, missing evidence or uncertainty shown.

Contradictions Detected:

Yes, no, unknown or not applicable.

Unsupported Assumptions Detected:

Yes, no, unknown or not applicable.

Output Authority Level:

Draft-only, advisory, decision support, controlled trigger, not authorized.

Output May Influence:

Internal review, human decision, downstream task, external communication, regulated process, affected-person outcome or other.

Output Decision:

ALLOW, HOLD, ESCALATE or REFUSE.

Reason:

Explain why the output may or may not proceed.

This section helps avoid the mistake of treating all outputs as equivalent.

A draft summary should not carry the same authority as a recommendation.

A recommendation should not automatically become an executable trigger.

The output record should preserve that distinction.

27.8 Human Oversight Record

The human oversight record is one of the most important sections.

It should prove that human oversight was meaningful, not merely present.

Template Section 7 — Human Oversight Record

Was Human Oversight Required?

Yes, no or conditional.

Reason Oversight Was Required:

Consequence level, policy, provider instruction, legal requirement, internal rule, escalation trigger or other.

Reviewer Name or Role:

Natural person or accountable role.

Reviewer Competence Confirmed:

Yes, no or partial.

Reviewer Training Confirmed:

Yes, no or partial.

Reviewer Authority Confirmed:

Yes, no or partial.

Evidence Available to Reviewer:

Source records, AI output, input data condition, warnings, policy version, authority state, downstream consequence.

Review Timestamp:

Date and time.

Review Occurred Before Consequence:

Yes or no.

Reviewer Action:

Approve, modify, reject, request more evidence, hold, escalate, override, refuse.

Reviewer Comments:

Short reason where relevant.

Override or Disagreement Recorded:

Yes, no or not applicable.

Escalation Triggered:

Yes or no.

Final Oversight State:

Satisfied, incomplete, failed, escalated, refused.

Oversight Decision:

ALLOW, HOLD, ESCALATE or REFUSE.

This section answers the core question:

Could the human meaningfully change the outcome before consequence attached?

If the reviewer could only click approve without evidence access, the record should not treat oversight as mature.

27.9 Authority and Scope Record

The authority and scope record shows whether the workflow was allowed to continue.

Template Section 8 — Authority and Scope

Workflow Scope:

Describe the approved scope.

Current Use Within Approved Scope?

Yes, no, partial or unknown.

Authority Holder:

Person, role, committee, system rule or policy authority.

Authority Type:

Approval, override, escalation, execution, review, suspension or refusal.

Authority Valid At Time of Use?

Yes, no, partial or unknown.

Authority Evidence:

Approval matrix, delegation record, role permission, policy version, system permission, workflow rule or other.

Consequence Level:

Low, moderate, high or critical.

Thresholds Applied:

Budget, risk, clinical, employment, public-service, legal, safety or other thresholds.

Downstream Effect:

None, internal task, official record, external communication, operational action, affected-person outcome or regulated decision.

Boundary Condition:

Within boundary, near boundary, boundary crossed, unknown.

Authority Decision:

ALLOW, HOLD, ESCALATE or REFUSE.

Reason:

Explain why authority supported or did not support continuation.

This section helps answer:

Was the workflow still authorized to continue under current conditions?

If authority is unclear, the correct state should often be HOLD or ESCALATE, not silent continuation.

27.10 Monitoring and Alert Record

The monitoring and alert record preserves relevant runtime signals.

Template Section 9 — Monitoring and Alerts

Monitoring Active?

Yes, no or partial.

Signals Checked:

Performance, input data, context, human oversight, authority, workflow drift, model fallback, downstream integration, log integrity, incident signals.

Active Alerts:

List alerts active at the time of use.

Relevant Drift Signals:

Model drift, operational drift, context drift, authority drift, scope drift or none.

Data Quality Alerts:

Stale data, missing data, source failure, representativeness concern or none.

Human Oversight Alerts:

Reviewer unavailable, review overdue, high approval rate, low override rate, workload pressure or none.

Authority Alerts:

Approver mismatch, threshold exceeded, policy mismatch, delegation expired, fallback authority reduction or none.

Log Integrity Alerts:

Missing logs, incomplete record, broken decision trail or none.

Monitoring Decision:

ALLOW, HOLD, ESCALATE or REFUSE.

Reason:

Explain action taken.

This section helps prove that monitoring was connected to action.

Monitoring is weak if alerts are visible but do not affect continuation.

27.11 Escalation and Incident Record

When conditions are uncertain, risky or invalid, escalation may be required.

Template Section 10 — Escalation and Incident

Escalation Required?

Yes, no or uncertain.

Escalation Trigger:

Stale evidence, missing oversight, serious risk signal, incident, policy conflict, use outside scope, authority issue, affected-person complaint, model fallback, data source failure or other.

Escalation Recipient:

Human overseer, manager, AI governance owner, legal, compliance, clinical safety, HR, procurement, risk, provider, authority or other.

Escalation Timestamp:

Date and time.

Use Suspended Pending Review?

Yes or no.

Evidence Preserved?

Yes, no or partial.

Provider Contacted?

Yes, no or not applicable.

Authority Contacted?

Yes, no or not applicable.

Affected Cases Identified?

Yes, no or not applicable.

Corrective Action:

Describe action taken.

Final Escalation Outcome:

Resolved, continued with conditions, suspended, refused, incident investigation opened, provider notified or authority notified.

Escalation Decision:

ALLOW, HOLD, ESCALATE or REFUSE.

This section helps answer:

Did the organization know when monitoring required action, and did it act?

Escalation records should show that risk did not disappear into informal discussion.

27.12 Downstream Consequence Record

AI governance must track where output becomes consequence.

Template Section 11 — Downstream Consequence

Did the AI Output Influence a Downstream Process?

Yes, no or unknown.

Downstream Process:

Name the process.

Type of Consequence:

Internal prioritization, official record, external communication, workflow trigger, approval, rejection, routing, escalation, clinical pathway, employment decision, public-service outcome, financial process or other.

Affected Person or Entity:

Candidate, worker, customer, patient, citizen, student, supplier, client, investor, internal user or other.

Human Review Before Consequence?

Yes, no or not applicable.

Was Output Modified Before Use?

Yes, no or not applicable.

Was External Communication Sent?

Yes, no or not applicable.

Was System Record Updated?

Yes, no or not applicable.

Was Workflow Triggered Automatically?

Yes, no or not applicable.

Was Consequence Reversible?

Yes, no, partial or unknown.

Consequence Timestamp:

Date and time.

Consequence Decision:

ALLOW, HOLD, ESCALATE or REFUSE.

This section is essential for proving whether AI output remained advisory or became operational.

The key question is:

When did the AI output begin to matter?

27.13 Final Control Decision

Every Evidence of Control record should end with a clear control decision.

Template Section 12 — Final Control Decision

Final State:

ALLOW, HOLD, ESCALATE or REFUSE.

Decision Reason:

Brief explanation.

Control Conditions Satisfied:

List satisfied conditions.

Control Conditions Missing or Weak:

List missing, weak or uncertain conditions.

Conditions for Continuation:

If continued with conditions, list them.

Remediation Required:

Yes or no.

Remediation Owner:

Person or role.

Remediation Due Date:

Date.

Next Review Trigger:

Time-based or event-based trigger.

Approver of Final Control Decision:

Person or role.

Date and Time:

Timestamp.

This final decision is important because it prevents evidence records from becoming passive documentation.

The record should support action.

A workflow either continues, holds, escalates or refuses.

The organization should know which state applies and why.

27.14 Evidence of Control Case Template

The following is a compact case-level template.

Case-Level Evidence of Control Template

1. **Case ID:**
2. **AI System:**
3. **Workflow:**
4. **Use Case:**
5. **Output Type:**
6. **Consequence Level:**
7. **Intended Purpose Match:** Yes / No / Partial / Unknown
8. **Provider Instructions Followed:** Yes / No / Partial / Unknown
9. **Input Data Freshness:** Current / Stale / Unknown / Not Applicable
10. **Input Data Completeness:** Complete / Incomplete / Unknown

11. **Context Inspectable:** Yes / No / Partial
12. **AI Output Stored:** Yes / No
13. **Human Oversight Required:** Yes / No / Conditional
14. **Human Oversight Completed Before Consequence:** Yes / No / Not Applicable
15. **Reviewer Had Authority:** Yes / No / Partial / Unknown
16. **Monitoring Alerts Active:** Yes / No
17. **Escalation Required:** Yes / No / Uncertain
18. **Downstream Consequence:** None / Internal / External / Regulated / Affected Person
19. **Logs Complete Enough for Reconstruction:** Yes / No / Partial
20. **Final Governance State:** ALLOW / HOLD / ESCALATE / REFUSE
21. **Reason for Final State:**
22. **Evidence Location:**
23. **Record Owner:**
24. **Next Review Date or Trigger:**

This compact template can be used by teams that need a practical workflow record without writing a long narrative each time.

For high-consequence cases, a fuller evidence record may be needed.

27.15 Evidence Quality Assessment

Not all evidence has the same quality.

The deployer should evaluate whether evidence is strong enough.

Evidence Quality Levels

Level 0 — No Evidence

The organization cannot show that the control existed or operated.

Level 1 — Assertion

The organization states that the control exists, but cannot show operational evidence.

Level 2 — Documented Process

A policy, procedure or training record exists, but case-level evidence is weak.

Level 3 — Operational Evidence

The control operated in the workflow and produced records.

Level 4 — Reconstructable Evidence

The organization can reconstruct the system, input, output, oversight, authority, monitoring and consequence.

Level 5 — Runtime-Assured Evidence

The control is monitored during operation and connected to ALLOW, HOLD, ESCALATE or REFUSE states.

The deployer should avoid relying on Level 1 evidence for high-risk workflows.

For high-consequence use, Level 4 or Level 5 evidence may be necessary in key areas such as human oversight, input data, authority, logging and escalation.

The key question is:

Does the evidence prove control, or merely describe intention?

27.16 Evidence Pack Structure

For each high-risk AI system, the deployer should maintain an Evidence of Control Pack.

Evidence of Control Pack Contents

- System record.
- Provider instructions.
- Intended purpose and actual use map.
- Instructions-to-controls mapping.
- Human oversight model.
- Human oversight training evidence.
- Input data source map.
- Input data validation evidence.
- Context management record where relevant.
- Output classification model.
- Authority and scope map.
- Monitoring plan.
- Alert and escalation rules.
- Log retention plan.
- Decision trail samples.
- Incident and escalation records.
- Workplace or affected-person information records.
- Third-party dependency map.
- Model dependency and fallback rules.
- Reassessment records.
- Readiness matrix result.

- Final deployment or continued-use decision.

This evidence pack should be organized so that an internal reviewer can quickly answer:

What is the system?

What is it used for?

Why is that use allowed?

Who controls it?

What data does it use?

What outputs does it produce?

Who reviews them?

What authority applies?

What happens when conditions fail?

Can a decision trail be reconstructed?

The evidence pack should not be treated as a static archive.

It should be updated when the system changes, the workflow changes, provider instructions change, use expands, incidents occur or monitoring signals drift.

27.17 Worked Example: Supplier Recommendation

The following example shows how the template may work in procurement.

Case ID: PROC-AI-2026-014

AI System: Supplier Risk Assistant

Workflow: Supplier approval

Use Case: AI-supported supplier risk summary and recommendation

Output Type: Supplier risk recommendation

Consequence Level: High

Affected Entity: Supplier and purchasing organization

Intended Purpose Match: Yes, approved for supplier risk summary and advisory recommendation

Input Data Freshness: Partial

Issue: Financial data current; sanctions check current; cybersecurity review pending

Human Oversight Required: Yes

Reviewer: Procurement risk manager

Reviewer Authority: Authority for advisory recommendation; final approval requires procurement committee

Monitoring Alerts: Cybersecurity review pending

Downstream Consequence: Supplier could be shortlisted, but not finally approved

Final Governance State: HOLD

Reason: Recommendation cannot proceed to final approval until cybersecurity review is complete

Action: Escalated to information security team

Evidence Preserved: AI output, source data, freshness status, reviewer action, escalation record

Next Trigger: Resume after cybersecurity review outcome

This example shows that the AI system did not fail.

It produced a recommendation.

But the control state was HOLD because supplier evidence was incomplete.

That is evidence of control.

27.18 Worked Example: HR Candidate Ranking

Case ID: HR-AI-2026-032

AI System: Candidate Screening Assistant

Workflow: Recruitment shortlist

Use Case: AI-supported CV summary and ranking

Output Type: Candidate ranking

Consequence Level: High

Affected Person: Job applicant

Intended Purpose Match: Partial

Issue: System approved for CV summarization, but ranking was used to determine shortlist priority

Input Data Freshness: Current

Human Oversight Required: Yes

Reviewer: HR recruiter

Reviewer Authority: Can shortlist, but must review source CVs

Monitoring Alert: Actual use drifting from summarization to ranking

Downstream Consequence: Candidate may be excluded from interview

Final Governance State: ESCALATE

Reason: Actual use exceeds approved output authority

Action: Escalated to HR governance owner; ranking use paused pending reassessment

Evidence Preserved: Output, source CVs, reviewer record, scope-drift alert, escalation record

Next Trigger: Governance decision on whether ranking use is approved with additional controls

This example shows how AI output can gain more authority than intended.

The correct response is not silent continuation.

It is escalation.

27.19 Worked Example: Clinical Summary

Case ID: CLIN-AI-2026-009

AI System: Clinical Summary Assistant

Workflow: Patient record review

Use Case: AI-generated patient summary for clinician

Output Type: Clinical summary

Consequence Level: Critical

Affected Person: Patient

Intended Purpose Match: Yes, summarization only

Input Data Freshness: Stale

Issue: Latest lab results not included

Human Oversight Required: Yes

Reviewer: Clinician

Reviewer Authority: Full clinical responsibility

Monitoring Alert: Missing latest lab data

Downstream Consequence: Summary may influence care pathway

Final Governance State: REFUSE for clinical reliance; HOLD for summary refresh

Reason: Summary cannot be treated as complete without latest lab results

Action: Retrieve updated labs and regenerate summary; clinician review required

Evidence Preserved: Original output, missing-data warning, clinician review, refresh request

Next Trigger: Updated summary after lab inclusion

This example shows that refusal can be a safety feature.

The system may still help after evidence is restored.

But the stale output should not influence care.

Chapter 27 Summary

The Evidence of Control Template is a practical tool for turning Article 26 deployer obligations into reconstructable governance evidence.

Its purpose is to help organizations answer one central question:

Can we prove that this AI-supported workflow was allowed to continue under valid control conditions?

The template covers system identity, intended purpose, instructions for use, input data condition, context, AI output, human oversight, authority, monitoring, escalation, downstream consequence and final control decision.

It supports both system-level and case-level evidence.

System-level evidence shows that governance controls exist.

Case-level evidence shows that controls operated when a specific AI-supported workflow became consequential.

The chapter emphasizes that evidence quality matters.

A policy statement is weaker than operational evidence.

Operational evidence is weaker than a reconstructable decision trail.

A reconstructable decision trail is strongest when connected to runtime ALLOW, HOLD, ESCALATE and REFUSE states.

The template also provides worked examples in procurement, HR and healthcare.

These examples show that evidence of control is not about forcing every workflow to continue.

Sometimes the strongest evidence of control is HOLD.

Sometimes it is ESCALATE.

Sometimes it is REFUSE.

The central principle is:

The organization should not only know what the AI system produced.

It should know why the output was allowed, limited, escalated or refused.

That is the practical foundation of Article 26 evidence of control.

Chapter 28

Runtime Monitoring and Incident Response Checklist

Legal anchor and framework boundary. [Article 26\(5\)](#); [Articles 72-73](#). Incident-response controls distinguish deployer notification from provider reporting duties and should be aligned with current guidance.

28.1 Purpose of the Checklist

The purpose of the Runtime Monitoring and Incident Response Checklist is to help deployers turn monitoring into operational control.

Monitoring is not the same as observation.

A dashboard may show activity.

A log may record events.

A metric may show performance.

An alert may indicate a problem.

But none of these create control unless the organization knows what to do next.

For high-risk AI systems, monitoring must answer a deeper question:

Do current conditions still support continuation?

This question is the operational centre of Article 26 deployer governance.

The deployer should not monitor only whether the system is online.

It should monitor whether the AI-supported workflow remains within valid control conditions.

This includes:

- intended purpose;
- actual use;
- provider instructions;
- input data condition;
- context freshness;
- human oversight;
- authority state;
- scope boundary;

- model or system version;
- fallback state;
- downstream effect;
- log integrity;
- risk signals;
- incident indicators;
- escalation status.

A high-risk AI system may be technically healthy while governance conditions are failing.

The system may respond quickly.

The API may be available.

The model may generate outputs.

The workflow may complete.

The dashboard may remain green.

But the input data may be stale.

Human oversight may be unavailable.

The output may be used outside intended purpose.

A recommendation may have become a downstream trigger.

A fallback model may be carrying full authority silently.

Logs may be incomplete.

A risk signal may be active but unassigned.

This is why runtime monitoring must include governance signals, not only technical signals.

The checklist in this chapter is designed to help deployers answer:

What should we monitor?

What should trigger action?

Who receives alerts?

When should the workflow continue?

When should it hold?

When should it escalate?

When should it refuse?

When should use be suspended?

What evidence must be preserved?

Who contacts the provider?

Who assesses serious risk or incident?

Who approves return to service?

This checklist should be used:

- before deployment;
- during pilot operation;
- during production monitoring design;
- after system updates;
- after model fallback events;
- after workflow expansion;
- after incidents or near misses;
- during periodic reassessment;
- during audit preparation.

The goal is not to create unnecessary bureaucracy.

The goal is to ensure that when the system shows a signal, the organization can convert that signal into responsible action.

Monitoring without response is visibility without control.

Incident response without evidence is action without reconstructability.

A mature deployer needs both.

28.2 Monitoring Scope Checklist

The first step is to define the scope of monitoring.

A deployer should not begin with tools.

It should begin with control questions.

The checklist should ask:

- Which AI system is being monitored?
- Which workflow is affected?
- Is the system high-risk or high-consequence?
- What is the intended purpose?
- What is the approved use case?
- What output does the system produce?
- What consequence can follow from the output?
- Who uses the system?
- Who may be affected?
- What data sources does the system rely on?
- What human oversight is required?
- What logs are generated?
- What downstream systems receive the output?
- What could go wrong silently?
- What would require immediate action?

Monitoring scope should include both system health and control health.

System health may include:

- availability;
- latency;
- error rate;
- throughput;
- API status;
- tool availability;
- integration status;
- cost or token usage;
- infrastructure events.

Control health may include:

- actual use versus intended purpose;
- data freshness;
- missing required inputs;
- context quality;
- human oversight availability;
- review timing;
- override rates;
- authority state;
- scope drift;

- model fallback;
- downstream automation;
- log completeness;
- active alerts;
- unresolved escalations.

The deployer should define monitoring scope at three levels.

System Level

This tracks the AI system as a whole.

Examples:

- system availability;
- model version;
- provider status;
- fallback events;
- global error patterns;
- monitoring coverage;
- unresolved incidents.

Workflow Level

This tracks how the system is used in a specific operational process.

Examples:

- recruitment screening;
- supplier approval;
- clinical summarization;
- compliance case review;
- public-service triage;
- customer support response.

Workflow-level monitoring asks whether the system remains used within the approved workflow and consequence level.

Case Level

This tracks a specific AI-supported output or workflow instance.

Examples:

- one supplier recommendation;
- one candidate ranking;

- one patient summary;
- one compliance case;
- one public-service eligibility support record;
- one customer response.

Case-level monitoring asks whether control conditions were valid for that specific instance.

A mature deployer should connect all three levels.

System-level monitoring may show broad risk.

Workflow-level monitoring may show use drift.

Case-level monitoring may show whether a specific output was allowed to influence consequence.

The key evidence question is:

Do we know what we are monitoring, and do we know why it matters?

If the answer is no, monitoring may become noise.

28.3 Runtime Signal Categories

A runtime signal is any event, metric, pattern or condition that may indicate whether the AI-supported workflow remains valid.

Deployers should define signal categories before deployment.

A practical monitoring model may include twelve signal categories.

1. Technical Health Signals

These show whether the system is functioning technically.

Examples:

system unavailable;

API failure;

latency spike;

tool failure;

integration failure;

timeout;

rate-limit event;

provider outage;

unexpected output format.

Technical health signals matter because malfunction can affect control.

But they are not enough by themselves.

2. Model and Version Signals

These show whether the AI system or model changed.

Examples:

- model version changed;
- provider updated system;
- fallback model used;
- routing changed;
- prompt template changed;
- system instruction changed;
- safety setting changed;
- model deprecation notice.

These signals matter because model or configuration change may alter validation, authority or output behaviour.

3. Input Data Signals

These show whether input data supports the workflow.

Examples:

- stale data;
- missing required field;
- source unavailable;
- source conflict;
- source reliability unknown;
- representativeness concern;
- user-entered context unverified;
- data format changed;
- data pipeline failure.

Input data signals are critical because output quality depends on input condition.

4. Context Signals

These are especially important for large language model and agentic workflows.

Examples:

outdated context retrieved;
wrong policy version included;
context compaction removed warning;
source citation missing;
conflicting context present;
context freshness unknown;
unapproved document retrieved;
old conversation memory used.

Context signals matter because context shapes reasoning.

5. Human Oversight Signals

These show whether human oversight remains meaningful.

Examples:

reviewer unavailable;
review overdue;
review time unusually short;
approval rate unusually high;
override rate unusually low;
escalation rate drops suddenly;
reviewer lacks authority;
reviewer lacks evidence access;
workload exceeds capacity.

Human oversight can degrade silently.

Monitoring should detect that degradation.

6. Authority Signals

These show whether the workflow remains authorized.

Examples:

approver authority expired;
approval threshold exceeded;
user lacks permission;
workflow consequence increased;
delegation mismatch;

model fallback authority reduction required;
output used beyond approved authority;
downstream trigger detected.

Authority signals help prevent unauthorized continuation.

7. Scope and Intended-Purpose Signals

These show whether actual use remains within approved scope.

Examples:

new user group;
new workflow;
new data source;
output used for new decision type;
advisory output used as decision trigger;
system used outside provider instructions;
use case expands without reassessment.

Scope signals help detect operational drift.

8. Output Quality Signals

These show whether outputs may be unreliable or inappropriate.

Examples:

inconsistent output;
hallucinated source;
unsupported recommendation;
contradiction with source evidence;
refusal rate changes;
low confidence;
repeated human correction;
output omits required warnings;
generated reasoning conflicts with policy.

Output quality signals should connect to review and escalation.

9. Downstream Effect Signals

These show whether AI output is influencing other systems or consequences.

Examples:

- output written to official record;
- output triggers workflow;
- output sent externally;
- output changes case status;
- recommendation becomes default approval;
- AI-generated score used as threshold;
- downstream automation added.

Downstream effect signals are essential because risk often appears when output becomes action.

10. Affected-Person Signals

These show whether people are affected or raising concerns.

Examples:

- worker complaint;
- candidate challenge;
- customer dispute;
- patient concern;
- citizen appeal;
- student complaint;
- supplier objection;
- repeated data correction requests;
- affected-person information failure.

Affected-person signals are important governance inputs.

They may reveal issues monitoring does not detect.

11. Incident and Near-Miss Signals

These show actual or potential harm.

Examples:

- incorrect decision;
- unsafe recommendation;
- discriminatory pattern;
- privacy incident;
- patient safety near miss;
- unauthorized action;
- uncontrolled workflow execution;

serious risk signal;
repeated failure pattern.

Near misses should be treated seriously because they reveal control weaknesses before harm escalates.

12. Evidence Integrity Signals

These show whether the organization can reconstruct what happened.

Examples:

missing logs;
incomplete decision trail;
broken source reference;
missing human review record;
missing authority record;
missing context record;
timestamp mismatch;
evidence overwritten;
audit trail gap.

Evidence integrity signals matter because without reconstructability the organization may lose accountability.

The deployer should define which signals matter for each system.

Not every system requires every signal.

But high-risk, high-consequence and agentic systems require broader monitoring coverage.

28.4 Alert Severity Levels

Not every signal requires the same response.

A deployer should define alert severity levels.

A practical model may use five levels.

Level 0 — Informational

The signal is useful for trend monitoring but does not require immediate action.

Examples:

low-risk usage increase;
routine model latency variation;
non-critical user feedback;
scheduled provider update notice.

Response:

record;
review during normal monitoring cycle.

Level 1 — Warning

The signal indicates possible degradation but does not yet block use.

Examples:

evidence near freshness threshold;
review workload rising;
minor output inconsistency;
low-severity user correction pattern;
upcoming provider instruction change.

Response:

notify owner;
monitor trend;
consider preventive action.

Level 2 — Control Review Required

The signal indicates that a control condition may be weakening.

Examples:

repeated stale data warnings;
override rate drops unexpectedly;
actual use expands to adjacent workflow;
fallback model used in moderate-consequence workflow;
log completeness issue in non-critical cases.

Response:

review control;
document decision;
remediate if needed.

Level 3 — HOLD / Escalation Required

The signal indicates that continuation may not be justified until reassessment occurs.

Examples:

required data stale;
human oversight unavailable;
authority unclear;
use outside intended purpose;
active monitoring alert in high-consequence workflow;
context source conflict;
model fallback without approved authority;
incomplete logs for high-risk case.

Response:

HOLD affected workflow;
escalate to owner;
preserve evidence;
reassess before continuation.

Level 4 — Suspend / Refuse / Serious Risk Review

The signal indicates serious risk, incident potential, invalid use or unacceptable loss of control.

Examples:

serious risk to health, safety or fundamental rights;
AI output caused or may cause harmful consequence;
unauthorized system action;
prohibited use;
unresolved critical data failure;
human oversight impossible for critical workflow;
provider warning of unsafe operation;
repeated high-risk failures;
decision trail unreconstructable for critical workflow.

Response:

suspend or refuse use;
preserve evidence immediately;
escalate to senior owner;

assess provider / authority notification;
open incident investigation;
review affected cases.

The severity model should be adapted to the sector.

Healthcare may require faster escalation for patient safety.

Financial services may require escalation for regulated process impact.

Employment may require escalation for discriminatory or unfair impact.

Public-sector AI may require escalation where rights, eligibility or access are affected.

Procurement may require escalation for sanctions, cybersecurity or critical supplier risk.

The key principle is proportionality.

The higher the consequence, the lower the tolerance for uncertainty.

28.5 ALLOW, HOLD, ESCALATE and REFUSE Logic

Monitoring should connect to governance states.

The four core states are:

ALLOW.

HOLD.

ESCALATE.

REFUSE.

ALLOW

ALLOW means current control conditions support continuation.

Minimum conditions may include:

use within intended purpose;
required input data available;
evidence freshness acceptable;
human oversight completed where required;
authority valid;
no unresolved critical alerts;

logs complete enough for reconstruction;
downstream consequence controlled.

ALLOW should be logged where the workflow is high-consequence.

The record should show why continuation was permitted.

HOLD

HOLD means the workflow pauses because a required control condition is missing, incomplete, uncertain or degraded.

HOLD may be triggered by:

stale evidence;
missing required data;
human oversight unavailable;
authority unclear;
context conflict;
active monitoring alert;
model fallback requiring review;
log integrity gap;
downstream effect unknown.

HOLD should preserve the workflow state and prevent consequence until the issue is resolved.

ESCALATE

ESCALATE means the issue requires review by a higher authority or specialist function.

Escalation may go to:

human overseer;
process owner;
AI governance owner;
risk owner;
compliance;
legal;
HR;
clinical safety;
procurement risk;
information security;

provider;
senior management;
market surveillance authority where required.

Escalation should be tracked until closure.

REFUSE

REFUSE means the workflow should not proceed under current conditions.

REFUSE may be required when:

use is outside intended purpose;
required human oversight is impossible;
evidence cannot be restored;
authority is absent;
output would create prohibited consequence;
critical logs are missing;
serious risk remains unresolved;
provider instructions prohibit the use;
fallback model is not approved.

REFUSE should be logged as evidence of control.

A refusal shows that the system had a boundary and preserved it.

The checklist should ask:

Are ALLOW conditions defined?
Are HOLD conditions defined?
Are ESCALATE conditions defined?
Are REFUSE conditions defined?
Are these states implemented in workflow logic?
Are users trained on these states?
Are these states visible in the interface?
Are state changes logged?
Can a workflow resume after HOLD only with documented reason?
Can REFUSE be overridden only through authorized governance review?

If these states are not defined, monitoring may fail to produce action.

28.6 Incident Triage Checklist

Incident triage determines what happened, how serious it is and what response is required.

Triage should begin quickly.

The organization should not wait until every fact is known before preserving evidence or preventing further harm.

The checklist should ask:

What signal triggered triage?

Which AI system is involved?

Which workflow is affected?

Which output, action or decision is involved?

What happened?

When did it happen?

Who detected it?

Who or what may be affected?

Is the system still in use?

Could continued use create further harm?

Is immediate HOLD or suspension required?

Is the issue technical, operational, data-related, oversight-related, authority-related, context-related or incident-related?

Are logs and context records preserved?

Is human review required?

Is provider contact required?

Is legal, compliance or safety review required?

Could this be a serious incident?

Could this indicate serious risk?

Are affected cases identifiable?

Who owns the next decision?

Triage should classify the issue.

Possible categories include:

technical malfunction;

model behaviour issue;

data freshness failure;

input data quality failure;

context retrieval failure;
human oversight failure;
authority failure;
scope drift;
downstream automation failure;
log integrity failure;
affected-person complaint;
near miss;
serious risk signal;
serious incident candidate.

The triage record should include:

triage owner;
triage time;
severity level;
immediate action;
evidence preserved;
escalation route;
next review time;
decision on continued use.

The key question is:

Can the system continue safely while the issue is assessed?

If the answer is uncertain and consequence is high, HOLD should be considered.

28.7 Evidence Preservation Checklist

Evidence preservation should happen early.

If evidence is not preserved, the organization may lose the ability to reconstruct what happened.

The checklist should ask whether the following have been preserved:

AI system name and version.
Model version or fallback model.
Provider status.
User prompt or system instruction where relevant.
Context bundle.

Retrieved documents or source identifiers.
Input data condition.
Data freshness timestamps.
AI output.
Tool calls.
Agent plan or intermediate steps where available.
Human review record.
Reviewer identity or role.
Authority state.
Policy version.
Monitoring alerts.
Logs.
Downstream system actions.
External communication.
Affected cases.
User reports.
Error messages.
Screenshots where useful.
Configuration state.
Escalation messages.
Provider communication.
Corrective action record.

Evidence should be protected from alteration.

Where possible, the organization should preserve:

original records;
timestamps;
hash or integrity metadata;
access records;
retention hold;
chain of custody where appropriate.

Evidence preservation should be proportionate to risk.

For a minor warning, standard log retention may be enough.

For a serious risk signal, stronger preservation may be needed.

For healthcare, public-sector, employment, financial or safety-related incidents, evidence preservation should be immediate and structured.

The key evidence question is:

Can we reconstruct the system state, evidence state, oversight state and consequence path at the time of the event?

If not, incident response will be weakened.

28.8 Provider and Third-Party Notification Checklist

The deployer should know when and how to contact the provider or relevant third parties.

This should be prepared before incidents occur.

The checklist should ask:

Who is the provider contact?

What is the provider incident channel?

What response time is expected?

What information must be supplied?

Is there a contractual notification requirement?

Are distributor or importer contacts relevant?

Are model provider contacts different from application provider contacts?

Are cloud, data or tool providers involved?

Are support hours adequate for the use case?

What happens if the provider is unavailable?

Does the provider need logs or evidence?

Can evidence be shared lawfully and securely?

Does the provider have a known issue or outage?

Has the provider issued safety or update notices?

Does the provider need to suspend, patch or update the system?

Is provider communication logged?

Who approves external provider communication?

Who tracks provider response?

Who decides whether system use resumes?

Who updates internal evidence records?

Provider communication should be documented.

The record should show:

date and time;

- provider contacted;
- issue summary;
- evidence shared;
- provider response;
- corrective action;
- expected timeline;
- impact on deployment;
- decision on continued use.

Third-party dependency matters especially in agentic systems.

An incident may involve:

- model provider;
- AI application provider;
- cloud provider;
- data provider;
- tool provider;
- workflow system;
- identity provider;
- logging provider;
- integration partner.

The deployer should know which third party affects which control condition.

The key question is:

Can the organization activate provider or third-party support without delay when control conditions fail?

If not, incident response readiness is incomplete.

28.9 Serious Risk and Serious Incident Assessment Checklist

Not every issue is a serious incident.

But the deployer must be able to assess seriousness quickly and consistently.

The checklist should ask:

Did the event cause harm?

Could the event have caused harm?

Did it affect health or safety?

Did it affect fundamental rights?

Did it affect employment opportunity or worker treatment?

Did it affect access to public services?

Did it affect financial, legal or regulated decisions?

Did it affect patients, citizens, students, workers, customers or vulnerable groups?

Did it affect many people or cases?

Is the harm reversible?

Did the AI output influence the consequence?

Was the system used according to instructions?

Was input data valid?

Was human oversight present and meaningful?

Was authority valid?

Did the system operate outside intended purpose?

Did downstream automation amplify the effect?

Did logs allow reconstruction?

Was provider malfunction involved?

Is continued use likely to create further risk?

A serious risk or serious incident assessment should involve appropriate functions.

Depending on context, this may include:

AI governance;

legal;

compliance;

risk;

data protection;

information security;

clinical safety;

HR;

public-service accountability;

procurement risk;

senior management;

provider contact.

The assessment should produce a decision:

not an incident;

minor issue;

near miss;

control weakness;

serious risk;
serious incident candidate;
serious incident confirmed;
provider notification required;
authority notification assessment required;
use suspended;
affected cases review required.

The record should include:

assessment date;
assessors;
facts known;
unknowns;
affected systems;
affected persons or groups;
immediate actions;
legal or regulatory assessment;
provider communication;
authority communication where required;
final classification;
corrective action.

The key governance principle is:

Uncertainty should not automatically become continuation.

If seriousness cannot be ruled out and consequence is high, the workflow should hold or suspend until assessment is complete.

28.10 Suspension and Return-to-Service Checklist

Suspending use is sometimes necessary.

Suspension should not be treated as panic.

It is a controlled response to serious uncertainty or risk.

The checklist should ask:

What system or function is being suspended?

Is suspension full or partial?

Which workflows are affected?
Which users are affected?
Which cases are paused?
Are downstream systems stopped?
Are external communications paused?
Is manual fallback available?
Who approved suspension?
Who communicates suspension internally?
Who contacts the provider?
Who preserves evidence?
Who reviews affected cases?
What conditions must be met before restart?
Who approves return to service?

Suspension may be:

Full Suspension

The AI system is stopped entirely.

Appropriate where serious risk is broad, uncontrolled or unknown.

Partial Suspension

Only a specific workflow, feature, model, tool or output type is paused.

Appropriate where the issue is contained.

Reduced-Authority Operation

The system continues only in draft-only, advisory or human-reviewed mode.

Appropriate where risk can be controlled with reduced authority.

Manual Fallback

The organization returns to human-only or non-AI workflow.

Appropriate where continued AI use is not justified.

Return to service should require evidence.

The checklist should ask:

Has root cause been identified?

Has the provider responded?

Has the issue been remediated?

Has testing been completed?
 Have affected cases been reviewed?
 Have logs been preserved?
 Have users been informed?
 Has human oversight been restored?
 Has input data validity been restored?
 Has authority been revalidated?
 Has monitoring been updated?
 Has documentation been updated?
 Has training been updated if needed?
 Has governance approved restart?
 Are restrictions or conditions required?

The return-to-service record should include:

reason for suspension;
 corrective action;
 evidence reviewed;
 testing result;
 residual risk;
 conditions for restart;
 approval owner;
 restart date;
 monitoring enhancements;
 next review date.

The key question is:

What evidence shows that the system is safe and governed enough to resume?

If that evidence is missing, return to service should not be approved.

28.11 Incident Communication Checklist

Incident communication must be clear and controlled.

Poor communication can create confusion, underreaction, overreaction or loss of evidence.

The checklist should ask:

Who needs to know internally?

Who owns the message?

What facts are known?

What is still unknown?

What action has been taken?

Is system use paused?

Are users instructed not to use the system?

Are affected teams informed?

Is senior management informed?

Is legal or compliance review required?

Is provider communication required?

Is authority communication required?

Are affected persons to be informed?

Is customer, patient, worker, citizen or supplier communication required?

Is external communication controlled?

Are records of communication retained?

Internal communication may include:

system users;

process owners;

human overseers;

technical teams;

governance teams;

legal and compliance;

risk management;

data protection;

senior management;

customer or citizen service teams;

HR or clinical leads where relevant.

Communication should avoid unsupported claims.

It should distinguish:

confirmed facts;

suspected issues;

unknowns;

immediate controls;

next steps.

The key evidence question is:

Can the organization show that relevant people were informed quickly enough to prevent further uncontrolled use?

If not, incident response may fail even if technical remediation succeeds.

28.12 Post-Incident Review Checklist

After immediate response, the organization should perform a post-incident review.

The purpose is not blame.

The purpose is learning and remediation.

The checklist should ask:

What happened?

Why did it happen?

When was it detected?

How was it detected?

Was monitoring sufficient?

Was human oversight sufficient?

Was input data valid?

Was context valid?

Was authority valid?

Did fallback occur?

Did downstream automation amplify the effect?

Were logs complete?

Was evidence preserved?

Was escalation timely?

Was provider response adequate?

Were affected cases reviewed?

Did the system need suspension?

Was return to service justified?

What controls failed?

What controls worked?

What remediation is required?

Who owns remediation?

What deadlines apply?

What monitoring changes are needed?

What training changes are needed?

What policy changes are needed?

What provider changes are needed?

What reassessment is required?

Should deployment scope be reduced?

Should use be refused in certain workflows?

Post-incident review should produce:

root cause analysis;

control failure analysis;

affected-case analysis;

corrective action plan;

owner assignment;

due dates;

evidence update;

reassessment record;

governance reporting.

The review should also identify whether the incident was isolated or systemic.

A single wrong output may be isolated.

Repeated stale data warnings may be systemic.

A human oversight failure caused by workload may be systemic.

A context retrieval problem may affect many workflows.

A fallback model authority issue may affect multiple systems.

A log integrity gap may undermine audit readiness broadly.

The key question is:

What must change so the same failure does not silently recur?

28.13 Monitoring Dashboard Checklist

A runtime monitoring dashboard should support decision-making, not just display metrics.

For high-risk AI systems, a useful dashboard may include:

System State

system status;

model version;
provider status;
fallback status;
deployment environment;
last update.

Workflow State

active workflows;
approved use cases;
actual use patterns;
scope drift indicators;
downstream integration status.

Input Data State

data source health;
freshness status;
missing required fields;
source failures;
representativeness warnings where relevant.

Context State

policy version;
context freshness;
retrieval quality;
conflicting context;
compaction warnings;
source citation completeness.

Human Oversight State

review queue;
overdue reviews;
reviewer availability;
review time;
approval rate;
override rate;
escalation rate;
workload pressure.

Authority State

approval threshold warnings;
authority mismatch;
delegation expiry;
model authority reduction;
output used beyond scope.

Output and Quality State

output volume;
error patterns;
human corrections;
refusals;
uncertainty events;
unsupported output warnings.

Downstream Consequence State

outputs sent to downstream systems;
automated triggers;
external communications;
official record updates;
affected-person outcomes.

Log Integrity State

log completeness;
missing decision trails;
retention status;
evidence gaps;
audit readiness.

Incident State

open alerts;
open HOLD states;
open escalations;
refused workflows;
suspended systems;
incident investigations;
remediation status.

A dashboard should also show governance state.

A useful summary may include:

- systems in ALLOW;
- systems in WARNING;
- systems in HOLD;
- systems in ESCALATE;
- systems in REFUSE;
- systems suspended;
- systems overdue for reassessment.

The key dashboard question is:

Can a responsible person see which AI-supported workflows are no longer clearly safe to continue?

If the dashboard cannot answer that, it may be operationally weak.

28.14 Incident Response Drill Checklist

Incident response should be tested.

A process that exists only in a document may fail under pressure.

The deployer should run tabletop exercises or simulations.

Possible drill scenarios include:

Scenario 1 — Stale Data

A high-risk workflow continues using stale input data.

Test:

- detection;
- HOLD trigger;
- evidence refresh;
- human review;
- affected-case review.

Scenario 2 — Human Oversight Unavailable

A required reviewer is unavailable, but outputs continue.

Test:

- reviewer availability monitoring;

backup assignment;
escalation;
workflow pause.

Scenario 3 — Model Fallback

Primary model fails and fallback model is used.

Test:

fallback detection;
authority reduction;
user notification;
logging;
review requirement.

Scenario 4 — Use Outside Intended Purpose

A team starts using the AI system for a new workflow.

Test:

scope drift detection;
governance escalation;
deployment hold;
reassessment.

Scenario 5 — Downstream Automation

AI recommendations begin triggering workflow actions.

Test:

downstream effect monitoring;
authority validation;
control gate;
refusal condition.

Scenario 6 — Missing Logs

A serious case cannot be reconstructed.

Test:

log integrity alert;
evidence preservation;
suspension decision;

remediation.

Scenario 7 — Affected-Person Complaint

A worker, patient, customer, student or citizen challenges an AI-supported outcome.

Test:

complaint intake;
case reconstruction;
human review;
correction pathway;
communication.

Scenario 8 — Serious Risk Signal

The AI system may have contributed to harmful or unlawful outcomes.

Test:

immediate triage;
suspension;
provider contact;
legal/compliance review;
authority notification assessment;
affected-case review.

Each drill should record:

scenario;
participants;
time to detect;
time to escalate;
evidence preserved;
decision made;
gaps found;
remediation actions;
owner;
due date.

The key question is:

Can the organization act under pressure, or only describe what it would do?

28.15 Minimum Runtime Monitoring Evidence Pack

For each high-risk AI system, the deployer should maintain a runtime monitoring evidence pack.

This pack should include:

Monitoring scope.

Signal categories.

Alert thresholds.

ALLOW, HOLD, ESCALATE and REFUSE rules.

Dashboard or monitoring view.

Monitoring owner.

Alert owner.

Escalation matrix.

Incident triage procedure.

Evidence preservation checklist.

Provider contact records.

Serious risk assessment procedure.

Suspension procedure.

Return-to-service procedure.

Communication plan.

Post-incident review template.

Incident drill records.

Monitoring review records.

Open alert register.

Remediation tracker.

Decision trail samples.

Evidence integrity checks.

The pack should answer:

What is monitored?

Who monitors it?

What triggers action?

Who receives alerts?

When does the workflow hold?

When does it escalate?

When does it refuse?

How is evidence preserved?

How are incidents assessed?

How does the organization decide whether the system can continue?

If the evidence pack cannot answer these questions, runtime monitoring may not be mature enough for high-risk deployment.

28.16 Runtime Monitoring Decision Table

The following table provides a practical summary.

Signal	Possible Risk	Governance State	Required Action	Evidence
Data stale	Output may rely on invalid evidence	HOLD	Refresh data or escalate	Freshness log
Human reviewer unavailable	Oversight cannot operate	HOLD / ESCALATE	Assign backup or pause workflow	Oversight record
Use outside intended purpose	Scope breach	ESCALATE / REFUSE	Governance reassessment	Use map
Fallback model used	Authority continuity issue	HOLD / Reduced authority	Apply fallback rules	Fallback record
Downstream automation detected	Advisory output may become action	ESCALATE	Validate authority and controls	Downstream map
Log missing	Decision cannot be reconstructed	HOLD / ESCALATE	Preserve evidence, fix logging	Log integrity record
Serious risk signal	Harm or rights impact possible	SUSPEND / ESCALATE	Incident assessment	Incident record
Output contradicts evidence	Reliability concern	HOLD	Human review required	Output review

Signal	Possible Risk	Governance State	Required Action	Evidence
High approval rate spike	Automation bias risk	REVIEW	Oversight quality review	Oversight metrics
Affected-person complaint	Possible harm or unfairness	ESCALATE	Case reconstruction	Complaint record

This table should be adapted by sector and consequence level.

The principle remains the same:

A monitoring signal should connect to a governance state and an evidence record.

Chapter 28 Summary

Runtime monitoring is not only technical observation.

For high-risk AI systems, runtime monitoring is the deployer's ability to detect whether current control conditions still support continuation.

This chapter provides a practical Runtime Monitoring and Incident Response Checklist.

It begins by defining monitoring scope across system level, workflow level and case level.

It then identifies twelve runtime signal categories:

Technical health

Model and version

Input data

Context

Human oversight

Authority

Scope and intended purpose

Output quality

Downstream effect

Affected-person signals

Incident and near-miss signals

Evidence integrity

The chapter proposes alert severity levels from informational signals to serious risk or suspension-level events.

It connects monitoring to the four governance states developed throughout this study:

ALLOW.

HOLD.

ESCALATE.

REFUSE.

The chapter also provides checklists for incident triage, evidence preservation, provider notification, serious risk assessment, suspension, return to service, communication, post-incident review, dashboards and incident drills.

The central message is simple:

Monitoring is useful only when it changes what the organization does.

A dashboard that shows risk but does not trigger action is not enough.

A log that records failure but does not support reconstruction is not enough.

An alert that no one owns is not enough.

For Article 26 deployer governance, runtime monitoring must connect signals to action, action to evidence, and evidence to control.

The central question is:

Can the organization detect when an AI-supported workflow should no longer continue, and can it prove what it did next?

If yes, monitoring supports governance.

If no, monitoring is only visibility.

Chapter 29

Board Briefing and Executive Summary Template

Legal anchor and framework boundary. [Article 26](#). The board template is a governance tool and does not create a statutory board-reporting format.

29.1 Purpose of the Board Briefing

The purpose of the Board Briefing and Executive Summary Template is to help deployers communicate Article 26 readiness to senior decision-makers.

AI governance cannot remain only a technical, legal or compliance exercise.

High-risk AI systems may affect employment, healthcare, finance, public services, procurement, education, customer treatment, operational resilience and fundamental rights.

This means senior leadership must understand not only whether AI systems exist, but whether they are controlled.

A board or executive committee does not need every technical detail of every model, prompt, log or workflow.

But it does need decision-grade visibility.

It needs to know:

- which high-risk or high-consequence AI systems are in use;
- what processes they affect;
- what people or entities may be affected;
- whether the systems are used within intended purpose;
- whether human oversight is meaningful;
- whether input data is suitable;
- whether monitoring detects relevant risk;
- whether logs can reconstruct decision trails;
- whether incident pathways are tested;
- whether provider and model dependencies are governed;
- whether there are unresolved control gaps;
- whether the organization should continue, hold, escalate or refuse certain uses.

The board briefing should therefore avoid two extremes.

The first extreme is technical overload.

A long technical report may contain useful detail but fail to show what leadership must decide.

The second extreme is superficial assurance.

A short statement such as “AI governance is in place” may comfort leadership but fail to show whether control conditions are valid.

The correct briefing sits between these extremes.

It should be concise, structured, evidence-based and decision-oriented.

The briefing should answer:

What is our current AI deployer risk position?

What evidence supports that position?

What changed since the last briefing?

Where are controls strong?

Where are controls weak?

Where do we need executive decision?

Where should use be limited, paused or refused?

This is especially important because Article 26 deployer readiness is not a one-time state.

It changes when:

- new AI systems are deployed;
- actual use expands;
- provider instructions change;
- model versions change;
- fallback models are used;
- data sources change;
- human oversight capacity changes;
- downstream integrations are added;
- incidents occur;
- affected-person complaints arise;
- regulatory expectations evolve.

The board briefing should therefore be periodic and event-triggered.

A periodic briefing may occur quarterly or semi-annually depending on risk.

An event-triggered briefing may occur after a serious incident, high-risk deployment, major system expansion, unresolved control failure or material provider dependency issue.

The central purpose is simple:

Leadership should not discover AI control failure only after harm, audit failure or regulatory inquiry.

The briefing should create early visibility.

29.2 What Senior Leadership Needs to Know

Senior leadership needs to understand AI governance in operational terms.

A useful executive briefing should not begin with abstract AI principles.

It should begin with the organization's control position.

The board needs to know:

1. Inventory

Which AI systems are in use?

Which are high-risk?

Which are high-consequence even if not formally classified as high-risk?

Which are pilots?

Which are production systems?

Which are agentic or tool-connected?

2. Process Impact

Which business, clinical, employment, public-service, financial, compliance or operational processes are affected?

Does the AI system support drafting, ranking, scoring, summarization, recommendation, routing, decision support or workflow execution?

3. People and Consequence

Who may be affected?

Workers?

Applicants?

Customers?

Patients?

Citizens?

Students?

Suppliers?

Clients?

Investors?

Internal users?

What consequence can follow from the AI output?

4. Control State

Are the required controls active?

Use according to instructions?

Human oversight?

Input data controls?

Monitoring?

Logs?

Escalation?

Affected-person information?

Provider dependency controls?

5. Evidence Quality

Can the organization prove that controls operate?

Is evidence based on assertion, policy, process records, operational evidence, reconstructable decision trails or runtime-assured control?

6. Current Risk

What are the most important AI governance risks?

Are there unresolved alerts?

Are there open incidents?

Are there systems in HOLD?

Are any systems operating with reduced authority?

Are any workflows outside approved scope?

7. Decision Required

What does leadership need to approve, fund, pause, escalate or refuse?

Does a system need remediation?

Does a deployment need limitation?

Does a provider risk need review?

Does oversight capacity need investment?

Does logging need improvement?

Does a use case need to stop?

Senior leadership does not need a full case record for every workflow.

But it needs enough evidence to know whether the organization can reconstruct and control AI-supported decisions.

A strong briefing should therefore distinguish between:

- AI adoption status;
- AI control maturity;
- AI evidence readiness;
- AI risk exposure;
- AI decision requests.

This prevents a common leadership mistake:

confusing AI activity with AI readiness.

An organization may be very active with AI and still weak in deployer governance.

The board briefing should make that visible.

29.3 Executive Summary Template

The following template can be used as the opening page of a board or executive AI governance briefing.

Executive Summary: Article 26 Deployer Governance

Reporting Period:

Insert reporting period.

Prepared By:

AI governance owner, compliance, risk, legal, technology or accountable function.

Executive Owner:

Senior accountable owner.

Overall Status:

Green / Amber / Red / Mixed.

Summary Statement:

Briefly state the organization's current deployer governance position.

Example:

“The organization has identified all currently approved high-risk AI systems and has established baseline controls for intended purpose, human oversight and monitoring. However, two systems require remediation before expanded production use because decision-trail logging and human oversight evidence remain incomplete.”

1. AI System Inventory Status

Total AI Systems Identified:

Number.

High-Risk AI Systems:

Number.

High-Consequence AI Systems:

Number.

Agentic / Tool-Connected Systems:

Number.

Systems in Pilot:

Number.

Systems in Production:

Number.

Unassessed or Pending Classification:
Number.

2. Material Use Cases

List the most important AI-supported workflows.

Use Case	System	Process Affected	Consequence Level	Status
Example	HR Screening Assistant	Recruitment shortlist	High	HOLD pending oversight remediation
Example	Supplier Risk Assistant	Procurement approval	High	Production with conditions
Example	Compliance Case Assistant	AML case summary	High	Controlled pilot

3. Control Maturity Overview

Control Domain	Status	Key Gap
System Identification	Green	Register complete
Intended Purpose	Amber	Actual use drift in HR pilot
Human Oversight	Amber	Review capacity not yet proven
Input Data	Amber	Freshness thresholds incomplete
Monitoring	Amber	Governance alerts not fully automated
Logs and Decision Trails	Red	Two workflows not reconstructable
Escalation	Green	Incident pathway tested
Affected-Person Information	Amber	Notices need update after scope change
Provider Dependency	Amber	Fallback rules incomplete

4. Key Risk Issues

Summarize the top three to five AI governance risks.

Example:

- HR AI tool is being used for ranking, although approved use was summarization only.
- Procurement AI system lacks complete cybersecurity evidence before supplier recommendation.
- Compliance assistant fallback model rules are not yet mapped to authority levels.
- Clinical summarization pilot requires stronger missing-data warnings.
- Decision-trail logs are incomplete for one public-service workflow.

5. Incidents, Near Misses and Escalations

Open Incidents:

Number and summary.

Near Misses:

Number and summary.

Systems in HOLD:

Number and reason.

Systems Suspended:

Number and reason.

Provider Notifications:

Number and summary.

Authority Notifications:

Number and summary where applicable.

6. Decisions Requested

List specific board or executive decisions.

Examples:

- Approve remediation budget for decision-trail logging.
- Approve temporary limitation of HR AI use to summarization only.
- Approve mandatory human review before supplier recommendations become approval workflows.
- Approve suspension of a high-risk workflow until input data controls are complete.
- Approve creation of an AI incident response function.

7. Next Reporting Date

Insert date or trigger.

This executive summary should be short enough to read quickly but strong enough to support real decision-making.

It should not hide uncertainty.

If a control is unknown, say unknown.

Unknown is not the same as green.

29.4 Board-Level AI Governance Questions

A board or executive committee should ask practical questions.

These questions should not be limited to whether the organization has an AI policy.

A policy is important, but it does not prove operational control.

The following questions are designed to test deployer governance maturity.

System Visibility

- Do we know which AI systems are currently in use?
- Do we know which are high-risk or high-consequence?
- Do we know which are pilots and which are production systems?
- Do we know which systems are agentic or connected to tools?
- Do we know which systems affect natural persons?

Intended Purpose and Actual Use

- Are systems being used within provider-defined intended purpose?
- Are internal approved use cases documented?
- Has actual use expanded beyond original approval?
- Are advisory outputs becoming operational triggers?
- Are any systems being used outside approved scope?

Human Oversight

- Who performs human oversight?
- Are those people competent and trained?
- Do they have authority to override or stop workflows?
- Does review occur before consequence attaches?
- Are override, escalation and refusal records monitored?

Input Data and Context

What data do high-risk systems rely on?

Are freshness thresholds defined?
Are missing or stale data conditions visible?
Can workflows hold when evidence is insufficient?
Are LLM context sources inspectable and logged?

Monitoring and Logs

Do we monitor governance conditions, not only uptime?
Can monitoring trigger HOLD, ESCALATE or REFUSE?
Can we reconstruct a specific AI-supported decision trail?
Are logs complete enough for audit and incident response?
Are log integrity gaps reported to leadership?

Authority and Boundaries

What authority does each AI output carry?
Are approval thresholds mapped?
Can downstream systems act on AI output?
Are model fallback rules connected to authority reduction?
Are boundary expansions detected?

Incidents and Escalation

What AI incidents or near misses occurred this period?
Were affected cases identified?
Was evidence preserved?
Were providers contacted where required?
Were any systems suspended or returned to service?

Affected Persons and Transparency

Were workers or representatives informed where required?
Were affected natural persons informed where required?
Do notices still reflect actual use?
Are complaints or challenges tracked as governance signals?
Can affected-person cases be reconstructed?

Provider and Third-Party Dependency

Which providers are critical?
What happens if a primary model is unavailable?
Are fallback models approved?
Do provider contracts support incident response and evidence access?
Are update and change notices tracked?

Strategic Governance

Are we investing enough in AI evidence infrastructure?

Are we scaling AI faster than our controls?

Which AI uses should not expand until remediation is complete?

What is our highest AI governance risk today?

What decision does leadership need to make now?

These questions help leadership move beyond passive assurance.

They make AI governance a live control conversation.

29.5 Board Dashboard Metrics

A board dashboard should not overwhelm leadership with technical metrics.

It should provide a small number of decision-grade indicators.

The dashboard should show whether AI systems are controlled, not only whether they are active.

A useful board dashboard may include the following metrics.

Inventory Metrics

Number of AI systems identified.

Number of high-risk systems.

Number of high-consequence systems.

Number of agentic or tool-connected systems.

Number of unassessed systems.

Number of systems in production.

Number of systems in pilot.

Control Maturity Metrics

Percentage of systems with documented intended purpose.

Percentage of systems with actual-use mapping.

Percentage of systems with assigned human oversight.

Percentage of systems with completed system-specific training.

Percentage of systems with input data controls.

Percentage of systems with runtime monitoring.

Percentage of systems with reconstructable decision trails.

Percentage of systems with tested escalation pathways.

Runtime Governance Metrics

Systems in ALLOW state.
Systems in HOLD state.
Systems in ESCALATE state.
Systems in REFUSE state.
Systems operating with reduced authority.
Systems overdue for reassessment.
Systems with unresolved monitoring alerts.
Systems with active provider dependency issues.

Human Oversight Metrics

Review volume.
Average review time.
Override rate.
Escalation rate.
Refusal rate.
Review backlog.
Reviewer capacity.
Training completion.
Oversight quality findings.

Data and Context Metrics

Stale evidence events.
Missing required data events.
Context retrieval failures.
Policy version mismatches.
Data source outages.
Evidence freshness exceptions.
Context compaction warnings.

Incident and Escalation Metrics

AI incidents.
Near misses.
Serious risk reviews.
Suspensions.
Return-to-service decisions.
Provider notifications.
Authority notification assessments.

Open remediation items.

Affected-Person Metrics

Complaints or challenges involving AI-supported workflows.

Data correction requests.

Worker or representative concerns.

Candidate review requests.

Patient, customer, citizen or student concerns.

Appeal outcomes linked to AI-supported processes.

Provider and Model Dependency Metrics

Provider outages.

Model fallback events.

Unauthorized fallback attempts.

Reduced-authority operations.

Provider instruction updates.

Model version changes.

Unresolved third-party evidence gaps.

The dashboard should use clear status categories.

Green means control conditions are operating and evidenced.

Amber means control exists but has gaps or requires monitoring.

Red means control is missing, failed or insufficient for consequence.

Grey means unknown or unassessed.

Grey should not be treated as safe.

Unknown is a risk state.

A board dashboard should also show trend.

Is the organization improving or degrading?

Are incidents increasing?

Are HOLD states increasing?

Are override rates dropping?

Are systems expanding faster than controls?

Are evidence gaps being remediated?

The goal is not to create perfect metrics.

The goal is to give leadership a realistic view of AI control maturity.

29.6 Board Decision Paper Template

When leadership must approve, pause, expand or refuse an AI system, the organization should prepare a decision paper.

The decision paper should be structured around control evidence.

Board Decision Paper: AI System Deployment or Continued Use

1. Decision Requested

State clearly what leadership is being asked to decide.

Examples:

Approve controlled production deployment.

Approve expansion to new workflow.

Approve continued use with conditions.

Approve suspension.

Approve remediation funding.

Refuse use under current conditions.

2. AI System Overview

System name.

Provider.

Use case.

Intended purpose.

Actual use.

User group.

Affected persons.

Output type.

Consequence level.

3. Strategic Rationale

Explain why the organization wants to use or continue using the system.

Include:

operational benefit;

efficiency benefit;
risk reduction benefit;
service improvement;
compliance benefit;
customer or citizen benefit;
innovation rationale.

4. Risk Classification

State whether the system is high-risk, high-consequence, agentic, tool-connected or otherwise sensitive.

Include rationale.

5. Control Evidence Summary

Summarize evidence for:

use according to instructions;
human oversight;
input data;
monitoring;
logs;
escalation;
authority;
affected-person information;
provider dependency.

6. Key Gaps

Identify unresolved gaps.

Examples:

logs incomplete;
human oversight capacity uncertain;
input data freshness thresholds undefined;
provider fallback rules incomplete;
affected-person notice not updated;
actual use drift detected.

7. Proposed Conditions

List conditions for approval.

Examples:

limited user group;
no downstream automation;
human review before consequence;
decision-trail logging required;
monthly monitoring report;
no model fallback without authority reduction;
scope review after pilot;
escalation pathway test before production.

8. Residual Risk

Explain the risk that remains after controls.

Be clear.

Do not hide uncertainty.

9. Alternatives

List options.

Approve.

Approve with conditions.

Hold pending remediation.

Pilot only.

Reduce authority.

Suspend.

Refuse.

10. Recommendation

State the recommended decision.

11. Accountability

Identify who owns the system, risk, monitoring, remediation and next review.

12. Next Review

State review date and triggers.

This decision paper helps leadership make explicit governance decisions.

It prevents AI deployment from becoming a purely technical rollout.

29.7 Board Briefing: Red Flag List

The board should receive clear red flags.

A red flag does not always mean immediate suspension.

But it means leadership attention is required.

Examples of AI governance red flags include:

High-risk AI system in production without clear owner.

Actual use differs from provider intended purpose.

AI output used for decisions beyond approved scope.

Human oversight exists only as a final approval click.

Human reviewers lack training or authority.

Review occurs after consequence attaches.

Input data freshness is unknown.

Required data sources are missing or unreliable.

Decision trails cannot be reconstructed.

Logs exist but do not connect input, output, review and consequence.

Monitoring covers uptime but not governance conditions.

Alerts are not assigned to owners.

No tested AI incident escalation pathway.

Provider contact route is unclear.

Fallback models are used without approval.

AI recommendations trigger downstream workflows without control gate.

Affected persons are not informed where required.

Workplace AI notices do not match actual use.

Complaints or appeals suggest AI-related issues.

System continues despite unresolved serious risk signal.

Model or provider changes are not reassessed.

Authority thresholds are not mapped.

Advisory output has become practically decisive.

Agentic system has tool access without permission controls.

Board receives only adoption metrics, not control metrics.

These red flags should be used during board review.

If several appear together, the organization may be scaling AI faster than governance.

The strongest board question in such cases is:

Should this use continue under current conditions?

29.8 Sample One-Page Board Briefing

The following sample shows how a concise board briefing may look.

AI Deployer Governance Board Briefing

Reporting Period: Q2 2026
Prepared By: AI Governance Office
Overall Status: Amber

Executive Summary

The organization has identified 18 AI systems in active use. Four are classified as high-risk or high-consequence. Two are in controlled pilot, and two are in production with conditions. The main governance gap is decision-trail reconstructability for AI-supported employment and procurement workflows. No confirmed serious incidents occurred during the period, but three near misses were recorded.

System Status

System	Use Case	Status	Main Issue
HR Screening Assistant	Candidate summaries and ranking	HOLD	Ranking use exceeds approved scope
Supplier Risk Assistant	Supplier recommendations	Amber	Cybersecurity evidence freshness gap
Compliance Case Assistant	Case summarization	Green	Controlled pilot with human review
Customer Support Agent	Draft responses	Amber	Downstream automation risk under review

Key Risks

HR system approved for summarization is being used informally for ranking.
Supplier recommendations may rely on incomplete cybersecurity review.
Customer support agent outputs may trigger workflow actions through downstream integration.
Decision trail logging is incomplete in two workflows.

Actions Taken

HR ranking use paused pending reassessment.

Supplier risk workflow updated to HOLD when cybersecurity review is pending.

Customer support downstream trigger disabled pending control-gate design.

Incident response drill completed for model fallback scenario.

Decisions Requested

Approve remediation budget for decision-trail logging.

Approve policy that AI-generated rankings cannot affect employment shortlists without documented human review.

Approve control-gate requirement before any AI output triggers downstream automation.

Next Review

Next board update in Q3 2026, or sooner if serious incident, model fallback in high-risk workflow, or unresolved HOLD exceeds 14 days.

This sample is intentionally short.

The supporting evidence pack should contain the detail.

The board paper should show the decision.

The evidence pack should support the proof.

29.9 Minimum Board Evidence Pack

For board-level oversight, the organization should maintain a minimum evidence pack.

This pack should be available to leadership, internal audit and governance functions.

It should include:

AI system inventory.

High-risk and high-consequence classification.

Material use-case map.

Intended purpose and actual use map.

Human oversight model.

Input data governance summary.

Monitoring dashboard.

Decision-trail maturity summary.

Incident and near-miss register.
Systems in HOLD, ESCALATE or REFUSE.
Provider and model dependency register.
Fallback and authority reduction rules.
Affected-person and workplace information records.
Open risk and remediation tracker.
Latest readiness matrix.
Internal audit or assurance findings.
Board decisions and approvals.
Reassessment schedule.
Serious risk or incident escalation records.
Evidence of return-to-service decisions where applicable.
The board evidence pack should not be a dumping ground.
It should be organized around decisions.
The structure should allow a director or executive to ask:
What are we using?
What can it affect?
What evidence proves control?
Where are the gaps?
What decisions are needed?
If the evidence pack cannot answer these questions, it is too weak for board oversight.

29.10 Executive Communication Principles

AI governance communication to leadership should follow several principles.

1. Say What Is Known

Do not hide uncertainty behind broad assurance.

If logs are incomplete, say so.

If actual use is not fully mapped, say so.

If human oversight quality is untested, say so.

If provider fallback rules are unclear, say so.

2. Distinguish Activity From Control

AI adoption metrics are not control metrics.

The number of AI tools used does not show readiness.

The number of users trained does not prove meaningful oversight.

The number of logs retained does not prove reconstructability.

3. Separate Technical Health From Governance Health

A system may be technically healthy and governance-weak.

The board should see both.

4. Use Control States

ALLOW, HOLD, ESCALATE and REFUSE are useful because they make governance actionable.

Leadership should know which systems are in which state.

5. Connect Risk to Consequence

Do not describe AI risk only as abstract model risk.

Explain who or what may be affected.

6. Show Evidence Quality

Differentiate between assertion, policy, process evidence, operational evidence and reconstructable evidence.

7. Ask for Clear Decisions

A board briefing should not only inform.

It should request action where needed.

8. Track Remediation

Every red or amber issue should have an owner, action and due date.

9. Report Change

Show what changed since the last briefing.

New systems.

New incidents.

New provider updates.

New model fallbacks.

New control gaps.

New remediations.

10. Keep the Language Operational

Avoid jargon where possible.

The board does not need to hear only about model architecture.

It needs to understand control, consequence and accountability.

These principles help transform AI governance reporting into leadership oversight.

Chapter 29 Summary

Board and executive briefing is essential for Article 26 deployer governance.

High-risk AI systems cannot be governed only through technical teams, compliance documents or isolated project controls.

Senior leadership needs decision-grade visibility into AI use, control maturity, evidence quality, unresolved risks and required decisions.

This chapter provides a Board Briefing and Executive Summary Template.

The briefing should answer:

Which AI systems are in use?

Which are high-risk or high-consequence?

What processes do they affect?

Who may be affected?

Are controls operating?

Can decisions be reconstructed?

Where are the gaps?

What must leadership decide?

The chapter distinguishes AI adoption from AI readiness.

An organization may use many AI systems but still lack evidence of control.

For this reason, board reporting should include control metrics, not only adoption metrics.

The chapter proposes board-level questions, dashboard metrics, a decision paper template, a red flag list, a sample one-page board briefing and a minimum board evidence pack.

The central governance principle is:

Leadership should not receive only reassurance.

It should receive enough evidence to know whether AI-supported workflows remain within valid control conditions.

The board does not need every technical detail.

But it must know when AI systems should continue, hold, escalate or refuse.

That is executive-level evidence of control.

Chapter 30

Audit Questions and Internal Review Guide

Legal anchor and framework boundary. [Article 12](#); [Article 26](#); [Article 73](#). Audit questions test operational evidence; they are not an official supervisory checklist.

30.1 Purpose of the Internal Review Guide

The purpose of this guide is to help internal audit, compliance, risk, legal, technology, data governance and AI governance teams review whether Article 26 deployer controls are operating in practice.

The review should not ask only:

Does the organization have an AI policy?

Does the organization have an AI system register?

Does the organization have human oversight language?

Does the organization retain logs?

Does the organization have an incident procedure?

These are useful questions, but they are not enough.

The stronger audit question is:

Can the organization prove that the AI system was used under valid control conditions when its output influenced consequence?

This guide is built around that question.

An internal review should test whether the deployer can show:

- what AI system was used;
- for what purpose;
- in which workflow;
- with what provider instructions;
- with what input data;
- with what human oversight;
- under what authority;
- with what monitoring;
- with what logs;
- with what escalation route;

- with what downstream consequence;
- and why continuation was allowed.

The review should distinguish between four levels of assurance.

First, existence.

Does the control exist?

Second, design.

Is the control designed appropriately for the risk?

Third, operation.

Did the control operate in real workflows?

Fourth, evidence.

Can the organization prove that the control operated when it mattered?

Many organizations will have existence and design evidence.

Fewer will have operational and reconstructable evidence.

The purpose of internal review is to find that gap before an external audit, incident, regulator inquiry, litigation, affected-person challenge or board concern exposes it.

This guide can be used for:

- pre-deployment review;
- post-deployment review;
- periodic internal audit;
- incident follow-up;
- board assurance;
- third-party assurance;
- sector-specific review;
- model fallback review;
- agentic AI review;
- high-risk AI readiness assessment.

The goal is not to punish AI adoption.

The goal is to make AI use governable.

Good internal review should help the organization move from claims of control to evidence of control.

30.2 Audit Scope Definition

Before beginning the review, the audit team should define scope clearly.

AI systems vary widely.

A narrow internal drafting tool does not require the same review depth as a high-risk AI system affecting employment, healthcare, finance, public services or regulated decisions.

The audit scope should identify:

- systems included;
- workflows included;
- business units included;
- time period reviewed;
- risk categories;
- high-risk systems;
- high-consequence systems;
- agentic or tool-connected systems;
- third-party systems;
- pilots and production deployments;
- case samples;
- documents to be reviewed;
- teams to be interviewed.

A useful scope statement may read:

“This internal review assesses whether the organization’s high-risk and high-consequence AI systems are deployed and operated under Article 26-aligned control conditions, including intended-purpose alignment, instructions for use, human oversight, input data controls, runtime monitoring, log retention, escalation pathways, affected-person information, authority boundaries and evidence of control.”

The review should avoid vague scope.

A statement such as “review AI governance” is too broad.

A stronger scope is:

“Review AI-supported recruitment screening workflow from candidate intake to shortlist decision, including system use, input data, human review, ranking influence, candidate information, logs, escalation and decision trail evidence.”

Or:

“Review supplier risk AI workflow from supplier data retrieval to recommendation and approval routing, including evidence freshness, human oversight, authority thresholds, downstream consequence and HOLD / ESCALATE logic.”

The more specific the scope, the stronger the audit result.

30.3 Audit Evidence Sources

The audit team should collect evidence from multiple sources.

No single source is enough.

Policies show intended control.

Logs show activity.

Interviews show practical understanding.

Case samples show whether controls operated.

Monitoring data shows patterns.

Incident records show how the organization responds under pressure.

A complete internal review may examine:

Governance Documents

- AI policy;
- AI risk framework;
- Article 26 readiness matrix;
- system register;
- high-risk classification records;
- governance committee minutes;
- board papers;
- internal procedures.

Provider and System Documents

- provider instructions for use;
- technical documentation;
- limitations;
- intended purpose;
- user guidance;

- update notices;
- incident contact details;
- model dependency records.

Workflow Documents

- process maps;
- use-case approvals;
- workflow diagrams;
- decision points;
- human review steps;
- downstream integration maps;
- access permission records.

Human Oversight Evidence

- role descriptions;
- training records;
- reviewer competence records;
- review logs;
- override records;
- escalation records;
- reviewer feedback;
- workload records.

Data and Context Evidence

- data source maps;
- data quality checks;
- freshness thresholds;
- retrieval logs;
- context bundles;
- policy version records;
- missing-data warnings;
- data lineage records.

Monitoring and Incident Evidence

dashboards;

alert records;

incident tickets;

near-miss records;

suspension records;

return-to-service records;
provider communications;
remediation tracker.

Logs and Decision Trails

system logs;
input records;
output records;
tool-call logs;
human review logs;
authority logs;
downstream action logs;
audit trail exports.

Affected-Person Evidence

worker notices;
affected-person information records;
complaint records;
appeals;
data correction requests;
challenge outcomes.

Case Samples

selected AI-supported decisions;
selected AI recommendations;
selected escalations;
selected overrides;
selected refusals;
selected incidents or near misses.

The audit team should not rely only on management assertions.

Every important control claim should be linked to evidence.

The internal review should ask:

Can this claim be proven from records?

If the answer is no, the finding should say so.

30.4 Core Audit Questions

The following core questions should be used across all high-risk AI internal reviews.

System Identification

Is the AI system recorded in an internal register?

Is the provider identified?

Is the system owner identified?

Is the business owner identified?

Is the technical owner identified?

Is the governance owner identified?

Is the system version recorded?

Is the deployment environment recorded?

Is the high-risk or high-consequence classification recorded?

Are affected processes and persons identified?

Intended Purpose and Actual Use

What is the provider-defined intended purpose?

What is the organization's approved use?

What is the actual use?

Does actual use match approved use?

Does actual use match provider instructions?

Has use expanded since deployment?

Are prohibited uses defined?

Are users trained on scope limits?

Is scope drift monitored?

Has use been reassessed after change?

Instructions for Use

Are provider instructions available?

Are instructions current?

Have instructions been reviewed internally?

Have instructions been translated into workflow controls?

Are users aware of limitations?

Are data requirements implemented?

Are human oversight requirements implemented?

Are monitoring requirements implemented?

Are provider updates tracked?

Are instruction changes linked to reassessment?

Human Oversight

Who performs human oversight?

Are reviewers competent?

Are reviewers trained?

Do reviewers have authority?

Do reviewers have access to evidence?

Can reviewers override?

Can reviewers escalate?

Can reviewers stop the workflow?

Does review occur before consequence?

Is review quality monitored?

Input Data and Context

What data sources are used?

Are data sources approved?

Are data sources reliable?

Are freshness thresholds defined?

Are missing fields flagged?

Is representativeness assessed where relevant?

Is user-entered data separated from verified data?

Is context inspectable for LLM workflows?

Are policy versions controlled?

Can stale evidence trigger HOLD?

Monitoring

What is monitored?

Are governance signals monitored, not only uptime?

Are input data conditions monitored?

Is actual use monitored?

Is human oversight monitored?

Is authority drift monitored?

Are downstream effects monitored?

Are alerts assigned to owners?

Can monitoring trigger HOLD, ESCALATE or REFUSE?

Are monitoring records reviewed?

Logs and Decision Trails

What logs are generated?

Are logs under deployer control?
Are logs retained long enough?
Are logs protected from alteration?
Can logs reconstruct input, output and review?
Can logs reconstruct authority state?
Can logs reconstruct downstream consequence?
Are log gaps monitored?
Are decision trails sampled?
Can a specific case be reconstructed?

Authority and Boundaries

What authority does the AI output carry?
Is the output draft-only, advisory, decision support or executable?
Are approval thresholds mapped?
Are reviewer authorities current?
Are system permissions mapped?
Are tool permissions mapped?
Can downstream systems act on the output?
Is authority drift monitored?
Are fallback authority rules defined?
Can unauthorized use be refused?

Escalation and Incident Response

What triggers escalation?
Who receives escalations?
Are serious risk criteria defined?
Are near misses recorded?
Are provider contacts known?
Is evidence preservation defined?
Is suspension procedure defined?
Is return-to-service procedure defined?
Are incident drills performed?
Are incidents reported to governance bodies?

Workplace and Affected-Person Information

Does the AI system affect workers?
Were workers informed where required?
Were worker representatives informed where required?
Does the AI system assist decisions relating to natural persons?

Were affected persons informed where required?

Does the notice reflect actual use?

Is a challenge route available?

Are complaints monitored?

Are notices updated after scope change?

Is information evidence retained?

These questions form the baseline.

The audit team should add sector-specific questions depending on the use case.

30.5 System-Level Review

A system-level review assesses whether the AI system is governed as a system.

It asks whether the organization knows what the system is, what it is for, who owns it, what controls apply and whether it is ready for use.

System-level review should test:

system register completeness;

provider documentation;

intended purpose;

risk classification;

approved use;

ownership;

deployment environment;

model version;

third-party dependencies;

fallback models;

monitoring plan;

logging plan;

human oversight model;

incident pathway;

reassessment schedule.

The audit team should select one system record and trace it to supporting evidence.

For example:

The register says the system is used for supplier risk recommendations.

The audit should verify:

provider instructions allow this type of use;
internal approval covers supplier recommendations;
users are trained;
supplier data sources are mapped;
human review is defined;
authority thresholds are mapped;
logs are retained;
monitoring detects stale supplier data;
escalation exists for missing due diligence;
decision trails can be reconstructed.

If the system register is disconnected from actual workflow evidence, the finding should note that the register is only an inventory, not evidence of control.

The system-level review should answer:

Is the system governable?

30.6 Workflow-Level Review

A workflow-level review assesses how the AI system operates inside a specific process.

This is often where real gaps appear.

A system may look well governed at system level but behave differently in practice.

The audit should map the workflow from start to consequence.

For example:

In HR recruitment:

candidate data received;
AI summary generated;
candidate ranked;
recruiter reviews;
shortlist created;
rejection or interview decision made.

In procurement:

supplier data retrieved;
AI summary generated;
risk recommendation produced;
procurement reviewer reviews;
approval route selected;
supplier shortlisted or approved.

In compliance:

alert generated;
AI retrieves case evidence;
AI summarizes;
analyst reviews;
case escalated or closed.

For each workflow, the audit should identify:

where AI enters;
what data it uses;
what output it creates;
who sees the output;
how the output influences the process;
when human review occurs;
when consequence attaches;
what logs are generated;
what escalation triggers exist;
what happens if conditions fail.

The key workflow-level question is:

Can the organization show where AI influence becomes consequence?

If the answer is unclear, the workflow is not sufficiently mapped.

30.7 Case-Level Sampling

Case-level sampling is essential.

Policies and workflows may look strong, but case records reveal whether controls operated.

The audit team should select samples such as:

ordinary AI-supported decisions;
high-consequence decisions;
cases with human override;
cases with escalation;
cases with refusal;
cases with stale data warnings;
cases involving model fallback;
cases involving affected-person complaint;
cases before and after system change.

For each sampled case, the audit should ask:

Which AI system was used?
What was the workflow?
What input data was used?
Was input data fresh?
Was context inspectable?
What output was generated?
Was the output stored?
Did a human review it?
Did review occur before consequence?
Did the reviewer have authority?
Were warnings visible?
Were logs complete?
Did the output influence downstream consequence?
Was escalation required?
Was escalation performed?
Why was continuation allowed?
Can the full decision trail be reconstructed?

If the audit cannot reconstruct sampled cases, that is a serious evidence gap.

Case-level sampling should produce one of four conclusions:

control operated and evidence is sufficient;
control operated but evidence is incomplete;
control did not operate as designed;
control status cannot be determined.

The fourth result should not be treated as safe.

Unknown is a governance risk.

30.8 Evidence Quality Testing

The audit should test evidence quality.

A document is not automatically strong evidence.

Evidence should be assessed according to strength.

Level 0 — No Evidence

The control cannot be shown.

Level 1 — Assertion

Management says the control exists, but no operational proof is available.

Level 2 — Documented Process

A policy or procedure exists, but no case-level evidence is available.

Level 3 — Operational Record

The control operated and produced records.

Level 4 — Reconstructable Decision Trail

The system, input, output, human review, authority, monitoring and consequence can be reconstructed.

Level 5 — Runtime-Assured Evidence

The control is monitored during operation and connected to ALLOW, HOLD, ESCALATE or REFUSE states.

The audit should rate evidence for each domain.

For example:

Human oversight may be Level 2 if a procedure exists.

It may be Level 3 if review records exist.

It may be Level 4 if the audit can reconstruct what the reviewer saw, what they did and whether review occurred before consequence.

It may be Level 5 if oversight quality is monitored and weak oversight triggers action.

The audit report should avoid saying “control exists” without evidence level.

A stronger statement is:

“Human oversight is documented and operational records exist, but sampled cases did not show whether reviewers had access to source evidence. Evidence quality is Level 3, not Level 4.”

This makes the finding actionable.

30.9 Interview Guide

Interviews help test whether documented controls are understood and usable.

The audit team should interview:

system owner;
business owner;
technical owner;
human reviewers;
compliance or risk owner;
data owner;
incident response owner;
provider relationship owner;
front-line users;
affected workflow managers.

Questions for system owners:

What is the system used for?
What is it not allowed to do?
Who owns the risk?
How do you know actual use has not drifted?
What would cause you to pause use?
What incidents or near misses occurred?
What evidence proves control?

Questions for human reviewers:

What does the AI system produce?
What do you check before accepting the output?
Can you see source evidence?
Can you override the output?
Can you stop the workflow?

When do you escalate?
Are you under time pressure?
Have you disagreed with the system?
What happens when you disagree?
What training did you receive?

Questions for data owners:

What data sources feed the system?
How is data freshness checked?
What happens when data is stale?
Are missing fields flagged?
Are data source failures monitored?
Are user-entered claims verified?
Can the system continue with incomplete data?

Questions for incident owners:

What counts as an AI incident?
Who triages incidents?
How is evidence preserved?
Who contacts the provider?
Who decides suspension?
Has the process been tested?
What was learned from recent near misses?

Interviews should be compared with records.

If staff describe a process that differs from documentation, the audit should investigate.

If reviewers do not understand their authority, the oversight control may be weak.

If users are unaware of prohibited uses, intended-purpose control may be weak.

If incident owners have never tested escalation, incident readiness may be weak.

30.10 Red Flags for Internal Audit

The following red flags should receive audit attention.

AI system in production but not in register.
No clear business owner.

Provider instructions unavailable.
Actual use differs from intended purpose.
System used by teams not included in approval.
Output used for higher consequence than approved.
Human review is only an approval click.
Review occurs after consequence.
Reviewers lack training.
Reviewers cannot access source evidence.
Override rate is near zero without explanation.
Escalation rate drops suddenly.
Input data freshness unknown.
Missing data does not trigger HOLD.
Old policy versions appear in context.
LLM context cannot be reconstructed.
Logs show output but not input or review.
Decision trail cannot be reconstructed.
AI recommendations trigger downstream automation.
Fallback models used without authority rules.
Model versions change without reassessment.
Provider incident channel unknown.
No tested suspension process.
Affected persons not informed where required.
Complaints are not linked to AI governance review.
Board reporting shows adoption but not control.
Audit receives only policy evidence, no case evidence.
System continues despite unresolved monitoring alerts.
No one knows who can refuse continuation.
Grey or unknown status is treated as green.

The presence of one red flag does not always mean the system must stop.
But multiple red flags indicate that evidence of control may be weak.

30.11 Finding Severity Model

Internal audit should classify findings clearly.
A practical severity model may include four levels.

Low

A documentation or process improvement issue with limited immediate risk.

Example:

Training records exist but are stored inconsistently.

Moderate

A control weakness that could reduce governance reliability but has not yet affected high-consequence workflows.

Example:

Monitoring exists but escalation thresholds are incomplete.

High

A control weakness that may affect high-risk or high-consequence workflows.

Example:

Human reviewers cannot see source evidence before approving AI-supported recommendations.

Critical

A control failure that may allow unsupported, unauthorized, unsafe or unreconstructable AI-supported consequence.

Example:

AI-supported employment ranking is used to reject candidates without approved use, meaningful human review or reconstructable decision trail.

Severity should consider:

- consequence level;
- affected persons;
- regulatory sensitivity;
- repeatability;
- detectability;
- reversibility;
- evidence quality;
- control failure type;
- history of incidents;
- management awareness;

remediation difficulty.

The audit report should avoid vague language.

Instead of:

“Human oversight could be improved.”

Use:

“Human oversight is not currently evidenced as meaningful for the HR ranking workflow because sampled cases show reviewer approval after shortlist generation, no source evidence access record and no documented override option. Severity: High.”

This makes the issue clear.

30.12 Internal Audit Report Template

The following structure can be used for an internal review report.

1. Executive Summary

Overall conclusion.

Systems reviewed.

Key strengths.

Key gaps.

Critical findings.

Required decisions.

2. Scope and Methodology

Systems reviewed.

Workflows reviewed.

Period covered.

Evidence sources.

Interviews conducted.

Case samples.

Limitations.

3. Control Maturity Summary

Table by domain:

Domain	Rating	Evidence Quality	Key Finding
System Identification	Green	Level 4	Register complete and linked to owners
Human Oversight	Amber	Level 3	Review records exist, but source evidence access not shown
Logs	Red	Level 2	Decision trails not reconstructable
Escalation	Amber	Level 3	Procedure exists but drill not completed

4. Detailed Findings

Each finding should include:

finding title;

severity;

affected system;

affected workflow;

condition observed;

evidence reviewed;

risk;

root cause;

recommendation;

owner;

due date;

management response.

5. Case Sample Results

Summarize sampled cases.

Include:

number sampled;

number reconstructable;

number with complete human review evidence;

number with input data freshness evidence;

number with authority evidence;

number with downstream consequence trace; exceptions.

6. Red Flags

List unresolved red flags.

7. Remediation Plan

Include owners and deadlines.

8. Follow-Up Plan

Define retest date and evidence required.

The audit report should be clear enough for senior leadership to understand.

It should not bury serious control gaps in technical detail.

30.13 Remediation Tracker

Every finding should connect to remediation.

A remediation tracker should include:

Finding ID	Finding	Severity	Owner	Action	Due Date	Status	Evidence Required
AI-01	Decision trails incomplete	High	Technical Owner	Add input/output/review linkage	30 Sept	Open	Sample reconstructable cases
AI-02	HR use outside approved scope	Critical	HR Owner	Pause ranking use and reassess	Immediate	In progress	Updated use approval
AI-03	Oversight training incomplete	Moderate	Governance Owner	Complete role-specific training	15 Aug	Open	Training records

Finding ID	Finding	Severity	Owner	Action	Due Date	Status	Evidence Required
AI-04	Fallback rules missing	High	AI Platform Owner	Create fallback authority matrix	31 Aug	Open	Approved fallback registry

Remediation should not close based on promises.

It should close based on evidence.

The audit team should define what evidence is required to close each finding.

For example:

A finding about incomplete logs should close only when sample cases can be reconstructed.

A finding about human oversight should close only when reviewers are trained, authorized and review records show meaningful intervention ability.

A finding about stale data should close only when freshness thresholds are implemented and tested.

A finding about escalation should close only when the escalation pathway is tested or used successfully.

The key closure question is:

Does the remediation produce evidence of control?

If not, the finding should remain open.

30.14 Continuous Assurance Cycle

AI governance assurance should not be one-off.

AI systems change.

Workflows change.

Models change.

Data changes.

Providers change.

Users change.

Authority changes.

Downstream systems change.

Risk changes.

Internal review should therefore follow a continuous assurance cycle.

Step 1 — Identify

Maintain an up-to-date AI system inventory.

Step 2 — Classify

Identify high-risk and high-consequence systems.

Step 3 — Map

Map intended purpose, actual use, workflows, data, human oversight, authority and downstream consequence.

Step 4 — Test

Test controls through evidence review, interviews and case sampling.

Step 5 — Rate

Rate control maturity and evidence quality.

Step 6 — Remediate

Assign owners, actions and due dates.

Step 7 — Re-test

Verify remediation through evidence.

Step 8 — Report

Report to governance bodies and senior leadership.

Step 9 — Monitor

Track runtime signals, incidents and drift.

Step 10 — Reassess

Reassess after change, incident, provider update, model fallback or expansion.

This cycle turns internal review into ongoing governance.

The goal is not only to pass an audit.

The goal is to keep the organization aware of whether AI-supported workflows remain controlled.

30.15 Internal Review Checklist Summary

The following condensed checklist may be used during review planning.

Planning

Define systems in scope.

Define workflows in scope.

Identify high-risk and high-consequence uses.

Identify owners.

Select case samples.

Identify evidence sources.

Evidence Review

Review system register.

Review provider instructions.

Review intended-purpose mapping.

Review human oversight records.

Review input data controls.

Review monitoring records.

Review logs and decision trails.

Review escalation records.

Review affected-person information.

Review provider dependency records.

Interviews

Interview system owner.

Interview business owner.

Interview human reviewers.

Interview technical owner.

Interview data owner.

Interview incident owner.

Interview compliance / risk / legal where relevant.

Case Testing

Reconstruct sampled cases.

Confirm input data condition.

Confirm AI output.

Confirm human review.

Confirm authority.

Confirm downstream consequence.

Confirm escalation where relevant.

Confirm logs.

Findings

Rate severity.

Rate evidence quality.

Identify root cause.

Assign remediation.

Define closure evidence.

Report to governance body.

Track follow-up.

This summary can be used as a practical audit working paper.

Chapter 30 Summary

Internal review should test whether Article 26 deployer controls operate in practice.

The central question is not whether the organization has AI governance documents.

The central question is whether the organization can prove that an AI-supported workflow was allowed to continue under valid control conditions.

This chapter provides an Audit Questions and Internal Review Guide.

It covers audit scope, evidence sources, core audit questions, system-level review, workflow-level review, case-level sampling, evidence quality testing, interviews, red flags, finding severity, report structure, remediation tracking and continuous assurance.

The guide emphasizes that case-level sampling is essential.

A policy may look strong, but a sampled case may reveal missing input data evidence, weak human review, unclear authority, incomplete logs or uncontrolled downstream consequence.

The chapter also introduces an evidence quality scale from no evidence to runtime-assured evidence.

This helps audit teams distinguish between assertion, documentation, operational records and reconstructable decision trails.

The most important audit principle is:

Do not close a finding because a process exists.

Close it only when evidence shows that the control operates.

For Article 26 deployer governance, internal review should help the organization move from policy confidence to operational proof.

That is the difference between AI governance as documentation and AI governance as control.

Chapter 31

From Human-in-the-Loop to Human-in-Control

Legal anchor and framework boundary. [Article 14](#); [Article 26\(2\)](#). Human-in-control is the framework's stronger operational standard for meaningful oversight, not a defined legal term.

31.1 Why Human-in-the-Loop Is Not Enough

Human-in-the-loop is one of the most familiar phrases in AI governance.

It sounds reassuring.

It suggests that a person remains involved.

It suggests that AI is not acting alone.

It suggests that responsibility has not been transferred fully to the system.

But human-in-the-loop is not always enough.

A human can be in the loop and still not be in control.

The person may see the output but not the evidence.

They may approve the recommendation but not understand the model's limitations.

They may have formal responsibility but no practical authority.

They may be present after the consequence has already attached.

They may be under time pressure.

They may be encouraged to follow the system.

They may not know that data is stale.

They may not know that the model has changed.

They may not know that the output has become a downstream trigger.

They may not know that the system is being used outside its intended purpose.

In such cases, human involvement may create the appearance of control without actual control.

This is one of the central risks in deployer governance.

An organization may say:

A human reviewed the output.

A human approved the decision.

A human remains responsible.

A human can override the AI system.

But the stronger question is:

Could the human meaningfully prevent the AI-supported workflow from creating unsupported, unauthorized or unsafe consequence?

If the answer is no, then human-in-the-loop may be only procedural.

It may not be protective.

This chapter argues that Article 26 deployer governance should move from human-in-the-loop toward human-in-control.

Human-in-control means that the human has enough competence, training, authority, evidence, time, support and organizational backing to understand, challenge, delay, escalate, override or refuse the AI-supported workflow before consequence attaches.

The difference is important.

Human-in-the-loop focuses on placement.

Human-in-control focuses on power.

Human-in-the-loop asks:

Is there a human somewhere in the process?

Human-in-control asks:

Can the human change what happens?

This distinction matters across every sector discussed in this study.

In employment, a recruiter may be in the loop but still rely entirely on AI ranking.

In finance, a compliance analyst may be in the loop but lack time to inspect source evidence.

In healthcare, a clinician may be in the loop but receive an incomplete AI-generated summary.

In procurement, a manager may be in the loop but not know supplier risk data is stale.

In public services, a case worker may be in the loop but not understand how AI prioritization shaped the case path.

In agentic AI, a human may be in the loop only after the agent has already triggered downstream actions.

In each case, the issue is not the absence of a human.

The issue is the absence of meaningful control.

The future of AI governance cannot rely on human presence alone.

It must ask whether the human has leverage.

31.2 The Limits of Presence-Based Oversight

Presence-based oversight is oversight based on the fact that a human is present somewhere in the workflow.

This is the weakest form of human oversight.

It may be appropriate for low-risk AI support tasks.

But it is insufficient for high-risk or high-consequence systems.

Presence-based oversight can fail in several ways.

1. The human sees only the final output

If the reviewer sees only the AI-generated answer, score, summary or recommendation, they may not be able to evaluate whether it is supported.

They may not know:

- what input data was used;
- what sources were excluded;
- whether evidence was current;
- whether policy version was correct;
- whether uncertainty existed;
- whether contradictions were detected;
- whether the output was outside intended purpose.

This creates shallow oversight.

The human reviews the surface, not the reasoning environment.

2. The human lacks authority

A person may be asked to review an output but lack authority to stop the workflow.

They may be able to comment but not block.

They may be able to approve but not reject.

They may be able to escalate informally but not pause consequence.

They may be accountable without being empowered.

This creates responsibility without control.

3. The human acts too late

A human review after consequence may support correction, appeal or audit.

But it is not the same as preventive control.

If a candidate has already been rejected, a supplier already approved, a patient already routed, a citizen already notified or a workflow already triggered, the human review may come too late.

The strongest oversight occurs before consequence binds.

4. The human is overloaded

A human reviewer may be competent and authorized but still unable to review carefully because the volume is too high.

High throughput can convert oversight into approval theatre.

If reviewers must process hundreds of AI outputs quickly, they may not have the time to inspect evidence, question assumptions or escalate uncertainty.

5. The human is shaped by automation bias

AI outputs often appear structured, objective and confident.

Humans may over-trust them.

A recommendation may feel neutral because it comes from a system.

A score may feel precise because it is numeric.

A summary may feel complete because it is well written.

This can reduce challenge.

6. The organization rewards continuation

If management rewards speed, throughput and low friction, humans may feel pressure to approve.

If escalation is treated as delay, reviewers may avoid it.

If override is treated as failure, reviewers may stop overriding.

If the organization expects the AI system to improve efficiency, human challenge may become culturally uncomfortable.

7. The system design nudges approval

Interfaces matter.

If the approve button is easy and the reject or escalate path is hidden, oversight is biased.

If warnings are buried, the reviewer may miss them.

If source evidence requires extra clicks, reviewers may not inspect it.

If the workflow defaults to continuation, humans may simply confirm the system's path.

These limits show why human-in-the-loop is not enough.

The organization must evaluate whether oversight is meaningful in practice.

The central test is simple:

If the AI output is wrong, unsupported, stale, unauthorized or unsafe, can the human realistically detect that and prevent consequence?

If not, the oversight model is weak.

31.3 Human-in-Control Conditions

Human-in-control requires several conditions to be present together.

No single condition is enough.

A human may have competence but no authority.

They may have authority but no evidence.

They may have evidence but no time.

They may have time but no escalation route.

They may have an escalation route but no organizational support.

Human-in-control therefore requires a combination of factors.

1. Competence

The human must understand the domain, workflow, output type and consequence.

They must know what the AI system is intended to do.

They must know what it is not intended to do.

They must know the limitations.

They must know what failure looks like.

Competence must be specific to the workflow.

A general manager is not automatically competent to oversee a clinical AI output.

A technical engineer is not automatically competent to oversee an employment decision.

A recruiter is not automatically competent to assess algorithmic ranking risk.

Competence must match the decision environment.

2. Training

The human must receive system-specific training.

Training should include:

- intended purpose;
- prohibited uses;
- output interpretation;
- limitations;
- input data requirements;
- warning states;
- escalation triggers;
- override procedure;
- refusal conditions;
- automation bias;
- documentation requirements.

Generic AI awareness is not enough for high-risk oversight.

3. Authority

The human must be able to act.

They must be able to approve, modify, reject, hold, escalate or refuse.

They must be able to stop the workflow before consequence attaches.

If the human can only confirm the AI output, they are not in control.

Authority must be current and matched to the consequence level.

4. Evidence access

The human must be able to inspect enough evidence to judge the output.

They should see:

- source data;
- freshness status;
- missing data warnings;
- policy version;
- system limitations;
- uncertainty;
- contradictions;
- prior escalations;
- downstream consequence.

Without evidence access, oversight becomes trust in the AI output.

5. Timing

The human must intervene before consequence.

Pre-consequence review is control.

Post-consequence review is correction.

Both may matter, but they are not equivalent.

A human-in-control model must identify the point where AI output begins to affect people, rights, opportunities, clinical pathways, financial processes, public services or operational systems.

Human control must exist before that point.

6. Interface support

The system interface must make challenge possible.

The reviewer should not be nudged only toward approval.

Warnings should be visible.

Evidence should be accessible.

Override and escalation should be easy enough to use.

The interface should show consequence, not only output.

7. Organizational support

The organization must support human intervention.

Reviewers must not be punished for escalation.

Overrides should be treated as useful governance signals.

HOLD should not be treated as failure.

Refusal should not be treated as disobedience.

A human-in-control model requires culture as well as process.

8. Documentation

Human control must be recorded.

The organization should know:

- what the human saw;
- what they decided;
- whether they changed the output;
- whether they escalated;
- whether they refused;
- whether review occurred before consequence.

Without records, the organization cannot prove control.

9. Monitoring

Human oversight should itself be monitored.

The organization should track:

- review times;
- approval rates;
- override rates;
- escalation rates;

- refusal rates;
- reviewer workload;
- reviewer feedback;
- appeal or complaint patterns;
- cases where AI output was accepted incorrectly.

If oversight quality degrades, the system should be reassessed.

Human-in-control is therefore not a slogan.

It is a control design.

It must be built, tested, monitored and evidenced.

31.4 Control Before Consequence

The most important moment in AI governance is the moment before consequence attaches.

This is the point where an AI-supported output begins to affect the real world.

It may affect a person.

It may affect a process.

It may affect a right.

It may affect an opportunity.

It may affect a clinical pathway.

It may affect a financial decision.

It may affect a public service.

It may affect a supplier relationship.

It may affect an operational system.

This is the moment where human control matters most.

A human can review an AI output at many stages, but not every review has the same power.

A review before consequence can prevent harm.

A review after consequence can only correct, explain or compensate.

For deployer governance, the distinction is essential.

The organization should identify consequence points in each AI-supported workflow.

Examples include:

- candidate removed from shortlist;
- worker flagged for performance concern;
- supplier recommended for approval;
- customer receives a final response;
- patient triage priority assigned;
- public-service case routed;
- student assessed or allocated;
- compliance case closed;
- financial recommendation acted upon;
- automated task triggered;
- external communication sent;
- official record updated.

At each consequence point, the deployer should ask:

Is human control required here?

If yes:

Did it occur before consequence?

Did the human have evidence?

Did the human have authority?

Could the human stop continuation?

Was the review logged?

Were unresolved warnings present?

Did the workflow continue despite uncertainty?

This is where human-in-control becomes operational.

It is not enough to place a human somewhere in the workflow.

The human must be positioned at the control boundary.

The control boundary is the point where continuation becomes meaningful.

If the human is placed before the AI output, they may not review the actual result.

If the human is placed too late, they may review after impact.

If the human is placed at the right boundary, they can prevent invalid continuation.

This is the design principle:

Human control should be located where AI output becomes consequence.

For agentic AI, this may require multiple control boundaries.

An agent may retrieve data, generate a plan, call tools, create a recommendation, request approval, update a system and trigger downstream action.

A single final review may not be enough.

Human control may be needed before:

- external tool execution;
- system write;
- high-risk recommendation;
- downstream trigger;
- affected-person communication;
- final decision;
- case closure.

The more agentic the system, the more important transition-level human control becomes.

The deployer should not ask only:

Was there a human in the loop?

It should ask:

Was there a human at the point where control mattered?

31.5 Human Control and Authority Leverage

Human-in-control depends on leverage.

Leverage means the human can change the course of the workflow.

Without leverage, human oversight becomes symbolic.

There are several forms of leverage.

Decision leverage

The human can accept, reject or modify the AI output.

Timing leverage

The human can pause the workflow before consequence.

Evidence leverage

The human can request additional information or refresh stale evidence.

Escalation leverage

The human can move uncertainty to a higher authority.

Boundary leverage

The human can identify use outside intended purpose and block it.

Authority leverage

The human can determine that the current approval, role or model does not support continuation.

Technical leverage

The human can disable, suspend or restrict system use where necessary.

Organizational leverage

The human is supported by policy and culture when they challenge the AI system.

A human may have one form of leverage but not another.

For example:

A reviewer may modify text but not stop the workflow.

A recruiter may change a summary but not challenge the ranking.

A clinician may disagree clinically but not know what data the AI used.

A procurement manager may reject a recommendation but not know that supplier data is stale.

A public-service case worker may escalate a concern but not delay automated routing.

A compliance analyst may disagree with an AI summary but still face pressure to close cases quickly.

These partial forms of leverage may be insufficient.

The deployer should define what leverage is required for each workflow.

A low-risk internal draft may require limited leverage.

A high-risk employment, healthcare, finance or public-service workflow may require strong leverage.

A critical workflow may require authority to halt, escalate or suspend.

This should be explicit.

The organization should not assume that because a human is accountable, they have leverage.

Accountability without leverage is unfair and ineffective.

If the organization expects humans to control AI-supported workflows, it must give them the tools to do so.

This includes:

- clear authority;
- accessible evidence;
- visible warnings;
- manageable workload;
- escalation route;
- override protection;
- documentation support;
- management backing.

The central question is:

What can the human actually do when the AI output should not be trusted?

If the answer is “not much,” the system is not human-in-control.

31.6 Human-in-Control in Different Sectors

Human-in-control looks different across sectors.

The principle is the same, but the control design changes with context.

Financial Services

In financial services, human control may involve compliance analysts, risk officers, portfolio managers, fund administrators, internal audit or senior management.

The human must be able to challenge AI-supported outputs before they affect regulated processes, client treatment, risk classification, reporting, transaction monitoring or investment decisions.

A human-in-control financial workflow should show:

- evidence source;
- policy version;
- risk threshold;
- reviewer authority;
- escalation path;
- final decision trail.

The human should not merely accept a generated compliance summary.

They should be able to inspect the underlying case.

Employment

In employment, human control must protect against AI becoming a practical gatekeeper.

A recruiter, hiring manager or HR reviewer must be able to review candidates beyond AI ranking, inspect source evidence, consider context and prevent unfair exclusion.

A human-in-control HR workflow should show:

- AI role;
- candidate evidence;
- criteria used;
- prohibited criteria excluded;
- human review before shortlist or rejection;
- challenge route;
- decision trail.

A human signing off on an AI-generated shortlist after the shortlist has already shaped attention is not enough.

Healthcare

In healthcare, human control means clinical judgment remains active.

A clinician must understand the AI system's intended purpose, input data condition, missing-data warnings and limitations.

They must be able to disregard or override the output before it influences patient care.

A human-in-control clinical workflow should show:

- patient data used;
- missing data;
- clinical limitations;
- clinician review;
- final clinical judgment;
- escalation for unsafe or incomplete output.

Clinical fluency is not clinical authority.

Procurement

In procurement, human control means AI-supported supplier recommendations do not bypass due diligence, authority thresholds or risk review.

A procurement reviewer must be able to inspect supplier evidence, identify stale data, apply approval thresholds and escalate risk.

A human-in-control procurement workflow should show:

- supplier data condition;
- risk checks;
- due diligence status;
- reviewer authority;
- approval threshold;
- HOLD where evidence is incomplete.

A polished supplier recommendation should not override missing risk evidence.

Public Services

In public services, human control protects administrative legitimacy.

A case worker or official must be able to inspect AI-supported classification, prioritization or recommendation before it affects access, eligibility, scrutiny or treatment.

A human-in-control public-service workflow should show:

- case evidence;
- legal or policy framework;
- AI role;

human review;
affected-person information;
appeal or challenge route;
final reasoning.

The public authority must not hide behind a human signature if AI shaped the pathway invisibly.

Agentic AI

In agentic AI, human control must apply to transitions, not only final outputs.

A human may need to approve before the agent sends a message, updates a system, triggers a workflow, executes a tool or allows a recommendation to become action.

A human-in-control agentic workflow should show:

tool permissions;
transition gates;
authority state;
evidence freshness;
human approval before consequence;
logs of agent steps;
refusal conditions.

Agentic AI makes human control more complex because consequence can emerge through chains of action.

31.7 From Oversight to Intervention

Oversight is watching.

Intervention is acting.

High-risk AI governance requires both.

A human may oversee an AI system by reviewing outputs, monitoring alerts or checking summaries.

But the purpose of oversight is not merely awareness.

The purpose is the ability to intervene.

Intervention may include:

correcting an output;

rejecting a recommendation;
requesting more evidence;
refreshing input data;
escalating uncertainty;
pausing the workflow;
blocking downstream action;
refusing continuation;
suspending system use;
reviewing affected cases;
notifying responsible functions;
triggering incident response.

If no intervention is possible, oversight is weak.

This means deployers should design explicit intervention paths.

For each high-risk AI workflow, the organization should define:

What can go wrong?

Who can detect it?

Who can intervene?

What intervention is available?

How quickly must it happen?

What evidence must be preserved?

Who confirms resolution?

A useful model is:

Detect.

Decide.

Intervene.

Document.

Review.

Detect

The human or system identifies a problem.

Decide

The responsible person determines whether to allow, hold, escalate or refuse.

Intervene

The workflow is changed, paused, escalated or stopped.

Document

The action and reason are recorded.

Review

The organization assesses whether control operated and whether remediation is needed.

This turns human oversight into a control cycle.

It also prevents a common failure:

humans notice problems but nothing changes.

In high-risk AI governance, noticing is not enough.

A warning without action is not control.

A concern without escalation is not control.

A review without authority is not control.

An override option that no one can use is not control.

Human-in-control requires intervention capacity.

31.8 Evidence That the Human Was in Control

Human-in-control must be evidenced.

The organization should be able to show that the human control condition existed and operated.

Evidence may include:

oversight role assignment;

competence criteria;

training records;

authority matrix;

workflow review logs;
source evidence access logs;
warning displays;
review timestamps;
override records;
escalation records;
HOLD records;
refusal records;
reviewer comments;
downstream consequence timestamps;
oversight quality monitoring;
reviewer workload records;
audit samples;
post-incident review records.

The strongest evidence does not simply show that a human clicked approve.

It shows that the human had meaningful control.

A strong evidence record can answer:

Who reviewed the AI output?

What did they see?

What evidence was available?

Were warnings visible?

Was data fresh?

Was the output within intended purpose?

Did the reviewer have authority?

Did review occur before consequence?

Could the reviewer override?

Did they modify, approve, reject, escalate or refuse?

Why did the workflow continue?

This is evidence of human control.

A weak evidence record shows only:

Human approval: yes.

This may not be sufficient for high-risk AI governance.

The deployer should therefore design human review records to capture control, not only approval.

For example:

Instead of recording:

“Approved by reviewer.”

The system should record:

“Reviewer approved after confirming input data freshness, source evidence availability, no active escalation alerts, authority threshold satisfied, and output limited to advisory use.”

For high-consequence workflows, this level of evidence may be essential.

The final evidence question is:

Could an auditor, regulator, board member, affected person or internal reviewer understand how the human controlled the AI-supported workflow?

If not, the human-in-control claim may not be provable.

Chapter 31 Summary

Human-in-the-loop is not enough for high-risk AI governance.

A human can be present in a workflow without having meaningful control.

They may see only the final output, lack authority, act too late, face workload pressure, suffer automation bias or be nudged toward approval by interface and organizational design.

This chapter argues that deployer governance should move from human-in-the-loop to human-in-control.

Human-in-control means that the human has competence, training, authority, evidence access, timing, interface support, organizational backing, documentation and monitoring sufficient to understand, challenge, delay, escalate, override or refuse the AI-supported workflow before consequence attaches.

The chapter emphasizes the importance of control before consequence.

Human review is strongest before AI output affects a person, process, right, opportunity, clinical pathway, financial process, public service or operational system.

Post-consequence review may support correction or audit, but it is not the same as preventive control.

The chapter also introduces the idea of authority leverage.

A human must be able to change the workflow.

Accountability without leverage is not control.

The chapter applies human-in-control across financial services, employment, healthcare, procurement, public services and agentic AI.

Finally, the chapter defines evidence that the human was in control.

The organization should be able to reconstruct who reviewed the output, what evidence they saw, what authority they had, whether review occurred before consequence, what action they took and why continuation was allowed.

The central question is:

Could the human meaningfully prevent the AI-supported workflow from creating unsupported, unauthorized or unsafe consequence?

If yes, the system may be human-in-control.

If no, human-in-the-loop may be only an appearance of governance.

Chapter 32

From Compliance to Evidence Architecture

Legal anchor and framework boundary. [Articles 12-14](#); [Article 26](#). Evidence architecture is a NextOne Observatory synthesis for turning legal and policy duties into evidence-producing controls.

32.1 Why Compliance Alone Is Not Enough

Compliance is necessary.

But compliance alone is not enough.

An organization may have policies, procedures, registers, training records, vendor contracts, governance committees and audit reports. These elements matter. They show that the organization has taken AI governance seriously.

But high-risk AI governance requires more than the existence of compliance artifacts.

It requires evidence that control operated.

This distinction is central to Article 26 deployer responsibility.

A deployer must not only say that a high-risk AI system is used according to instructions. It must be able to show how those instructions were translated into operational controls.

A deployer must not only say that human oversight exists. It must be able to show that the human was competent, trained, authorized and supported enough to intervene before consequence attached.

A deployer must not only say that input data was appropriate. It must be able to show that the data was relevant, current, complete enough and suitable for the intended purpose.

A deployer must not only say that monitoring exists. It must be able to show that monitoring detected meaningful control signals and triggered action when conditions degraded.

A deployer must not only say that logs are retained. It must be able to show that logs can reconstruct the decision trail.

This is the difference between compliance and evidence architecture.

Compliance asks:

Do we have the required documents and procedures?

Evidence architecture asks:

Can we prove that the required controls operated when the AI-supported workflow became consequential?

This difference matters because AI systems often fail silently.

The system may not crash.

The workflow may not stop.

The dashboard may remain green.

The human reviewer may click approve.

The log may show completion.

But the control condition may have failed.

The data may have been stale.

The actual use may have drifted.

The human reviewer may not have had authority.

The output may have become a downstream trigger.

The fallback model may not have been approved for the same authority level.

The affected person may not have been informed where required.

A compliance file may look complete while the operational evidence is weak.

This is why deployers must move beyond compliance as documentation.

They must build compliance as evidence architecture.

32.2 Documents Do Not Prove Control

Documents are important.

But documents do not automatically prove control.

A policy shows what the organization intends.

A procedure shows how the organization expects work to happen.

A training record shows that someone completed a module.

A vendor contract shows that a provider relationship exists.

A system register shows that the organization identified the AI system.

A risk assessment shows that risks were considered.

But none of these documents alone proves that control operated at the moment of consequence.

For example:

A human oversight policy does not prove that a human reviewed the AI output before consequence.

A data governance policy does not prove that the input data was fresh.

A monitoring procedure does not prove that a warning was acted upon.

A log retention policy does not prove that a specific decision can be reconstructed.

A provider instruction file does not prove that users followed the instructions.

A board briefing does not prove that a system remained within intended purpose.

This does not make documents useless.

It means documents must be connected to operational evidence.

A mature deployer should not discard documentation.

It should strengthen documentation by linking it to workflow records, logs, review actions, monitoring signals and decisions.

For example:

A human oversight policy should link to reviewer assignments, training records, review timestamps, override logs and escalation records.

A data policy should link to input source maps, freshness checks, missing-data warnings and HOLD decisions.

A monitoring procedure should link to alerts, owners, response actions and incident records.

A logging policy should link to reconstructable decision trails.

A provider instruction should link to internal workflow controls.

This is how documents become part of evidence architecture.

The key question is:

What operational evidence proves that this document mattered?

If a policy exists but no workflow evidence shows that it operated, the policy may be a governance intention rather than a governance control.

32.3 The Evidence Chain

Evidence architecture depends on evidence chains.

An evidence chain connects the elements needed to reconstruct why an AI-supported workflow was allowed to continue.

A strong evidence chain may include:

- AI system identity;
- provider;
- system version;
- intended purpose;
- actual use;
- workflow;
- input data sources;
- input data freshness;
- context supplied;
- AI output;
- warnings;
- human review;
- reviewer authority;
- monitoring alerts;
- escalation state;
- downstream consequence;
- final decision;
- reason continuation was allowed.

The evidence chain matters because AI-supported workflows are distributed.

The system may sit in one platform.

The input data may come from another.

The human review may occur in a workflow tool.

The final decision may be recorded in a case system.

The monitoring alert may appear in a dashboard.

The provider instructions may sit in a document repository.

The incident record may sit in another system.

If these records are not connected, the organization may have evidence fragments but no evidence chain.

This becomes a problem during audit, incident response, affected-person challenge or regulator inquiry.

The organization may know that records exist somewhere.

But it may not be able to reconstruct the path from AI input to consequence.

Evidence architecture solves this by linking records around the workflow.

For each high-risk AI system, the deployer should be able to answer:

Where is the system record?

Where are the instructions for use?

Where is the approved use case?

Where is the input data record?

Where is the AI output?

Where is the human review?

Where is the authority record?

Where is the monitoring signal?

Where is the log?

Where is the downstream consequence?

Where is the final decision?

Where is the evidence that continuation was allowed?

This does not require one single system for all organizations.

But it does require a coherent evidence design.

Without that design, the organization may rely on manual reconstruction after something goes wrong.

That is fragile.

Evidence architecture should be designed before the incident.

Not after it.

32.4 Evidence Architecture as Operational Memory

Evidence architecture is operational memory.

It allows the organization to remember what happened, why it happened, who was involved, what evidence existed, which controls operated and whether continuation was justified.

This is especially important because AI-supported workflows may involve many hidden steps.

An LLM may receive context.

An agent may call tools.

A model may route between providers.

A fallback may occur.

A summary may be compacted.

A recommendation may be converted into a structured field.

A human may approve.

A downstream system may act.

Without operational memory, the organization may later see only the final result.

That is not enough.

For high-risk AI governance, operational memory should preserve:

- what the system saw;
- what the system generated;
- what the human saw;
- what the human decided;
- what authority applied;
- what warnings existed;
- what downstream effect followed;
- what monitoring signals were active;
- what evidence supported continuation.

This memory supports several functions.

It supports human oversight because reviewers can see relevant context.

It supports monitoring because patterns can be detected over time.

It supports audit because controls can be tested.

It supports incident response because events can be reconstructed.

It supports affected-person review because the pathway to outcome can be examined.

It supports board oversight because leadership can see control maturity.

It supports regulatory cooperation because the organization can provide evidence rather than only explanation.

Operational memory also protects the organization from institutional forgetting.

People leave roles.

Systems change.

Policies update.

Providers change.

Model versions change.

Workflows are redesigned.

If evidence is not preserved, the organization may lose the ability to understand why a prior AI-supported decision was made.

This matters because the consequences of AI-supported workflows may appear later.

A procurement decision may create supplier risk months later.

An employment screening process may be challenged after candidates are rejected.

A clinical summary may be reviewed after a patient safety concern.

A public-service decision may be appealed.

A financial compliance case may be audited.

At that point, the organization needs operational memory.

It needs evidence of control.

32.5 From Checklist Thinking to Control Thinking

Many organizations begin AI governance with checklists.

Checklists are useful.

They help ensure that important topics are not forgotten.

They create structure.

They support consistency.

They help teams begin.

But checklist thinking has limits.

A checklist can create the appearance of readiness if each item is marked complete.

The risk is that the organization focuses on completion rather than control.

For example:

“Human oversight assigned” may be checked.

But is the human competent?

“Training completed” may be checked.

But was the training system-specific?

“Logs retained” may be checked.

But can they reconstruct the decision?

“Monitoring active” may be checked.

But does it trigger action?

“Provider instructions stored” may be checked.

But were they translated into workflow controls?

“Input data reviewed” may be checked.

But was data fresh at the point of consequence?

Control thinking asks deeper questions.

It asks:

What control is this checklist item supposed to create?

Where does that control operate?

Who owns it?

What evidence does it produce?

What happens when it fails?

Can the workflow continue without it?

Is the control still valid under current conditions?

This transforms the checklist into an evidence architecture.

The goal is not to abandon checklists.

The goal is to make checklists evidence-producing.

Each checklist item should link to:

- control objective;
- operational implementation;
- evidence record;
- owner;
- failure state;
- reassessment trigger.

For example:

Checklist item: human oversight.

Control objective: prevent unsupported AI output from creating consequence.

Operational implementation: authorized reviewer must review output and source evidence before final decision.

Evidence record: review timestamp, reviewer role, evidence shown, action taken, override/escalation record.

Failure state: HOLD if reviewer unavailable or review occurs after consequence.

Reassessment trigger: review backlog, low override rate, workflow expansion.

This is control thinking.

It changes the question from:

Did we complete the checklist?

To:

Did the control operate, and can we prove it?

That is the maturity shift Article 26 requires.

32.6 Evidence Architecture Components

A deployer evidence architecture should contain several core components.

1. AI System Register

The register identifies AI systems, owners, providers, intended purpose, actual use, risk classification and affected workflows.

It is the map of what exists.

2. Use-Case Approval Record

This record defines what the system is approved to do internally.

It should include scope, user groups, output role, consequence level, prohibited uses and reassessment triggers.

3. Instructions-to-Control Mapping

This maps provider instructions and limitations to internal procedures, workflow controls, training and monitoring.

It shows how instructions became operational.

4. Input Data Record

This records source, freshness, completeness, representativeness where relevant, missing data and warnings.

It shows whether the system had suitable inputs.

5. Context Record

For LLM and agentic systems, this records what context was supplied, what sources were retrieved, whether context was compacted and whether governance-critical context was preserved.

It shows what the system was allowed to rely on.

6. Output Record

This stores or references the AI output and classifies it as draft, summary, score, recommendation, ranking, decision support, tool call or workflow trigger.

It shows what the system produced.

7. Human Oversight Record

This records who reviewed, what evidence they saw, what authority they had, what action they took and whether review occurred before consequence.

It shows whether human control operated.

8. Authority Record

This records whether the workflow, user, reviewer, model and system had authority for the consequence level.

It shows whether continuation was authorized.

9. Monitoring Record

This records runtime signals, alerts, drift indicators, data issues, oversight issues, fallback events and downstream changes.

It shows whether the system remained under observation.

10. Escalation Record

This records HOLD, ESCALATE, REFUSE, incident response, provider contact, suspension and return-to-service decisions.

It shows whether risk triggered action.

11. Decision Trail

This connects system, input, output, human review, authority, monitoring, escalation and consequence.

It shows why continuation was allowed.

12. Reassessment Record

This records periodic and event-based reassessment.

It shows whether control conditions remained valid over time.

Together, these components form evidence architecture.

The architecture does not need to be perfect on day one.

But it must be intentional.

The organization should know what evidence it needs, where it comes from, who owns it and how it connects.

32.7 Evidence Architecture and Runtime Governance

Evidence architecture supports runtime governance.

Runtime governance asks whether current control conditions support continuation.

Evidence architecture provides the information needed to answer that question.

For example:

Runtime governance asks:

Is input data fresh?

Evidence architecture provides freshness records.

Runtime governance asks:

Is human oversight available?

Evidence architecture provides reviewer assignment, workload and review records.

Runtime governance asks:

Is the output within intended purpose?

Evidence architecture provides intended-purpose and actual-use mapping.

Runtime governance asks:

Is authority valid?

Evidence architecture provides approval thresholds and authority records.

Runtime governance asks:

Did fallback occur?

Evidence architecture provides model dependency and fallback records.

Runtime governance asks:

Can the decision be reconstructed?

Evidence architecture provides logs and decision trails.

Without evidence architecture, runtime governance becomes guesswork.

The system may detect a signal, but the organization may not know whether it matters.

The human reviewer may see a warning, but not the evidence behind it.

The governance owner may see an incident, but not the path to consequence.

The board may see a dashboard, but not evidence quality.

Evidence architecture makes runtime governance actionable.

It allows the organization to apply the four core states:

ALLOW.

HOLD.

ESCALATE.

REFUSE.

ALLOW requires evidence that control conditions are satisfied.

HOLD requires evidence that something is missing, stale, uncertain or degraded.

ESCALATE requires evidence that higher authority or specialist review is needed.

REFUSE requires evidence that continuation is not allowed under current conditions.

The quality of these decisions depends on evidence.

A system cannot responsibly HOLD if it cannot identify what is missing.

It cannot responsibly ESCALATE if it cannot identify the trigger.

It cannot responsibly REFUSE if it cannot explain the boundary.

It cannot responsibly ALLOW if it cannot show valid conditions.

This is why evidence architecture and runtime governance must be designed together.

32.8 Evidence Architecture and Accountability

Accountability requires more than assigning responsibility.

A named person or function may be accountable.

But accountability is weak if that person cannot access evidence.

A board may be accountable for oversight.

But board oversight is weak if leadership receives only high-level assurance.

A human reviewer may be accountable for a decision.

But reviewer accountability is unfair if the reviewer could not see source evidence or stop the workflow.

A deployer may be accountable for use.

But deployer accountability is weak if provider dependency, model fallback or downstream automation is invisible.

Evidence architecture supports real accountability by making responsibility inspectable.

It helps answer:

Who owned the system?

Who approved use?

Who reviewed the output?

Who had authority?

Who saw the warning?

Who escalated?

Who allowed continuation?

Who decided to suspend or resume?

Who is responsible for remediation?

This is not about blame.

It is about reconstructability.

In complex AI-supported workflows, accountability can blur.

The provider built the system.

The deployer configured it.

A business team used it.

A human reviewed it.

An agent called tools.

A downstream system acted.

An affected person experienced the consequence.

Without evidence architecture, the organization may struggle to understand where responsibility sat at each stage.

Evidence architecture does not solve every legal or ethical question.

But it gives the organization a factual basis for accountability.

It prevents governance from becoming a chain of assumptions.

32.9 Evidence Architecture and Trust

Trust in high-risk AI cannot be built only on performance claims.

Performance matters.

Accuracy matters.

Reliability matters.

But trust also depends on control.

A system may be accurate on average but still unsafe in specific conditions.

A system may perform well in testing but drift in operation.

A system may generate helpful outputs but be used outside intended purpose.

A system may appear reliable but lack human intervention capacity.

A system may be efficient but unreconstructable.

Evidence architecture supports trust by showing that the organization can answer hard questions.

Can we show what happened?

Can we show why it happened?

Can we show who reviewed it?

Can we show what evidence was used?

Can we show whether data was fresh?

Can we show whether authority was valid?

Can we show whether the system remained within scope?

Can we show when we paused?

Can we show when we refused?

Can we show how we remediated?

This is the foundation of institutional trust.

Not trust as belief.

Trust as evidence.

Not trust because the system sounds intelligent.

Trust because the organization can prove control.

This is especially important for affected persons.

A candidate, worker, patient, citizen, customer, student or supplier may not trust a process simply because the organization says AI was governed.

Trust requires the possibility of explanation, review, correction and accountability.

Evidence architecture makes those possibilities real.

32.10 Building Evidence Architecture Step by Step

Organizations do not need to build perfect evidence architecture immediately.

But they should begin deliberately.

A practical implementation path may follow ten steps.

Step 1 — Identify High-Risk and High-Consequence AI Systems

Start with systems that create the greatest consequence.

Do not attempt to solve all AI governance equally at once.

Step 2 — Map Workflows

Identify where AI enters the workflow, what output it produces, who sees it and where consequence attaches.

Step 3 — Define Control Conditions

For each workflow, define what must be true for continuation:

- intended purpose match;
- input data validity;
- human oversight;

- authority;
- monitoring;
- logs;
- escalation;
- affected-person information.

Step 4 — Identify Evidence Sources

Determine where evidence currently exists.

System register.

Workflow tool.

Data system.

Review record.

Monitoring dashboard.

Log store.

Incident system.

Provider documentation.

Step 5 — Find Evidence Gaps

Ask whether a sampled case can be reconstructed.

If not, identify what is missing.

Step 6 — Design Evidence Links

Connect records through IDs, timestamps, workflow references or case references.

The goal is not duplication.

The goal is traceability.

Step 7 — Create Control States

Define ALLOW, HOLD, ESCALATE and REFUSE conditions.

Step 8 — Train Owners and Reviewers

Evidence architecture depends on people understanding what must be recorded and why.

Step 9 — Test With Cases

Run real or simulated cases.

Can the organization reconstruct the path?

Can it explain continuation?

Can it identify when HOLD should occur?

Step 10 — Report and Reassess

Report evidence maturity to governance bodies and reassess after changes.

This step-by-step approach makes evidence architecture achievable.

The organization should begin where consequence is highest.

Over time, the architecture can mature.

32.11 Minimum Evidence Architecture for Article 26

A minimum Article 26 evidence architecture should include the following.

System-Level Evidence

- AI system register;
- provider instructions;
- intended purpose;
- approved internal use;
- system owner;
- risk classification;
- affected workflow;
- affected persons or groups;
- model or system version;
- provider dependency record.

Workflow-Level Evidence

- workflow map;
- output role;
- consequence point;
- human oversight step;
- authority threshold;
- input data source;

- monitoring signal;
- escalation path;
- downstream system map.

Case-Level Evidence

- input data condition;
- AI output;
- context record where relevant;
- human review;
- authority validation;
- monitoring alerts;
- downstream consequence;
- final decision state.

Control-State Evidence

- ALLOW record;
- HOLD record;
- ESCALATE record;
- REFUSE record;
- reason for state;
- owner;
- timestamp;
- resolution.

Reassessment Evidence

review date;

trigger;

findings;

remediation;

approval;

next review.

This minimum architecture supports the central deployer question:

Was this AI-supported workflow allowed to continue under valid control conditions?

Without this architecture, the organization may be unable to prove Article 26 operational readiness.

32.12 The Strategic Shift

The strategic shift is simple but important.

AI governance must move:

from policy to proof;

from assertion to reconstruction;

from documentation to control;

from human presence to human leverage;

from monitoring dashboards to pause conditions;

from logs to decision trails;

from compliance files to evidence architecture;

from post-event explanation to pre-consequence admissibility.

This does not mean compliance teams become engineers.

It means compliance, risk, legal, technology, data and business teams must work from a shared evidence model.

Each function owns part of the architecture.

Legal may interpret obligations.

Compliance may define control expectations.

Risk may assess consequence.

Technology may implement logs and monitoring.

Data teams may maintain input data controls.

Business teams may own workflows.

Human reviewers may operate oversight.

Internal audit may test evidence.

The board may oversee readiness.

Evidence architecture connects them.

It gives the organization a common language:

What system?

What use?

What evidence?

What control?

What consequence?

What state?

What decision?

This is the language of operational AI governance.

Chapter 32 Summary

Compliance is necessary, but compliance alone is not enough.

A deployer may have policies, procedures, registers, training records and vendor documents, but still lack evidence that control operated when an AI-supported workflow became consequential.

This chapter argues that Article 26 deployer governance requires a shift from compliance documentation to evidence architecture.

Evidence architecture connects system identity, intended purpose, actual use, provider instructions, input data, context, AI output, human oversight, authority, monitoring, escalation, downstream consequence and final decision state.

The chapter distinguishes documents from control.

A policy does not prove that oversight operated.

A log retention policy does not prove that a decision trail is reconstructable.

A provider instruction file does not prove that users followed instructions.

Documents become valuable when they are connected to operational evidence.

The chapter introduces evidence chains, operational memory and the move from checklist thinking to control thinking.

Checklist items should not simply be marked complete. They should identify the control objective, evidence produced, owner, failure state and reassessment trigger.

Evidence architecture supports runtime governance by giving the organization the information needed to decide whether a workflow should ALLOW, HOLD, ESCALATE or REFUSE.

It also supports accountability and trust by making AI-supported workflows reconstructable.

The strategic shift is:

from policy to proof;

from assertion to reconstruction;

from documentation to control;

from logs to decision trails;

from compliance files to evidence architecture.

The central question remains:

Can the organization prove that the AI-supported workflow was allowed to continue under valid control conditions?

That is the practical foundation of Article 26 deployer governance.

Chapter 33

Article 26 as Runtime Governance

Legal anchor and framework boundary. [Article 26](#). Article 26 as runtime governance is the central interpretive thesis of this study, not an official title or statutory characterisation.

33.1 The Central Argument

The central argument of this study is that Article 26 should not be understood only as a compliance checklist.

It should be understood as a runtime governance model.

This distinction matters.

A checklist asks whether the deployer has addressed a set of obligations.

Runtime governance asks whether the AI-supported workflow remains under valid control conditions while it is being used.

A checklist may ask:

Are provider instructions available?

Is human oversight assigned?

Is input data considered?

Are logs retained?

Is monitoring in place?

Is there an incident process?

Runtime governance asks:

Is the system being used according to instructions right now?

Is human oversight meaningful at the point of consequence?

Is the input data fresh and suitable for this workflow?

Are logs sufficient to reconstruct this decision?

Is monitoring detecting the right control signals?

Is escalation triggered when conditions degrade?

Should the workflow continue, hold, escalate or refuse?

This is a different level of governance.

It moves Article 26 from policy implementation into operational control.

The Article 26 deployer is not only expected to own documents.

The deployer must operate the system responsibly.

That means the deployer must understand the system's role, its actual use, its input conditions, its human oversight model, its monitoring signals, its logs, its escalation pathways and its effect on real workflows.

The deployer must also be able to prove those controls.

This is why Article 26 becomes runtime governance.

The obligation is not satisfied only at the moment of deployment.

It continues during operation.

A high-risk AI system can be properly deployed and later become improperly used.

A system can be approved for one workflow and later used in another.

A human oversight model can be designed well and later degrade under workload pressure.

Input data can be valid at one point and stale later.

A model can be replaced by a fallback model.

A recommendation can become a downstream trigger.

A policy can change.

A reviewer's authority can expire.

A data source can stop refreshing.

A workflow can expand.

A system can continue to operate while the control assumptions around it have changed.

This is the core runtime problem.

Article 26 should therefore be read as requiring continued validity.

The deployer must not only ask:

Was this AI system approved?

The deployer must ask:

Is this AI-supported workflow still allowed to continue under current conditions?

That question sits at the heart of operational AI governance.

33.2 From Deployment Approval to Continued Validity

Deployment approval is a point in time.

Continued validity is a condition over time.

This is one of the most important distinctions in AI governance.

An organization may conduct a risk assessment before deployment.

It may review provider documentation.

It may approve a use case.

It may define human oversight.

It may configure access controls.

It may train users.

It may create logs and monitoring.

It may document an incident process.

These steps are necessary.

But they do not guarantee that the system remains valid during use.

The conditions that supported deployment can change.

The system may be used by new teams.

The output may influence new decisions.

The workflow may become more automated.

The input data may become stale.

The provider may update instructions.

The model version may change.

A fallback model may be introduced.

Human reviewers may become overloaded.

Downstream systems may start acting on the output.

The consequence level may increase.

This means deployer governance must move from approval to reassessment.

Approval says:

The system may be used under these conditions.

Runtime governance asks:

Are those conditions still present?

This is continued validity.

Continued validity should be assessed through runtime signals and reassessment triggers.

Useful triggers include:

- actual use changes;
- new user group;
- new affected-person group;
- new data source;
- stale evidence;
- provider instruction update;
- model version change;
- fallback model use;
- downstream integration change;
- human oversight degradation;
- authority change;
- incident or near miss;
- complaint or appeal;
- monitoring alert;
- output becoming more influential;
- scope expansion;
- policy change;
- regulatory change.

When these triggers occur, the deployer should not assume that the original approval still applies.

The organization should reassess.

The reassessment may conclude that the system can continue.

It may impose conditions.

It may reduce authority.

It may require additional human review.

It may require evidence refresh.

It may require escalation.

It may suspend or refuse use.

This is runtime governance in practice.

The key principle is:

Approval is not permanent permission.

Approval is permission under defined conditions.

When the conditions change, permission must be reassessed.

33.3 Article 26 as an Operating Model

Article 26 can be translated into an operating model.

This operating model has several connected parts.

1. Use According to Instructions

The deployer must ensure that the system is used according to instructions for use.

Operationally, this means:

- provider instructions are reviewed;
- limitations are understood;
- internal procedures reflect those instructions;
- users are trained;
- actual use is monitored;
- use outside intended purpose triggers reassessment;
- prohibited use triggers refusal or escalation.

The runtime governance question is:

Is the current workflow still within the system's intended and approved use?

2. Human Oversight

The deployer must assign human oversight to persons with competence, training, authority and support.

Operationally, this means:

- reviewers are identified;
- reviewers are trained;
- reviewers can access evidence;
- reviewers can override;
- reviewers can hold, escalate or refuse;
- review occurs before consequence;
- oversight quality is monitored.

The runtime governance question is:

Can a competent human meaningfully intervene before the AI output creates consequence?

3. Input Data Responsibility

Where input data is under the deployer's control, it must be relevant and sufficiently representative in view of the intended purpose.

Operationally, this means:

- data sources are mapped;
- freshness thresholds are defined;
- missing data is flagged;
- stale evidence can trigger HOLD;
- user-entered data is distinguished from verified data;
- data quality is monitored.

The runtime governance question is:

Does the input data currently support the authority attached to the output?

4. Monitoring

The deployer must monitor operation on the basis of instructions for use.

Operationally, this means:

- technical health is monitored;
- governance signals are monitored;

- actual use is monitored;
- human oversight is monitored;
- input data condition is monitored;
- authority drift is monitored;
- downstream effects are monitored;
- monitoring connects to action.

The runtime governance question is:

Can the organization detect when continued use is no longer justified?

5. Log Retention and Decision Trails

The deployer must retain logs where required and where under its control.

Operationally, this means:

- logs identify the system;
- logs preserve input, output, review and consequence;
- logs support reconstruction;
- log integrity is protected;
- decision trails can be sampled and audited.

The runtime governance question is:

Can the organization reconstruct why the AI-supported workflow was allowed to continue?

6. Risk and Incident Response

The deployer must act where risks, malfunction or incidents arise, including informing relevant parties where required and suspending use where appropriate.

Operationally, this means:

- incident pathways exist;
- serious risk criteria are defined;
- evidence preservation is ready;
- provider contact is known;
- suspension and return-to-service procedures exist;
- near misses are reviewed;
- remediation is tracked.

The runtime governance question is:

When something goes wrong, can the organization detect, stop, preserve, escalate and correct?

7. Information Duties

Where required, workers, representatives or affected natural persons must be informed.

Operationally, this means:

- information duties are assessed;
- notices are specific enough;
- notices reflect actual use;
- affected persons know AI's role where required;
- challenge routes exist;
- notices are updated when use changes.

The runtime governance question is:

Do affected people understand enough about AI involvement to make oversight, challenge or accountability meaningful?

Together, these elements form an operating model.

Article 26 is not only a legal obligation.

It becomes an architecture of controlled use.

33.4 Runtime Governance States

A runtime governance model needs clear states.

Throughout this study, four states have been used:

ALLOW.

HOLD.

ESCALATE.

REFUSE.

These states turn governance into action.

Without them, governance can remain descriptive.

A dashboard may show risk.

A policy may state requirements.

A log may record events.

A human may see an output.

But the workflow may still continue automatically.

Runtime states prevent this.

ALLOW

ALLOW means current control conditions support continuation.

The system is used within intended purpose.

Input data is suitable.

Human oversight is satisfied where required.

Authority is valid.

Monitoring shows no unresolved critical alerts.

Logs support reconstruction.

Downstream consequence is controlled.

ALLOW should not mean “nothing went wrong.”

It means the organization has enough evidence to justify continuation.

HOLD

HOLD means continuation is paused because required conditions are missing, incomplete, stale, uncertain or degraded.

HOLD is not failure.

It is controlled uncertainty.

A workflow may enter HOLD because:

input data is stale;

reviewer is unavailable;

authority is unclear;

context is incomplete;

logs are missing;

fallback model requires reassessment;

downstream effect is unknown;

policy version conflict exists.

HOLD gives the organization time to restore validity.

ESCALATE

ESCALATE means the issue requires higher authority, specialist review or governance decision.

Escalation may be required for:

- use outside intended purpose;
- unresolved monitoring alert;
- policy conflict;
- serious risk signal;
- model substitution;
- high-consequence override;
- affected-person challenge;
- incident candidate;
- authority boundary issue.

ESCALATE prevents front-line users or automated systems from carrying decisions they are not authorized to resolve.

REFUSE

REFUSE means the workflow should not proceed under current conditions.

Refusal may be required when:

- use is prohibited;
- evidence cannot support the output;
- human oversight is impossible;
- authority is absent;
- serious risk remains unresolved;
- logs are insufficient for required accountability;
- provider instructions do not support the use;
- fallback model is not approved;
- downstream consequence cannot be controlled.

REFUSE is a governance feature.

A system that can refuse unsafe continuation is more mature than one that always produces an answer.

These four states make Article 26 operational.

They allow the deployer to ask at each critical point:

Can this workflow continue?

Should it pause?

Should it escalate?

Should it refuse?

This is runtime governance.

33.5 The Runtime Control Loop

Article 26 as runtime governance can be expressed as a control loop.

The loop has seven stages.

Stage 1 — Define

The deployer defines intended purpose, approved use, user groups, output role, consequence level, human oversight, data requirements, monitoring signals, logs and escalation rules.

Stage 2 — Observe

The deployer observes the system in operation.

This includes system health, input data, context, human oversight, actual use, authority, downstream effects, model fallback and incident signals.

Stage 3 — Compare

The deployer compares current conditions with approved control conditions.

Is use within scope?

Is evidence fresh?

Is oversight available?

Is authority valid?

Are logs complete?

Is downstream consequence controlled?

Stage 4 — Decide

The deployer determines the governance state:

ALLOW.

HOLD.

ESCALATE.

REFUSE.

Stage 5 — Act

The workflow continues, pauses, escalates, refuses, suspends, refreshes evidence, requests human review, contacts the provider or triggers incident response.

Stage 6 — Record

The deployer records the decision and evidence.

This creates the decision trail.

Stage 7 — Reassess

The deployer reviews patterns, incidents, drift, complaints, fallback events, oversight quality and control failures.

The system, workflow or governance model is updated where needed.

This control loop is continuous.

It does not end after deployment.

It does not end after the first audit.

It does not end after the first successful use.

It continues for as long as the high-risk AI system remains in use.

This is how Article 26 becomes operational.

33.6 Runtime Governance and Agentic AI

Runtime governance becomes even more important for agentic AI.

Agentic AI systems may not simply produce outputs.

They may plan, retrieve evidence, call tools, route workflows, update systems, trigger actions and interact with downstream services.

This creates multiple transition points.

Each transition may require control.

For example:

prompt to plan;

plan to tool call;

tool call to retrieved evidence;

evidence to recommendation;

recommendation to approval request;

approval to system update;

system update to external consequence.

A final human review may not be enough.

The agent may create operational effect before final review occurs.

Runtime governance must therefore operate across the chain.

For agentic AI, Article 26-style governance should include:

tool permission mapping;

transition-level logging;

context inspection;

evidence freshness checks;

model dependency monitoring;

fallback authority reduction;

downstream effect mapping;

human approval before high-consequence transitions;

refusal logic;

incident pathways for autonomous actions.

The runtime governance question becomes:

Can the deployer prove that each relevant agentic transition remained within valid control conditions before consequence attached?

This is harder than governing a single output.

But it is necessary.

Agentic AI makes hidden authority drift easier.

A recommendation may become a task.

A task may become a system update.

A system update may become a customer communication.

A customer communication may become a legal or operational consequence.

The deployer must know where that chain begins and where control is required.

Runtime governance is the answer to that problem.

33.7 Runtime Governance and Evidence Architecture

Runtime governance depends on evidence architecture.

A system cannot decide whether continuation is allowed unless it has evidence about current conditions.

The evidence architecture provides:

- system identity;
- approved use;
- provider instructions;
- input data condition;
- context record;
- AI output record;
- human oversight record;
- authority record;
- monitoring record;
- escalation record;
- downstream consequence record;
- final decision state.

Runtime governance uses this evidence to decide whether the workflow should continue.

Evidence architecture answers:

What is true?

Runtime governance asks:

What should happen next?

The two must work together.

Without evidence architecture, runtime governance becomes unsupported judgment.
Without runtime governance, evidence architecture becomes passive documentation.
Together, they create evidence of control.
This is the central maturity shift.
The organization does not merely store records.
It uses records to govern continuation.
It does not merely monitor signals.
It converts signals into ALLOW, HOLD, ESCALATE and REFUSE decisions.
It does not merely assign human oversight.
It gives humans evidence and authority.
It does not merely retain logs.
It builds decision trails.
This is Article 26 as an operating system for deployer governance.

33.8 The Role of Boards and Senior Management

Runtime governance also changes what boards and senior management need to know.

Leadership should not only ask:

Are we compliant?

It should ask:

Which AI-supported workflows are currently allowed to continue?

Which are in HOLD?

Which require escalation?

Which have been refused?

Which systems have unresolved control gaps?

Which high-risk AI systems cannot produce reconstructable decision trails?

Which workflows rely on stale evidence?

Which systems have weak human oversight?

Which systems are using fallback models?

Which AI outputs are becoming downstream triggers?

Which affected-person complaints reveal governance problems?

This gives leadership decision-grade visibility.

A board does not need to inspect every log.

But it needs to know whether the organization has runtime control.

Leadership should receive:

system inventory;

high-risk and high-consequence map;

control maturity;

unresolved alerts;

systems in HOLD or REFUSE;

incidents and near misses;

decision-trail maturity;

human oversight quality;

provider dependency risk;

remediation status;

executive decisions required.

This moves board oversight from general reassurance to operational accountability.

The board should not be told only:

AI governance is in place.

It should be shown:

These systems are controlled.

These systems are under remediation.

These systems are paused.

These systems have incomplete evidence.

These systems require decision.

This is how runtime governance becomes visible at leadership level.

33.9 Runtime Governance and Trustworthy AI

Trustworthy AI is often discussed in terms of ethics, fairness, transparency, accountability and safety.

These principles matter.

But principles become real only when they shape operation.

Runtime governance is the bridge.

Fairness becomes real when input data, outcome patterns, affected-person challenges and human review are monitored.

Transparency becomes real when AI involvement, system role and decision trails are visible.

Accountability becomes real when authority, ownership and evidence are reconstructable.

Safety becomes real when monitoring can trigger HOLD, ESCALATE, REFUSE or suspension.

Human oversight becomes real when humans have competence, authority, evidence and time.

Reliability becomes real when the system is monitored during use, not only tested before deployment.

Trustworthy AI therefore requires runtime signals.

It requires the ability to know when trust should be limited.

This is an important point.

Trustworthy AI does not mean trusting the system more.

It means knowing when the system is worthy of trust, and when it is not.

A mature deployer should be able to say:

This output can be used.

This output requires review.

This workflow must hold.

This use is outside scope.

This evidence is stale.

This human review was insufficient.

This fallback model reduces authority.

This decision cannot be reconstructed.

This system should be suspended.

That is not weakness.

That is trust discipline.

Runtime governance makes trust conditional on evidence.

That is exactly what high-risk AI requires.

33.10 Runtime Governance as Competitive Advantage

Runtime governance should not be seen only as regulatory burden.

It can become a strategic advantage.

Organizations that build runtime governance will be better positioned to:

deploy AI safely;

scale AI responsibly;

respond to incidents quickly;

satisfy audits;

support regulator inquiries;

build customer and public trust;

manage provider dependency;

reduce hidden operational risk;

govern agentic AI;

protect affected persons;

support board confidence;

differentiate in regulated markets.

Many organizations will adopt AI quickly.

Fewer will be able to prove control.

That difference may become commercially and institutionally important.

In financial services, evidence of control may support supervisory confidence.

In healthcare, it may support patient safety and clinical trust.

In employment, it may support fairness and contestability.

In public services, it may support legitimacy.

In procurement, it may support supplier risk management.

In agentic enterprise workflows, it may support operational resilience.

The organizations that treat Article 26 only as paperwork may experience it as friction.

The organizations that translate it into runtime governance may turn it into stronger architecture.

They will know where AI is used.

They will know what it affects.

They will know who controls it.

They will know when it should stop.

They will know how to prove what happened.

That is a trust advantage.

It is also an operational advantage.

33.11 Article 26 in One Sentence

Article 26 can be summarized operationally in one sentence:

A deployer should be able to prove that a high-risk AI system was used under valid, monitored, human-controllable and reconstructable conditions when its output influenced consequence.

This sentence contains the core ideas.

Used under valid conditions means the system remained within intended purpose, provider instructions, approved scope, suitable data and valid authority.

Monitored means the organization could detect degradation, drift, malfunction, oversight weakness, stale evidence, model fallback or downstream change.

Human-controllable means competent, trained, authorized and supported humans could intervene before consequence attached.

Reconstructable means logs and evidence chains can show what happened, why it happened, who reviewed it, what authority applied and why continuation was allowed.

When output influenced consequence means governance focuses on the point where AI begins to matter.

This sentence can serve as the executive interpretation of the entire study.

It is not a replacement for the legal text.

It is an operational translation.

It tells the deployer what Article 26 demands in practice:

not simply AI use,

but controlled AI use;

not simply human presence,

but human leverage;

not simply logs,

but decision trails;

not simply monitoring,

but pause conditions;

not simply compliance,

but evidence of control.

Chapter 33 Summary

Article 26 should be understood as runtime governance.

It is not only a deployer compliance checklist.

It is an operating model for ensuring that high-risk AI systems remain under valid control conditions during use.

This chapter argues that deployer governance must move from deployment approval to continued validity.

A system may be approved at deployment, but the conditions that supported approval can change. Actual use can drift, input data can become stale, human

oversight can weaken, model fallback can occur, downstream systems can attach consequence, and authority can become unclear.

Article 26 becomes operational when its obligations are translated into runtime control questions:

Is the system being used according to instructions?

Is actual use within intended purpose?

Is input data suitable and fresh?

Is human oversight meaningful before consequence?

Are logs sufficient to reconstruct the decision trail?

Is monitoring detecting governance conditions?

Are escalation and suspension pathways ready?

Are affected persons informed where required?

The chapter presents Article 26 as an operating model built around ALLOW, HOLD, ESCALATE and REFUSE states.

These states make governance actionable.

They prevent uncertainty from being silently absorbed into continuation.

The chapter also introduces the runtime control loop:

Define.

Observe.

Compare.

Decide.

Act.

Record.

Reassess.

This loop turns deployer governance into an ongoing process rather than a one-time approval.

For agentic AI, runtime governance becomes even more important because AI systems may execute multi-step chains, call tools and influence downstream systems.

Each relevant transition must remain within valid control conditions before consequence attaches.

The chapter concludes that runtime governance depends on evidence architecture.

Evidence architecture provides the records.

Runtime governance uses them to decide whether continuation is allowed.

The central operational sentence is:

A deployer should be able to prove that a high-risk AI system was used under valid, monitored, human-controllable and reconstructable conditions when its output influenced consequence.

That is Article 26 as runtime governance.

Chapter 34

Final Conclusions and Future Outlook

Legal anchor and framework boundary. [Article 26](#); [Article 96](#). Conclusions are bounded by the official text and future Commission guidance; readers should verify later amendments and final guidance.

34.1 The Core Conclusion

The core conclusion of this study is simple:

Article 26 is not only about using high-risk AI systems responsibly.

It is about proving that responsible use continued to exist when the AI system affected real workflows, real people, real decisions and real consequences.

This is the central shift.

The deployer is not merely the organization that purchases, accesses or operates an AI system.

The deployer is the organization that places the system into a human, legal, operational, technical and institutional environment.

That environment determines whether the system remains controlled.

The same AI system may be low-risk in one workflow and high-consequence in another.

The same output may be harmless as a draft and powerful as a decision trigger.

The same recommendation may be advisory in one context and practically binding in another.

The same model may be acceptable for internal support but insufficient for regulated decisions.

The same human review step may be meaningful in one workflow and symbolic in another.

This is why deployer governance cannot be reduced to system procurement.

It must follow the AI system into operation.

Throughout this study, Article 26 has been interpreted as an operational governance obligation built around several connected questions:

Is the system being used according to instructions?

Is actual use still within intended purpose?

Is input data suitable, current and relevant?

Is human oversight meaningful before consequence attaches?

Is authority valid?

Are logs sufficient to reconstruct the decision trail?

Is monitoring capable of detecting degraded control conditions?

Can the workflow pause, escalate or refuse?

Can the organization prove why continuation was allowed?

These questions form the practical foundation of deployer responsibility.

They also define the difference between AI governance as documentation and AI governance as control.

34.2 The Main Thesis Restated

The main thesis of this study can be restated as follows:

A deployer should be able to prove that a high-risk AI system was used under valid, monitored, human-controllable and reconstructable conditions when its output influenced consequence.

Each word matters.

Valid means that the system was used within intended purpose, provider instructions, approved internal scope, suitable input conditions and valid authority.

Monitored means that the deployer observed not only technical uptime, but also governance conditions such as data freshness, actual use, human oversight, authority drift, model fallback, context integrity and downstream effect.

Human-controllable means that competent, trained, authorized and supported humans could understand, challenge, delay, escalate, override or refuse the AI-supported workflow before consequence attached.

Reconstructable means that logs and evidence chains can show what system was used, what data and context shaped the output, what the AI produced, who reviewed it, what authority applied, what monitoring signals existed and why continuation was allowed.

When output influenced consequence means that governance should focus on the moment AI begins to matter.

Not every AI output creates consequence.

But when it does, the deployer must be able to show control.

This sentence is not a substitute for the legal text.

It is an operational translation.

It turns Article 26 into a deployer control model.

34.3 The Most Important Concepts Developed in This Study

This study has developed several concepts that can help organizations interpret Article 26 in practice.

Evidence of Control

Evidence of control means proof that a control operated when it mattered.

It is stronger than policy.

It is stronger than intention.

It is stronger than a checklist.

Evidence of control requires records showing system identity, input condition, output, human review, authority, monitoring, escalation and downstream consequence.

Continued Validity

Continued validity means that deployment approval remains conditional.

A system may be approved under certain conditions, but those conditions can change.

Actual use can drift.

Input data can become stale.

Human oversight can degrade.

Authority can expire.

Model fallback can occur.

Downstream integrations can create new consequence.

Continued validity asks:

Do the conditions that justified use still exist?

Control Before Consequence

Control before consequence means that oversight, evidence checks and authority validation must occur before the AI-supported output affects a person, process, right, opportunity, clinical pathway, financial process, public service, supplier relationship or operational system.

Post-consequence review may be useful.

But it is not the same as preventive control.

Human-in-Control

Human-in-control is stronger than human-in-the-loop.

A human must not only be present.

They must have competence, training, authority, evidence, time, interface support, organizational backing and the practical ability to intervene.

The core test is:

Could the human meaningfully prevent unsupported, unauthorized or unsafe consequence?

Runtime Governance

Runtime governance is the continuous process of observing current conditions, comparing them with approved control conditions, deciding whether continuation is allowed, acting accordingly, recording the decision and reassessing over time.

It converts Article 26 from a static compliance checklist into an operating model.

ALLOW, HOLD, ESCALATE and REFUSE

These four states make governance actionable.

ALLOW means continuation is justified.

HOLD means required conditions are missing or uncertain.

ESCALATE means higher authority or specialist review is needed.

REFUSE means the workflow should not proceed.

These states prevent uncertainty from being silently absorbed into continuation.

Evidence Architecture

Evidence architecture is the connected structure of records that allows the organization to reconstruct AI-supported workflows.

It connects system, use, data, context, output, oversight, authority, monitoring, escalation, consequence and decision.

Without evidence architecture, runtime governance is fragile.

Authority Continuity

Authority continuity means that the authority attached to AI-supported workflows must remain valid when conditions change.

Model substitution, workflow expansion, downstream automation or role changes can alter authority conditions.

Fallback should not preserve authority silently.

Context as Control

Context management is not only a cost or performance issue.

The context supplied to an AI system shapes reasoning, relevance, authority and output.

Stale, incomplete, irrelevant or unauthorized context can create governance failure.

Reasoning Traceability

Fluent output is not enough.

High-risk AI governance requires inspectable reasoning conditions: facts, rules, constraints, assumptions, evidence, contradictions, uncertainty and refusal boundaries.

Reasoning layers and hybrid AI may help make outputs more governable.

Together, these concepts create a practical language for deployer governance.

34.4 What Article 26 Means for Deployers

For deployers, Article 26 means responsibility does not stop at adoption.

The deployer must understand how the AI system behaves inside its own organization.

This includes:

- who uses it;
- what workflow it affects;
- what data it receives;
- what output it produces;
- who reviews it;
- what authority applies;
- what downstream systems receive the output;
- what people or entities may be affected;
- what evidence is retained;
- what happens when conditions fail.

The deployer must also understand that AI risk is contextual.

A model is not governed in the abstract.

It is governed in use.

An AI system used for internal drafting has one risk profile.

The same system used for employment decisions has another.

The same system used for clinical support has another.

The same system used for public-service eligibility has another.

The same system connected to tools and downstream automation has another.

This means deployer governance should be use-case specific.

A general AI policy is not enough.

The organization needs workflow-level control.

For each high-risk or high-consequence AI use, the deployer should define:

- approved use;
- prohibited use;
- output role;
- consequence point;
- human oversight model;
- input data requirements;
- monitoring signals;

- logging requirements;
- escalation triggers;
- refusal conditions;
- reassessment triggers.

This is the practical work of Article 26.

34.5 What Article 26 Means for Providers and the AI Value Chain

Although this study focuses on deployers, the provider-deployer relationship is central.

Providers supply systems, instructions, limitations, technical documentation, logging capabilities, monitoring expectations and incident communication channels.

Deployers operationalize those systems inside real workflows.

If provider information is unclear, deployer governance becomes harder.

If deployers ignore provider instructions, control weakens.

If logs are insufficient, reconstruction becomes difficult.

If provider updates are not communicated, continued validity may fail.

If fallback or model substitution is invisible, authority continuity may break.

This means Article 26 creates pressure across the AI value chain.

Deployers need providers to supply information that is usable operationally, not only legally.

Useful provider information should help the deployer understand:

- intended purpose;
- limitations;
- required user competence;
- input data requirements;
- output interpretation;
- known failure modes;
- monitoring recommendations;
- logging capabilities;
- update and change process;

- incident contact route;
- fallback or dependency implications.

Deployers then need to translate that information into internal controls.

The future provider-deployer relationship will therefore depend on evidence readiness.

Providers that help deployers build evidence of control may become more trusted.

Deployers that demand operationally useful documentation may become more mature buyers.

AI governance will increasingly be a shared evidence problem.

34.6 What Article 26 Means for Boards and Senior Management

For boards and senior management, Article 26 creates a leadership visibility challenge.

Leadership should not be satisfied with statements such as:

AI governance is in place.

Human oversight exists.

Monitoring is active.

Logs are retained.

The board should ask:

Which systems are high-risk or high-consequence?

Which workflows do they affect?

Which outputs can influence consequence?

Which systems are in ALLOW, HOLD, ESCALATE or REFUSE state?

Which systems lack reconstructable decision trails?

Which systems have weak human oversight?

Which systems have unresolved data freshness issues?

Which systems rely on unapproved fallback models?

Which AI outputs have become downstream triggers?

Which affected-person complaints reveal governance weakness?

Which risks require executive decision?

This is decision-grade AI oversight.

Boards do not need to manage every technical detail.

But they do need visibility into control maturity, unresolved risk and evidence quality.

The key leadership question is:

Are we scaling AI faster than we are scaling control?

If the answer is yes, the organization should slow down, remediate, restrict scope or refuse certain uses.

This is not anti-innovation.

It is responsible scaling.

34.7 What Article 26 Means for Human Oversight

The future of human oversight must move beyond human-in-the-loop language.

A human in the loop may not be enough.

The human may be present but powerless.

The human may approve but not understand.

The human may be accountable but unsupported.

The human may act after consequence.

The human may rubber-stamp.

The human may be overwhelmed.

The human may not see the evidence.

Article 26 points toward a stronger model.

Human oversight should be performed by persons with competence, training, authority and support.

Operationally, this means:

- the human understands the system and domain;

- the human sees relevant evidence;
- the human sees warnings and limitations;
- the human knows the consequence level;
- the human can override;
- the human can hold;
- the human can escalate;
- the human can refuse;
- the human acts before consequence;
- the human review is documented;
- oversight quality is monitored.

This is human-in-control.

The future of deployer governance depends on whether organizations can make this real.

34.8 What Article 26 Means for Agentic AI

Agentic AI makes Article 26 more important, not less.

As AI systems move from answering questions to participating in workflows, the control problem becomes harder.

An agent can plan.

It can retrieve evidence.

It can call tools.

It can update systems.

It can route tasks.

It can trigger workflows.

It can send messages.

It can continue through multiple steps.

This means the deployer must govern transitions, not only outputs.

A final human review may not be enough if the agent has already created operational effects.

Agentic governance requires:

- tool permission mapping;
- transition-level control gates;
- context records;
- input freshness checks;
- authority validation;
- fallback detection;
- downstream effect mapping;
- human review before high-consequence transitions;
- refusal logic;
- reconstructable agent paths.

The future question for agentic AI is:

Can the deployer prove that each relevant transition remained within valid control conditions before consequence attached?

If not, agentic AI may create hidden authority drift.

A system may appear advisory while becoming operationally constructive.

That is one of the most important future governance risks.

34.9 What Article 26 Means for Regulated Sectors

Article 26 will matter differently across sectors.

In financial services, it will connect to auditability, compliance, model risk, operational risk, outsourcing, fund administration and supervisory confidence.

In employment, it will connect to recruitment, worker monitoring, evaluation, promotion, termination, contestability and fairness.

In healthcare, it will connect to patient safety, clinical judgment, validation context, missing data and meaningful clinician oversight.

In procurement, it will connect to supplier risk, due diligence, sanctions, cybersecurity, approval thresholds and critical third-party risk.

In education and public services, it will connect to access, opportunity, administrative legitimacy, affected-person information and reviewability.

Across all sectors, the same pattern appears:

AI output becomes governance-relevant when it influences consequence.

The sector changes the consequence.

The control principle remains the same.

The deployer must know:

What did the AI system affect?

Was the effect allowed?

Was the evidence valid?

Was human control meaningful?

Can the decision trail be reconstructed?

34.10 The Future of AI Governance

The future of AI governance will be shaped by several trends.

1. From Static Governance to Runtime Governance

Organizations will need controls that operate during use, not only before deployment.

Policies and assessments will remain important, but runtime signals will become central.

2. From Human-in-the-Loop to Human-in-Control

Human oversight will need to show real intervention capacity.

A human signature will not be enough where the human lacks evidence, authority or timing.

3. From Logs to Decision Trails

Technical logs will not be sufficient.

Organizations will need reconstructable decision trails that connect input, output, human review, authority, monitoring and consequence.

4. From Model Risk to Workflow Risk

AI risk will increasingly be understood at the workflow level.

The question will not only be whether the model works.

The question will be what the model affects.

5. From Explainability to Inspectability

Natural-language explanations will not be enough.

Organizations will need inspectable facts, rules, constraints, assumptions, context and evidence.

6. From Resilience to Governed Fallback

Fallback models and model substitution will need governance.

Technical continuity must not silently preserve authority.

7. From Context Optimization to Context Governance

Context management will become part of control management.

Freshness, relevance, authority and inspectability of context will matter.

8. From AI Adoption Metrics to AI Control Metrics

Boards will need to know not only how much AI is used, but how much AI use is controlled.

9. From Compliance Burden to Trust Infrastructure

Organizations that build evidence of control may create trust advantages.

They will be better prepared for audits, incidents, regulators, customers, workers, patients, citizens and partners.

10. From AI Systems to AI Control Surfaces

The governance boundary will expand beyond the model.

It will include data, context, tools, humans, workflows, downstream systems, authority, monitoring and evidence.

This is the future shape of deployer governance.

34.11 Practical Final Recommendations

This study concludes with practical recommendations for deployers.

1. Build an AI System Register, But Do Not Stop There

A register is the beginning.

It must connect to workflows, risk, evidence, monitoring and decision trails.

2. Map Actual Use

Do not rely only on approved use.

Monitor how people actually use the system.

Actual use is where risk appears.

3. Identify Consequence Points

Know where AI output begins to affect people, processes or operational systems.

Place control before those points.

4. Strengthen Human Oversight

Ensure humans have competence, training, authority, evidence, time and support.

Measure oversight quality.

5. Govern Input Data and Context

Track freshness, completeness, source reliability, policy version and context integrity.

Do not let stale context carry current authority.

6. Create Decision Trails

Logs should support reconstruction.

A decision trail should show system, input, output, human review, authority, monitoring, escalation and consequence.

7. Implement ALLOW, HOLD, ESCALATE and REFUSE

Do not let every workflow default to continuation.

Governance must include pause, escalation and refusal.

8. Govern Fallback and Model Substitution

Fallback is not neutral.

Changing the model changes the control condition.

9. Monitor Governance Signals

Monitor actual use, human oversight, authority drift, downstream effects, data quality, context integrity, fallback and evidence gaps.

10. Prepare Incident Response

Know who preserves evidence, who contacts the provider, who decides suspension and who approves return to service.

11. Treat Affected-Person Feedback as Governance Signal

Complaints, appeals, challenges and correction requests may reveal control weaknesses.

12. Report Control Maturity to Leadership

Boards should see AI control states, not only AI adoption.

13. Reassess Continuously

AI governance is not complete after deployment.

It must respond to change.

These recommendations are practical because Article 26 is practical.

It requires organizations to operate AI systems responsibly, not only describe responsibility.

34.12 Final Reflection

The promise of AI is not only speed.

It is not only automation.

It is not only efficiency.

It is not only prediction, generation or reasoning.

The real promise of AI in high-consequence environments depends on whether organizations can use it without losing control of the conditions that make decisions legitimate.

This is the challenge Article 26 places before deployers.

It asks them to move beyond the idea that AI governance can be solved through policies alone.

It asks them to build systems of use, oversight, evidence, monitoring and intervention.

It asks them to ensure that humans are not merely present, but capable.

It asks them to ensure that logs are not merely stored, but reconstructable.

It asks them to ensure that monitoring does not merely observe, but triggers action.

It asks them to ensure that AI outputs do not silently become authority.

It asks them to ensure that continuation remains justified.

The most mature AI systems of the future may not be the ones that always answer.

They may be the ones that know when an answer should not become action.

They may be the ones that can show what evidence supported the output.

They may be the ones that can expose uncertainty.

They may be the ones that can pause.

They may be the ones that can escalate.

They may be the ones that can refuse.

They may be the ones that preserve human control before consequence attaches.

This is the deeper meaning of trustworthy AI.

Trustworthy AI is not AI that asks for blind trust.

It is AI deployed in systems that make trust conditional, inspectable, reversible and accountable.

Article 26 points in that direction.

Its practical message is not simply:

Use high-risk AI carefully.

Its deeper message is:

Build the conditions under which careful use can be proven.

That is the future of deployer governance.

Chapter 34 Summary

This final chapter concludes the study by restating its central argument:

Article 26 should be understood not only as a deployer compliance obligation, but as a runtime governance model.

A deployer should be able to prove that a high-risk AI system was used under valid, monitored, human-controllable and reconstructable conditions when its output influenced consequence.

The chapter summarizes the main concepts developed across the study:

- evidence of control;
- continued validity;
- control before consequence;
- human-in-control;
- runtime governance;
- ALLOW, HOLD, ESCALATE and REFUSE;
- evidence architecture;
- authority continuity;
- context as control;
- reasoning traceability.

It explains what Article 26 means for deployers, providers, boards, human oversight, agentic AI and regulated sectors.

The chapter argues that future AI governance will move from static compliance to runtime control, from human-in-the-loop to human-in-control, from logs to decision trails, from model risk to workflow risk, from explainability to inspectability, and from compliance burden to trust infrastructure.

The practical recommendations emphasize system registers, actual-use mapping, consequence points, human oversight, input data and context governance, decision trails, control states, fallback governance, monitoring, incident response, affected-person feedback, board reporting and continuous reassessment.

The final reflection is that trustworthy AI is not built by asking people to trust AI systems blindly.

It is built by creating conditions where trust can be inspected, challenged, limited, paused, escalated, refused and proven.

The future of deployer governance is evidence-based runtime control.

That is the central meaning of Article 26 in practice.

Appendix A

Article 26 Clause-by-Clause Operational Mapping

Legal anchor and framework boundary. [Article 26](#). The mapping distinguishes the Regulation’s clauses from the study’s operational control and evidence interpretations.

A.1 Purpose of This Appendix

This appendix translates Article 26 of Regulation (EU) 2024/1689 into operational deployer controls.

It is designed to help organizations move from legal obligation to practical implementation.

The purpose is not to replace legal advice.

The purpose is to help deployers ask:

What does this obligation mean operationally?

What evidence should we preserve?

What runtime signal should we monitor?

What should happen if the condition fails?

This appendix uses the following mapping structure:

Article 26 Theme	Operational Meaning	Evidence Required	Runtime Signal	Failure State

The failure states follow the governance model developed in this study:

ALLOW — current control conditions support continuation.

HOLD — required conditions are missing, stale, incomplete or uncertain.

ESCALATE — higher authority, specialist review or governance decision is required.

REFUSE — the workflow should not proceed under current conditions.

This appendix should be read as a practical control map for deployers of high-risk AI systems.

A.2 Article 26(1): Use According to Instructions for Use

Article 26 requires deployers of high-risk AI systems to take appropriate technical and organisational measures to ensure that such systems are used in accordance with the instructions for use accompanying the systems.

Article 26 Theme	Operational Meaning	Evidence Required	Runtime Signal	Failure State
Use according to instructions	Provider instructions must be reviewed, understood and translated into internal controls.	Provider instructions, internal control mapping, user procedures, training records.	Actual use matches provider instructions.	ALLOW
Instructions unavailable	The deployer cannot confirm proper use if instructions are missing or inaccessible.	Missing instruction record, provider request record.	System used without verified instructions.	HOLD / ESCALATE
Instructions not operationalized	Instructions exist but are not reflected in workflow controls.	Gap between provider documentation and internal procedure.	Users act without instruction-based controls.	HOLD
Use outside intended purpose	Actual workflow use exceeds provider-defined or internally approved scope.	Intended-purpose record, actual-use evidence, scope drift report.	New workflow, new user group, new consequence level.	ESCALATE / REFUSE
Provider instruction update	Provider changes limitations, use guidance or operating requirements.	Update notice, internal review record, reassessment record.	Provider update received or detected.	HOLD pending reassessment
Prohibited use	The system is used for a purpose that instructions or internal governance prohibit.	Prohibited-use record, access logs, workflow evidence.	Prohibited use detected.	REFUSE

Operational Control Questions

- Do we have the latest instructions for use?
- Have the instructions been translated into internal workflow controls?
- Do users understand the limitations?
- Is actual use monitored against approved use?
- What happens if actual use drifts?

Minimum Evidence

- Provider instructions for use.
- Internal instructions-to-controls mapping.
- Approved use-case record.

- Prohibited-use list.
- Training records.
- Actual-use monitoring record.
- Reassessment trigger record.

A.3 Article 26(2): Human Oversight

Article 26 requires deployers to assign human oversight to natural persons who have the necessary competence, training, authority and support.

Article 26 Theme	Operational Meaning	Evidence Required	Runtime Signal	Failure State
Human oversight assigned	A named role or person is responsible for oversight.	Oversight role record, reviewer assignment.	Reviewer assigned and active.	ALLOW
Competence	Reviewer must understand the domain, workflow, system role and consequence.	Competence criteria, qualifications, role description.	Reviewer competence confirmed.	ALLOW
Training	Reviewer must receive system-specific training.	Training material, completion records, refresher records.	Reviewer trained for current system version and workflow.	ALLOW
Authority	Reviewer must be able to approve, reject, override, hold, escalate or refuse.	Authority matrix, override logs, escalation permissions.	Reviewer authority valid at review time.	ALLOW
Support	Reviewer must have time, evidence, interface support and organizational backing.	Review screen design, evidence access logs, workload metrics.	Reviewer has evidence and capacity.	ALLOW
Reviewer unavailable	Required human oversight cannot operate.	Availability record, review queue, backup assignment.	Reviewer absent or review overdue.	HOLD
Review after consequence	Oversight occurs too late to prevent impact.	Timestamp comparison: output, review, consequence.	Review timestamp after consequence timestamp.	ESCALATE
Rubber-stamp risk	Reviewer approves without meaningful evidence access or intervention capacity.	High approval rate, low review time, low override rate.	Automation bias or superficial review signal.	ESCALATE
Human cannot override	Human presence exists but no practical intervention route exists.	Interface limitation, permission record, process limitation.	Reviewer can approve only.	HOLD / ESCALATE
Human oversight impossible	Required oversight cannot be restored for high-consequence workflow.	Reviewer unavailable, no backup, no authority.	Workflow requires review but none available.	REFUSE

Operational Control Questions

- Who performs human oversight?
- Are they competent and trained?
- Do they have authority to intervene?
- Can they access the evidence behind the AI output?
- Does review occur before consequence attaches?
- Is oversight quality monitored?

Minimum Evidence

- Oversight assignment.
- Competence criteria.
- Training record.
- Authority matrix.
- Evidence access record.
- Review timestamp.
- Override / escalation / refusal records.
- Oversight quality monitoring.

A.4 Article 26(3): Input Data Under Deployer Control

Article 26 requires that where deployers exercise control over input data, they must ensure that input data is relevant and sufficiently representative in view of the intended purpose of the high-risk AI system.

Article 26 Theme	Operational Meaning	Evidence Required	Runtime Signal	Failure State
Input data mapped	Deployer knows what data the system receives.	Data source map, data owner, workflow map.	Source known and approved.	ALLOW
Relevance	Data must fit the intended purpose and workflow.	Relevance assessment, use-case data mapping.	Data source matches approved use.	ALLOW
Representativeness	Where relevant, data must be sufficiently representative for the intended purpose.	Representativeness review, limitations, sampling analysis.	Representativeness accepted or limitations visible.	ALLOW

Article 26 Theme	Operational Meaning	Evidence Required	Runtime Signal	Failure State
Freshness	Data must be current enough for the consequence level.	Freshness threshold, retrieval timestamp, source update timestamp.	Data current at point of use.	ALLOW
Missing required data	Required input is absent or incomplete.	Missing-data warning, required-field check.	Required field missing.	HOLD
Stale data	Data exists but is no longer current enough.	Freshness warning, source timestamp.	Freshness threshold exceeded.	HOLD
Source unreliable	Data comes from unverified, draft, superseded or inappropriate source.	Source reliability classification.	Unreliable source used for decision-critical output.	ESCALATE
Representativeness unknown	Dataset suitability cannot be confirmed for high-risk use.	Missing representativeness assessment.	Unknown suitability.	HOLD / ESCALATE
Data outside approved purpose	Data is used for a purpose not covered by approval.	Data-use mismatch record.	New use of existing data.	ESCALATE / REFUSE
Input data cannot support output	Data condition is insufficient for consequence.	Data quality record, human review note.	Critical missing/stale/invalid data.	REFUSE

Operational Control Questions

- What data does the system receive?
- Who controls the data?
- Is the data relevant to the intended purpose?
- Is the data fresh enough?
- Is it sufficiently representative where required?
- Can stale or missing data trigger HOLD?

Minimum Evidence

- Data source map.
- Data owner record.
- Freshness threshold.
- Completeness check.
- Representativeness assessment where relevant.
- Missing-data rule.
- Input data decision state: ALLOW / HOLD / ESCALATE / REFUSE.

A.5 Article 26(4): Informing Provider or Distributor and Suspending Use Where Relevant

Article 26 includes obligations around action where a deployer has reason to consider that use of the high-risk AI system may result in risk, including informing the provider or distributor and, where appropriate, suspending use.

Article 26 Theme	Operational Meaning	Evidence Required	Runtime Signal	Failure State
Risk signal detected	Deployer identifies possible risk from system use.	Alert, complaint, incident report, monitoring record.	Serious or repeated risk signal.	ESCALATE
Provider notification	Provider or distributor must be informed where required.	Provider contact record, notification timestamp.	Provider-related risk or malfunction.	ESCALATE
Use suspension	Use may need to be suspended where risk cannot be controlled.	Suspension decision, affected workflow list.	Serious risk or unresolved control failure.	HOLD / REFUSE
Continued use with conditions	Use may continue only with reduced authority or additional controls.	Conditional approval, monitoring enhancement, human review rule.	Risk controlled through mitigation.	ALLOW with conditions
Failure to notify	Provider/distributor not informed despite relevant risk.	Missing notification record.	Incident requires external communication but none sent.	ESCALATE
Failure to suspend	Workflow continues despite serious unresolved risk.	Monitoring alert, continued-use logs.	Serious risk active, workflow still operating.	REFUSE / SUSPEND
Return to service	System resumes after risk is remediated and approved.	Return-to-service record, evidence reviewed, approval.	Remediation complete and validated.	ALLOW

Operational Control Questions

- What qualifies as a risk signal?
- Who decides whether provider notification is required?
- Who can suspend use?
- What evidence is preserved before suspension?
- What conditions must be satisfied before return to service?

Minimum Evidence

- Risk signal record.
- Incident triage record.

- Provider or distributor contact route.
- Notification record.
- Suspension decision.
- Return-to-service approval.
- Affected-case review.

A.6 Article 26(5): Monitoring Operation

Article 26 requires deployers to monitor the operation of high-risk AI systems on the basis of the instructions for use.

Article 26 Theme	Operational Meaning	Evidence Required	Runtime Signal	Failure State
Monitoring plan	Deployer defines what must be monitored.	Monitoring plan, signal list, owner.	Monitoring active.	ALLOW
Technical monitoring	Availability, latency, errors, integration health.	Dashboard, logs, incident records.	Technical health normal.	ALLOW
Governance monitoring	Actual use, data freshness, human oversight, authority, downstream effects.	Governance dashboard, alert records.	Governance state normal.	ALLOW
Monitoring based on instructions	Monitoring reflects provider instructions and limitations.	Instructions-to-monitoring mapping.	Instruction-based signals monitored.	ALLOW
Alert ownership	Every significant alert has an owner.	Alert routing, owner matrix.	Alert assigned.	ALLOW
Monitoring gap	Critical condition is not monitored.	Missing signal, audit finding.	Control blind spot.	HOLD / ESCALATE
Alert without action	Alert exists but no response occurs.	Open alert record, no owner action.	Alert unresolved beyond threshold.	ESCALATE
Silent degradation	System technically works but governance condition fails.	Data stale, oversight missing, scope drift.	Control health degraded.	HOLD
Monitoring failure	Monitoring unavailable for high-risk workflow.	Dashboard outage, log failure.	No runtime visibility.	HOLD
Critical signal	Serious risk, invalid authority, unsafe output, missing logs.	Critical alert.	Continued use no longer justified.	REFUSE / SUSPEND

Operational Control Questions

- Are we monitoring more than uptime?
- Do we monitor data, context, human oversight, authority and downstream effect?
- Are alerts assigned to owners?

- Can monitoring trigger HOLD, ESCALATE or REFUSE?
- Is monitoring reviewed by governance teams?

Minimum Evidence

- Monitoring plan.
- Signal categories.
- Alert thresholds.
- Alert owner matrix.
- Dashboard records.
- Alert response record.
- Monitoring review minutes.
- HOLD / ESCALATE / REFUSE logs.

A.7 Article 26(6): Log Retention

Article 26 requires deployers to keep logs automatically generated by the high-risk AI system, to the extent such logs are under their control, for an appropriate period and at least six months unless otherwise provided in applicable law.

Article 26 Theme	Operational Meaning	Evidence Required	Runtime Signal	Failure State
Logs identified	Deployer knows which logs are generated and controlled.	Log inventory, ownership record.	Logs available.	ALLOW
Retention period	Logs retained for required period.	Retention policy, storage record.	Retention active.	ALLOW
Log integrity	Logs protected from alteration or loss.	Access controls, integrity checks.	Log integrity normal.	ALLOW
Decision trail	Logs support reconstruction of system, input, output, review and consequence.	Sample decision trail.	Case reconstructable.	ALLOW
Logs under provider control only	Deployer lacks access to necessary logs.	Provider dependency record.	Critical logs unavailable.	ESCALATE
Logs incomplete	Logs exist but do not support reconstruction.	Audit sample failure.	Missing input/output/review link.	HOLD / ESCALATE
Missing human review log	Cannot prove oversight occurred.	Missing review timestamp or reviewer record.	Review evidence absent.	HOLD
Missing consequence log	Cannot show whether AI output affected outcome.	Downstream action missing.	Consequence path unknown.	ESCALATE

Article 26 Theme	Operational Meaning	Evidence Required	Runtime Signal	Failure State
Log loss	Required logs unavailable or corrupted.	Log integrity alert.	Decision trail unreconstructable.	ESCALATE / REFUSE for critical workflow
Retention failure	Logs not retained long enough.	Retention gap, deletion record.	Legal / audit evidence gap.	ESCALATE

Operational Control Questions

- What logs are generated?
- Which logs are under deployer control?
- Can logs reconstruct the decision trail?
- Are logs retained long enough?
- Are log gaps monitored?
- What happens if logs are incomplete?

Minimum Evidence

- Log inventory.
- Retention policy.
- Log storage record.
- Integrity controls.
- Sample reconstructable decision trail.
- Log gap alerts.
- Remediation record.

A.8 Article 26(7): Workplace Information

Article 26 includes a workplace-related information obligation where a deployer puts into service or uses a high-risk AI system at the workplace. The deployer must inform workers' representatives and affected workers that they will be subject to the use of the high-risk AI system.

Article 26 Theme	Operational Meaning	Evidence Required	Runtime Signal	Failure State
Workplace use identified	Deployer knows whether system affects workers.	Workplace AI assessment, HR use map.	Worker-impacting use identified.	ALLOW

Article 26 Theme	Operational Meaning	Evidence Required	Runtime Signal	Failure State
Worker representatives informed	Representatives are informed where required.	Notice record, communication timestamp.	Notification completed before use.	ALLOW
Affected workers informed	Workers subject to system use are informed.	Worker notice, acknowledgement where appropriate.	Notification completed.	ALLOW
Notice describes AI role	Information explains system role in understandable terms.	Notice content, process description.	Notice matches actual use.	ALLOW
Notice outdated	Use changed but notice not updated.	Scope change, old notice.	Actual use differs from information given.	HOLD / ESCALATE
No notice before use	Workplace high-risk AI used without required information.	Missing notice evidence.	System active in workplace.	ESCALATE
Hidden worker impact	AI affects workers indirectly but was not classified as workplace-impacting.	Monitoring, audit or complaint.	Worker impact discovered late.	ESCALATE
Worker challenge	Worker or representative raises concern.	Complaint, review record.	Challenge received.	ESCALATE
Continued use despite information failure	System continues after missing or misleading notice.	Active workflow, notice gap.	Transparency condition failed.	HOLD / REFUSE depending consequence

Operational Control Questions

- Does the system affect workers?
- Were representatives informed?
- Were affected workers informed?
- Does the notice explain what the AI system does?
- Does the notice still match actual use?
- Are worker concerns treated as governance signals?

Minimum Evidence

- Workplace AI assessment.
- Worker representative notice.
- Worker notice.
- Notice version.
- Actual-use comparison.

- Challenge / complaint record.
- Notice update trigger.

A.9 Article 26(8): Fundamental Rights Impact Assessment Link

Article 26 includes specific references to deployers subject to Article 27 obligations where relevant. Article 27 concerns fundamental rights impact assessments for certain high-risk AI systems. This appendix treats the Article 27 link as a deployer readiness and evidence integration issue.

Article 26 Theme	Operational Meaning	Evidence Required	Runtime Signal	Failure State
FRIA applicability assessed	Deployer determines whether Article 27 applies.	FRIA applicability assessment.	Assessment completed.	ALLOW
FRIA completed where required	Assessment performed before relevant use.	FRIA document, approval record.	FRIA available.	ALLOW
FRIA linked to controls	Risks identified in FRIA translated into operational controls.	Risk-to-control mapping.	Controls active.	ALLOW
FRIA not updated	Use changes but FRIA not reassessed.	Scope change, old FRIA.	Actual use differs from assessed use.	HOLD / ESCALATE
FRIA missing	Required FRIA not completed.	Gap record.	High-risk use pending or active.	HOLD / REFUSE
Fundamental rights concern	Monitoring, complaint or appeal suggests rights impact.	Concern record, escalation.	Rights-related signal detected.	ESCALATE
Continued use despite unresolved FRIA risk	System operates with unresolved material rights risk.	Risk register, active use log.	Unresolved high-risk condition.	REFUSE / SUSPEND

Operational Control Questions

- Does Article 27 apply?
- Has a fundamental rights impact assessment been completed where required?
- Are FRIA risks linked to runtime controls?
- Are affected-person complaints linked back to the FRIA?
- Does use expansion trigger reassessment?

Minimum Evidence

- FRIA applicability assessment.
- FRIA record where required.
- Risk-to-control mapping.

- Monitoring link.
- Reassessment record.
- Remediation tracker.

A.10 Article 26(9): Public Authority and EU Database Considerations

Article 26 includes obligations for certain deployers that are public authorities, Union institutions, bodies, offices or agencies, or persons acting on their behalf, including registration-related obligations in relation to the EU database for certain high-risk AI systems. This appendix treats this as a public-sector evidence and accountability requirement.

Article 26 Theme	Operational Meaning	Evidence Required	Runtime Signal	Failure State
Public authority status assessed	Deployer determines whether public authority / acting-on-behalf condition applies.	Applicability assessment.	Status confirmed.	ALLOW
Registration requirement assessed	Deployer checks EU database obligation where applicable.	Registration assessment.	Requirement known.	ALLOW
Registration completed where required	Relevant system registered before use where required.	Database registration record.	Registration complete.	ALLOW
Public-facing process mapped	AI system's role in public service is documented.	Public process map, affected-person map.	Public process known.	ALLOW
Registration missing	Required registration not completed.	Missing registration record.	Public-sector high-risk system active.	HOLD / ESCALATE
Actual use differs from registered use	System use changes after registration.	Actual-use monitoring, registration comparison.	Use drift detected.	ESCALATE
Public accountability gap	Public authority cannot reconstruct AI influence on case.	Missing decision trail.	Case reconstruction fails.	ESCALATE
Continued use without required public-sector evidence	System remains active despite registration/ evidence gap.	Active logs, unresolved gap.	Accountability condition failed.	HOLD / REFUSE

Operational Control Questions

- Is the deployer a public authority or acting on behalf of one?
- Does registration apply?
- Has registration been completed where required?

- Does registered use match actual use?
- Can the authority reconstruct how AI influenced a public-facing decision?

Minimum Evidence

- Public authority applicability assessment.
- EU database registration assessment.
- Registration record where required.
- Public process map.
- Actual-use monitoring.
- Public-sector decision trail.

A.11 Article 26(10): Affected Natural Persons

Article 26 includes an information obligation where high-risk AI systems are used to make or assist decisions relating to natural persons. The deployer must inform natural persons that they are subject to the use of the high-risk AI system, where applicable under the provision.

Article 26 Theme	Operational Meaning	Evidence Required	Runtime Signal	Failure State
Natural-person impact assessed	Deployer determines whether AI assists decisions relating to natural persons.	Affected-person assessment.	Impact known.	ALLOW
Information provided	Affected persons informed where required.	Notice record, timing record.	Notice completed before or at appropriate stage.	ALLOW
AI role explained	Notice explains how AI assists the decision or process.	Notice content, plain-language explanation.	Notice matches actual use.	ALLOW
Challenge route available	Affected persons have route for questions, correction or review where relevant.	Complaint / challenge process.	Route available.	ALLOW
Notice missing	Affected natural persons not informed where required.	Missing notice evidence.	AI used in decision-assisting role.	HOLD / ESCALATE
Notice misleading	Notice understates AI role or does not match actual use.	Actual-use comparison.	AI influence greater than disclosed.	ESCALATE
Affected-person complaint	Person challenges AI-supported outcome.	Complaint, case review, decision trail.	Complaint received.	ESCALATE
Cannot reconstruct case	Organization cannot explain AI influence on person's outcome.	Missing case evidence.	Decision trail incomplete.	ESCALATE / REFUSE for continued use

Article 26 Theme	Operational Meaning	Evidence Required	Runtime Signal	Failure State
Continued use despite unresolved information failure	System continues affecting persons despite missing information.	Active use log, notice gap.	Transparency condition failed.	HOLD / REFUSE depending risk

Operational Control Questions

- Does the system assist decisions relating to natural persons?
- Were affected persons informed where required?
- Is the AI role explained clearly?
- Does information reflect actual use?
- Can a person's AI-supported pathway be reconstructed?

Minimum Evidence

- Affected-person assessment.
- Notice record.
- Notice version.
- AI role explanation.
- Challenge route.
- Complaint / appeal records.
- Case-level decision trail.

A.12 Article 26 Operational Mapping Summary Table

The following summary table consolidates the key deployer obligations into an operational view.

Article 26 Area	Core Operational Question	Minimum Evidence	Runtime Failure State
Instructions for use	Is the system being used according to provider instructions and approved scope?	Instructions, internal control mapping, actual-use record	HOLD / ESCALATE / REFUSE
Human oversight	Can a competent, trained, authorized and supported human intervene before consequence?	Reviewer assignment, training, authority, review logs	HOLD / ESCALATE / REFUSE
Input data	Does input data support the intended purpose and consequence level?	Data map, freshness, completeness, representativeness evidence	HOLD / ESCALATE / REFUSE

Article 26 Area	Core Operational Question	Minimum Evidence	Runtime Failure State
Risk response	Can the organization act when risk is detected?	Incident triage, provider contact, suspension process	ESCALATE / SUSPEND
Monitoring	Does monitoring detect technical and governance degradation?	Monitoring plan, signals, alerts, response records	HOLD / ESCALATE
Log retention	Can the organization reconstruct the AI-supported workflow?	Logs, retention, integrity, decision trail	HOLD / ESCALATE
Workplace information	Are workers and representatives informed where required?	Worker notice, representative communication	HOLD / ESCALATE
FRIA link	Has fundamental rights impact been assessed where required?	FRIA assessment, risk-to-control mapping	HOLD / ESCALATE / REFUSE
Public authority / EU database	Are public-sector registration and accountability duties handled where required?	Registration assessment, record, public process map	HOLD / ESCALATE
Affected natural persons	Are natural persons informed where required and can their case be reviewed?	Notice, challenge route, case trail	HOLD / ESCALATE / REFUSE

A.13 Failure-State Interpretation

Article 26 implementation should not assume that every gap has the same consequence. The correct response depends on risk, sector, consequence and reversibility.

ALLOW

Use ALLOW where:

- instructions are followed;
- data is suitable;
- oversight is meaningful;
- authority is valid;
- monitoring shows no unresolved critical signal;
- logs support reconstruction;
- information duties are satisfied where required.

HOLD

Use HOLD where:

- evidence is stale;

- data is incomplete;
- reviewer is unavailable;
- context is unclear;
- logs are incomplete;
- actual use requires verification;
- notice accuracy needs confirmation.

HOLD means continuation pauses until the condition is restored.

ESCALATE

Use ESCALATE where:

- use may be outside intended purpose;
- authority is unclear;
- provider input is required;
- legal, compliance, HR, clinical, procurement, public-sector or risk review is needed;
- affected-person complaint raises governance concerns;
- monitoring shows repeated or serious control degradation.

ESCALATE means the issue exceeds ordinary workflow handling.

REFUSE

Use REFUSE where:

- use is prohibited;
- human oversight cannot be provided where required;
- evidence cannot support the output;
- authority is absent;
- serious risk remains unresolved;
- logs cannot support required accountability;
- fallback or substitution is not approved;
- downstream consequence cannot be controlled.

REFUSE means the workflow should not proceed under current conditions.

A.14 Article 26 Evidence Pack

A deployer should maintain an Article 26 evidence pack for each high-risk AI system.

Minimum Article 26 Evidence Pack

1. AI system record.
2. Provider instructions for use.
3. Intended purpose and approved internal use.
4. Actual-use monitoring record.
5. Instructions-to-controls mapping.
6. Human oversight assignment.
7. Human oversight competence and training records.
8. Human authority and override matrix.
9. Input data source map.
10. Input data freshness and completeness rules.
11. Representativeness assessment where relevant.
12. Monitoring plan.
13. Runtime signal catalogue.
14. Alert response rules.
15. Log inventory.
16. Log retention and integrity controls.
17. Sample decision trail.
18. Incident triage procedure.
19. Provider / distributor contact route.
20. Suspension and return-to-service procedure.
21. Workplace information assessment.
22. Affected-person information assessment.
23. FRIA applicability assessment where relevant.

24. Public authority / EU database registration assessment where relevant.

25. Reassessment triggers.

26. Remediation tracker.

27. Board or governance review record.

This evidence pack should answer one central question:

Can the deployer prove that the high-risk AI system was used under valid, monitored, human-controllable and reconstructable conditions when its output influenced consequence?

A.15 Practical Implementation Sequence

A deployer can implement Article 26 operationally through the following sequence.

Step 1 — Identify

Identify all high-risk and high-consequence AI systems.

Step 2 — Classify

Determine system risk, affected process and consequence level.

Step 3 — Map

Map intended purpose, actual use, input data, users, affected persons and downstream systems.

Step 4 — Translate Instructions

Convert provider instructions into internal controls.

Step 5 — Assign Human Oversight

Assign competent, trained, authorized and supported reviewers.

Step 6 — Govern Input Data

Define data relevance, representativeness, freshness and missing-data rules.

Step 7 — Build Monitoring

Monitor technical health and governance conditions.

Step 8 — Build Decision Trails

Ensure logs can reconstruct system, input, output, review, authority and consequence.

Step 9 — Define Failure States

Implement ALLOW, HOLD, ESCALATE and REFUSE logic.

Step 10 — Prepare Incident Response

Create provider notification, suspension and return-to-service procedures.

Step 11 — Inform Where Required

Handle workplace, affected-person, FRIA and public authority requirements.

Step 12 — Reassess

Reassess after use changes, model changes, provider updates, incidents, complaints or monitoring alerts.

A.16 Appendix A Summary

Article 26 can be operationalized as a deployer control map. Its obligations are not isolated checklist items. They connect into a runtime governance structure.

Instructions for use define the permitted operating conditions. Human oversight provides intervention capacity. Input data controls determine whether the system receives suitable evidence. Monitoring detects when conditions degrade. Logs preserve reconstructability. Risk and incident response provide escalation and suspension pathways. Workplace and affected-person information make AI use visible where required. Public authority and FRIA-related obligations connect AI use to institutional accountability and fundamental rights protection.

The operational question behind Article 26 is: Can the deployer prove that the AI-supported workflow remained within valid control conditions when the output influenced consequence?

This appendix provides a clause-by-clause practical mapping to support that proof.

Appendix B

ALLOW / HOLD / ESCALATE / REFUSE Decision Tables

Legal anchor and framework boundary. [Article 14](#); [Article 26](#). ALLOW, HOLD, ESCALATE and REFUSE are NextOne Observatory decision states, not legal categories.

B.1 Purpose of This Appendix

This appendix provides practical decision tables for AI deployers using high-risk or high-consequence AI systems.

The purpose is to translate governance signals into operational decisions.

A deployer should not only ask:

Is there a policy?

Is there monitoring?

Is there human oversight?

Is there a log?

The deployer must also ask:

What should the workflow do when a control condition is satisfied, missing, degraded or invalid?

This appendix uses four governance states:

ALLOW

The AI-supported workflow may continue because current control conditions support continuation.

HOLD

The workflow should pause because required conditions are missing, stale, incomplete, uncertain or degraded, but may be restored.

ESCALATE

The issue requires review by a higher authority, specialist function, governance owner, provider or responsible decision-maker.

REFUSE

The workflow should not proceed under current conditions.

These states help prevent one of the most dangerous AI governance failures:

silent continuation.

Silent continuation occurs when the system completes the workflow even though the conditions required for valid continuation are no longer present.

The tables below are designed to help organizations decide when to continue, pause, escalate or refuse.

B.2 General Decision Logic

Governance State	Meaning	Typical Condition	Operational Action	Evidence Required
ALLOW	Current control conditions support continuation.	System is within intended purpose, evidence is current, oversight is satisfied, authority is valid, logs are complete.	Continue workflow.	Record reason for continuation where consequence is high.
HOLD	Required condition is missing, stale, incomplete or uncertain.	Data stale, reviewer unavailable, context incomplete, authority unclear, log gap.	Pause workflow until condition is restored.	HOLD record, owner, missing condition, resolution action.
ESCALATE	Issue exceeds ordinary workflow handling.	Scope drift, policy conflict, serious risk signal, affected-person complaint, fallback	Send to responsible authority or specialist review.	Escalation record, recipient, reason, outcome.

Governance State	Meaning	Typical Condition	Operational Action	Evidence Required
REFUSE	Workflow should not proceed.	authority issue. Prohibited use, no valid authority, required oversight impossible, evidence cannot support output, serious risk unresolved.	Block or stop workflow.	Refusal record, reason, boundary condition, remediation path.

Core Principle

A workflow should not move to consequence merely because the AI system can produce an output.

The question is:

Are current control conditions strong enough to justify continuation?

B.3 Intended Purpose Decision Table

Condition	Governance State	Action	Evidence
Actual use matches provider intended purpose and approved internal use.	ALLOW	Continue under normal controls.	Intended-purpose map, approved use record.
Actual use is adjacent to approved use but consequence remains low.	HOLD / ESCALATE	Pause or request governance review before expansion.	Scope-change record.
Actual use expands to new workflow or new user group.	ESCALATE	Governance review required.	Actual-use monitoring record.

Condition	Governance State	Action	Evidence
Output is used for higher consequence than approved.	ESCALATE / REFUSE	Stop higher-consequence use until reassessed.	Output authority record.
System used for prohibited purpose.	REFUSE	Block use.	Prohibited-use rule, refusal log.
Intended purpose unknown or provider instructions unavailable.	HOLD	Do not use for high-risk workflow until clarified.	Missing instruction record, provider request.
Provider instructions changed and internal controls not yet updated.	HOLD	Pause affected use pending reassessment.	Provider update notice, reassessment record.

Practical Example

A HR AI system approved for CV summarization begins being used to rank candidates.

Correct state: **ESCALATE**.

Reason: actual use has moved from summarization to ranking, and ranking may affect employment opportunity.

B.4 Input Data Decision Table

Condition	Governance State	Action	Evidence
Required input data is present, current and relevant.	ALLOW	Continue.	Data freshness record, source map.
Non-critical input data is missing but consequence is low.	ALLOW with warning / HOLD	Continue with warning or pause depending risk.	Missing-data warning.
Required input	HOLD	Pause until data is	Required-field

Condition	Governance State	Action	Evidence
data is missing.		supplied.	failure record.
Input data is stale beyond threshold.	HOLD	Refresh data before use.	Freshness timestamp.
Data source reliability is unknown.	HOLD / ESCALATE	Verify source or escalate.	Source reliability record.
Conflicting data sources exist.	ESCALATE	Specialist or owner review required.	Conflict record.
Input data appears unsuitable for intended purpose.	ESCALATE / REFUSE	Reassess before use.	Data suitability finding.
Data cannot be restored or validated for high-consequence workflow.	REFUSE	Do not rely on AI output.	Refusal record.

Practical Example

A procurement AI recommends a supplier, but the sanctions check is current and the cybersecurity review is missing.

Correct state: **HOLD** for final approval.

Reason: recommendation may support preliminary review, but final approval lacks required evidence.

B.5 Context Decision Table for LLM and Agentic Systems

Condition	Governance State	Action	Evidence
Context sources are approved, current and inspectable.	ALLOW	Continue.	Context bundle ID, source list.
Context includes older background information clearly marked as historical.	ALLOW	Continue with appropriate weighting.	Context metadata.

Condition	Governance State	Action	Evidence
Context freshness is unknown for decision-critical evidence.	HOLD	Refresh or verify context.	Freshness gap record.
Wrong policy version retrieved.	HOLD / ESCALATE	Stop output reliance and retrieve current version.	Policy mismatch record.
Context compaction removed warnings or limitations.	HOLD	Rebuild context with governance-critical elements.	Compaction record.
Context includes unapproved or unauthorized source.	ESCALATE	Review access and source validity.	Source authorization record.
Context conflict cannot be resolved.	ESCALATE	Specialist review required.	Conflict log.
Context cannot be reconstructed for high-consequence output.	REFUSE / ESCALATE	Do not rely on output; investigate evidence gap.	Missing context record.

Practical Example

A compliance assistant uses an old policy summary in the context window.

Correct state: **HOLD**.

Reason: the output may be fluent, but the policy context may no longer support the recommendation.

B.6 Human Oversight Decision Table

Condition	Governance State	Action	Evidence
Competent, trained, authorized reviewer completes review before	ALLOW	Continue.	Review record, training record, authority record.

Condition	Governance State	Action	Evidence
consequence.			
Reviewer is assigned but training is incomplete.	HOLD / ESCALATE	Pause high-consequence use or assign trained reviewer.	Training gap record.
Reviewer is unavailable and no backup exists.	HOLD	Pause until oversight is restored.	Reviewer availability record.
Review occurs after consequence attaches.	ESCALATE	Review oversight design; possible remediation.	Timestamp comparison.
Reviewer can approve but cannot override or stop.	HOLD / ESCALATE	Redesign authority or workflow.	Authority gap record.
Reviewer lacks source evidence access.	HOLD	Provide evidence access before review.	Evidence access gap.
Reviewer workload makes meaningful review unrealistic.	ESCALATE	Increase capacity or reduce workflow authority.	Workload metrics.
Required human oversight cannot be provided for high-consequence workflow.	REFUSE	Do not proceed.	Refusal record.

Practical Example

A clinician receives an AI-generated summary, but the system does not show that recent lab results were missing.

Correct state: **HOLD**.

Reason: human oversight is weakened because the reviewer cannot see the input data condition.

B.7 Authority Decision Table

Condition	Governance State	Action	Evidence
Reviewer and workflow authority are valid for consequence level.	ALLOW	Continue.	Authority matrix, approval record.
Authority exists for advisory output only.	ALLOW with limits	Output may remain advisory but not trigger decision.	Output authority level.
Approval threshold exceeded.	ESCALATE	Route to higher authority.	Threshold record.
Authority unclear.	HOLD	Pause until authority is confirmed.	Authority validation record.
Reviewer role changed or delegation expired.	HOLD / ESCALATE	Revalidate authority.	Role/delegation record.
AI output becomes downstream trigger without approved authority.	ESCALATE / REFUSE	Stop trigger until control gate approved.	Downstream effect map.
User lacks permission for requested action.	REFUSE	Block action.	Permission record.
System attempts action outside authorized scope.	REFUSE	Block and log.	Refusal log.

Practical Example

A supplier recommendation moves from advisory memo into automatic approval routing.

Correct state: **ESCALATE**.

Reason: recommendation has become an operational trigger and authority must be reassessed.

B.8 Model Fallback and Substitution Decision Table

Condition	Governance State	Action	Evidence
Primary model available and approved.	ALLOW	Continue.	Model registry.
Fallback model approved for same workflow and authority level.	ALLOW	Continue and log fallback.	Fallback approval record.
Fallback model approved only for reduced authority.	ALLOW with reduced authority / HOLD	Limit output to draft or advisory use.	Authority reduction record.
Fallback model used in high-consequence workflow without approval.	HOLD / ESCALATE	Pause and governance review.	Fallback event log.
Fallback model cannot support required evidence quality.	HOLD	Require fresh evidence or human review.	Evidence quality gap.
Fallback model changes provider/data/security conditions.	ESCALATE	Review third-party and data implications.	Provider dependency record.
Fallback model not approved for workflow.	REFUSE	Block fallback use.	Refusal record.
Model substitution occurred silently and cannot be reconstructed.	ESCALATE / REFUSE	Investigate and suspend affected workflow if necessary.	Missing model event record.

Practical Example

A high-risk HR workflow falls back from an approved model to a general-purpose model not approved for candidate ranking.

Correct state: **REFUSE** for ranking.

Reason: fallback should not silently preserve authority.

B.9 Monitoring Signal Decision Table

Signal	Governance State	Action	Evidence
No active alerts and controls normal.	ALLOW	Continue.	Monitoring record.
Low-severity warning.	ALLOW with monitoring	Track trend.	Warning record.
Repeated warning in same control area.	ESCALATE	Review systemic issue.	Trend report.
Required data source unavailable.	HOLD	Pause affected workflow.	Source outage record.
Human oversight backlog exceeds threshold.	HOLD / ESCALATE	Reduce volume or add reviewers.	Oversight metrics.
Actual use drift detected.	ESCALATE	Governance review.	Use drift record.
Log integrity issue detected.	HOLD / ESCALATE	Fix logging before high-consequence use continues.	Log integrity alert.
Serious risk signal detected.	ESCALATE / REFUSE / SUSPEND	Incident triage and possible suspension.	Incident record.
Monitoring unavailable for high-risk workflow.	HOLD	Pause until visibility restored.	Monitoring outage record.

Practical Example

A dashboard shows normal uptime but repeated stale-data warnings in a procurement workflow.

Correct state: **HOLD** for affected decisions.

Reason: technical health does not prove governance health.

B.10 Logging and Decision Trail Decision Table

Condition	Governance State	Action	Evidence
Logs reconstruct system, input, output, review, authority and consequence.	ALLOW	Continue.	Sample decision trail.
Logs show output but not input condition.	HOLD / ESCALATE	Improve logging before high-consequence use.	Log gap record.
Logs show human approval but not what reviewer saw.	HOLD	Add evidence access record.	Oversight evidence gap.
Logs do not show downstream consequence.	ESCALATE	Map downstream effect and update logs.	Downstream gap record.
Logs missing for non-critical workflow.	ESCALATE	Remediate.	Log incident.
Logs missing for high-consequence workflow.	HOLD / REFUSE	Pause affected use until reconstructability restored.	Critical log gap.
Log integrity compromised.	ESCALATE / REFUSE	Preserve evidence, investigate, suspend if necessary.	Integrity alert.
Decision trail cannot be reconstructed after complaint or incident.	ESCALATE	Internal review and remediation.	Audit finding.

Practical Example

A public-service AI recommendation influenced case routing, but the organization cannot reconstruct which input data was used.

Correct state: **ESCALATE**.

Reason: case-level accountability is not supported.

B.11 Affected-Person Information Decision Table

Condition	Governance State	Action	Evidence
Information duty assessed and not applicable.	ALLOW	Continue.	Applicability assessment.
Affected persons informed where required.	ALLOW	Continue.	Notice record.
Notice exists and matches actual use.	ALLOW	Continue.	Notice version, actual-use map.
Notice is generic and does not explain AI role.	HOLD / ESCALATE	Improve notice before expanded use.	Notice quality review.
AI role changed but notice not updated.	HOLD	Update information before continued affected-person use.	Scope-change record.
Affected natural persons not informed where required.	ESCALATE	Legal/governance review.	Missing notice record.
Complaint or challenge received.	ESCALATE	Reconstruct case and review outcome.	Complaint record.
Case cannot be reconstructed after affected-person challenge.	ESCALATE / REFUSE	Investigate; suspend similar use if systemic.	Reconstruction failure record.

Practical Example

An AI system assists public-service eligibility decisions, but the notice only says “technology may be used to improve services.”

Correct state: **ESCALATE**.

Reason: information may not meaningfully describe AI's role in the decision-assisting process.

B.12 Workplace AI Decision Table

Condition	Governance State	Action	Evidence
Workplace use assessed and no worker impact exists.	ALLOW	Continue.	Workplace impact assessment.
Workers and representatives informed where required.	ALLOW	Continue.	Notice records.
Notice reflects actual AI role.	ALLOW	Continue.	Notice-use comparison.
System begins affecting worker evaluation or task allocation.	ESCALATE	Reassess workplace information and controls.	Actual-use change record.
Workers not informed before relevant use.	HOLD / ESCALATE	Pause or remediate information process.	Missing notice record.
Worker complaint received.	ESCALATE	Review case and system behaviour.	Complaint record.
AI monitoring tool creates disciplinary signal without proper oversight.	ESCALATE / REFUSE	Stop disciplinary reliance pending review.	Workflow evidence.
Workplace use continues despite unresolved information or oversight failure.	REFUSE / SUSPEND	Stop affected workflow.	Refusal/suspension record.

Practical Example

A productivity AI tool begins being used in performance discussions, although workers were only told it supports scheduling.

Correct state: **ESCALATE**.

Reason: actual use changed and may affect worker treatment.

B.13 Agentic AI Tool-Use Decision Table

Condition	Governance State	Action	Evidence
Agent tool access is read-only and output remains internal draft.	ALLOW	Continue with logs.	Tool permission map.
Agent retrieves evidence from approved sources.	ALLOW	Continue.	Tool-call log.
Agent attempts unapproved tool call.	REFUSE	Block tool call.	Refusal log.
Agent attempts write action without approval.	HOLD / REFUSE	Require human approval or block.	Tool action log.
Agent output triggers downstream workflow.	ESCALATE	Validate control gate and authority.	Downstream effect record.
Agent sends external message without required review.	REFUSE	Block and escalate.	Communication log.
Agent uses stale retrieved context.	HOLD	Refresh evidence.	Retrieval timestamp.
Agent path cannot be reconstructed.	ESCALATE / REFUSE	Investigate and restrict use.	Missing agent trace.

Practical Example

A customer support agent drafts a refund response and automatically sends it to the customer without human review, although the workflow was approved for draft-only use.

Correct state: **REFUSE**.

Reason: draft-only authority was exceeded and external consequence occurred.

B.14 Sector-Specific Decision Tables

B.14.1 Financial Services

Condition	Governance State	Action
AI compliance summary uses current policy and source evidence.	ALLOW	Human review may proceed.
Policy version unknown.	HOLD	Retrieve current policy before use.
AI recommendation could close regulated case.	ESCALATE	Compliance authority review required.
Fallback model used for regulated decision support without approval.	HOLD / ESCALATE	Apply authority reduction.
Case cannot be reconstructed for audit.	ESCALATE	Internal audit review.
AI output caused or may cause regulatory misreporting.	REFUSE / SUSPEND	Stop affected workflow and incident review.

B.14.2 Employment

Condition	Governance State	Action
AI summarizes CVs and recruiter reviews source CVs before shortlist.	ALLOW	Continue.
AI ranking used to exclude candidates without review.	REFUSE	Stop ranking-based exclusion.
Candidate data incomplete.	HOLD	Request or verify missing evidence.
Reviewer lacks training on AI ranking limitations.	HOLD	Train or assign competent reviewer.
Worker monitoring output affects performance process.	ESCALATE	HR governance review.

Condition	Governance State	Action
Complaint alleges unfair AI-supported outcome.	ESCALATE	Reconstruct case.

B.14.3 Healthcare

Condition	Governance State	Action
Clinical summary uses current patient records and clinician review occurs before care impact.	ALLOW	Continue.
Latest labs missing.	HOLD	Refresh record before clinical reliance.
AI output suggests action outside intended purpose.	REFUSE	Do not rely on output.
Clinician cannot see source evidence.	HOLD	Provide source access.
Patient safety near miss occurs.	ESCALATE / SUSPEND	Incident review.
System output cannot be reconstructed.	ESCALATE	Clinical safety and governance review.

B.14.4 Procurement

Condition	Governance State	Action
Supplier evidence current and reviewer authority valid.	ALLOW	Continue.
Financial data stale.	HOLD	Refresh before recommendation.
Cybersecurity review pending for sensitive-data supplier.	HOLD	Pause final approval.
Contract value exceeds approver threshold.	ESCALATE	Higher approval required.
Supplier on exclusion/sanctions list.	REFUSE	Block supplier approval.
AI recommendation becomes automatic approval trigger.	ESCALATE	Control gate required.

B.14.5 Public Sector and Education

Condition	Governance State	Action
AI supports case summary and official reviews source evidence.	ALLOW	Continue.
AI prioritization affects access without transparency assessment.	ESCALATE	Review information duty and fairness.
Case evidence incomplete.	HOLD	Request missing information.
AI output affects student assessment without human review.	ESCALATE / REFUSE	Stop unsupported reliance.
Citizen complaint received.	ESCALATE	Reconstruct case pathway.
Public-service case cannot be reconstructed.	ESCALATE / SUSPEND	Review affected workflows.

B.15 Control-State Escalation Ladder

The four states can also be understood as a ladder.

Level	State	Meaning	Typical Owner
1	ALLOW	Workflow may continue.	Workflow owner / reviewer
2	HOLD	Workflow pauses until condition restored.	Workflow owner / data owner / reviewer
3	ESCALATE	Higher authority must review.	Governance, risk, compliance, legal, clinical, HR, procurement, public authority owner
4	REFUSE	Workflow must not proceed.	Authorized governance owner / system owner / senior

Level	State	Meaning	Typical Owner
			accountable owner

Escalation Principle

A workflow should not remain indefinitely in HOLD.

If a HOLD condition cannot be resolved within a defined time or if consequence is high, the case should move to ESCALATE.

If escalation confirms that conditions cannot support continuation, the case should move to REFUSE.

B.16 Minimum Evidence for Each State

State	Minimum Evidence
ALLOW	Control conditions satisfied, timestamp, system/workflow ID, reviewer or automated check, reason for continuation.
HOLD	Missing or degraded condition, timestamp, owner, action needed, deadline or trigger for review.
ESCALATE	Escalation reason, recipient, timestamp, evidence package, decision owner, outcome.
REFUSE	Boundary condition, reason for refusal, affected workflow, authority for refusal, remediation path if available.

Practical Rule

If the organization cannot explain why a workflow was allowed to continue, it should not treat ALLOW as proven.

B.17 Decision Table for Reassessment Triggers

Trigger	Governance State	Required Action
Provider instruction update	HOLD / ESCALATE	Review impact before continued use.

Trigger	Governance State	Required Action
Model version change	HOLD	Validate against approved use.
Fallback model event	HOLD / ESCALATE	Apply fallback authority rules.
New data source added	ESCALATE	Data governance review.
Actual use expands	ESCALATE	Reassess intended purpose and risk.
Downstream integration added	ESCALATE	Map consequence and control gate.
Human oversight model changes	HOLD	Confirm competence, training, authority and support.
Incident or near miss	ESCALATE	Incident review and possible suspension.
Affected-person complaint	ESCALATE	Reconstruct case and review controls.
Log gap detected	HOLD / ESCALATE	Restore reconstructability.
Authority threshold changes	HOLD	Revalidate approval path.
Regulatory or internal policy change	ESCALATE	Reassess affected systems.

B.18 Compact Operational Decision Tree

The following decision tree can be used by workflow owners.

Step 1 — Is the use within intended purpose?

- Yes → go to Step 2.
- No / unclear → HOLD or ESCALATE.
- Prohibited → REFUSE.

Step 2 — Is input data valid for this consequence?

- Yes → go to Step 3.
- Missing / stale / incomplete → HOLD.
- Unsuitable or impossible to validate → REFUSE or ESCALATE.

Step 3 — Is human oversight required?

- No → go to Step 5.
- Yes → go to Step 4.

Step 4 — Is meaningful human oversight available before consequence?

- Yes → go to Step 5.
- Reviewer unavailable / not trained / no authority / no evidence → HOLD.
- Oversight impossible → REFUSE.

Step 5 — Is authority valid?

- Yes → go to Step 6.
- Unclear → HOLD.
- Exceeded threshold → ESCALATE.
- Absent → REFUSE.

Step 6 — Are monitoring alerts active?

- No critical alerts → go to Step 7.
- Warning only → ALLOW with monitoring.
- Control alert → HOLD or ESCALATE.
- Serious risk alert → REFUSE or SUSPEND.

Step 7 — Can the decision trail be reconstructed?

- Yes → ALLOW.
 - Partially → HOLD or ESCALATE depending consequence.
 - No, and consequence is high → REFUSE until evidence is restored.
-

B.19 Appendix B Summary

The ALLOW / HOLD / ESCALATE / REFUSE model turns AI governance into operational decision-making.

It prevents AI-supported workflows from continuing automatically when required control conditions are missing, stale, uncertain or invalid.

The model helps deployers translate Article 26 obligations into runtime decisions:

- instructions for use become intended-purpose checks;
- input data obligations become freshness and suitability states;
- human oversight becomes intervention capacity;
- monitoring becomes control-state detection;
- logs become reconstructable decision trails;

- incident response becomes escalation and suspension;
- affected-person information becomes transparency condition;
- authority becomes permission to continue.

The most important lesson is:

ALLOW should be earned by evidence.

If evidence is incomplete, the workflow should HOLD.

If the issue exceeds ordinary workflow authority, it should ESCALATE.

If the use is unsupported, unauthorized or unsafe, it should REFUSE.

This is how deployers move from passive compliance to runtime control.

Appendix C

Evidence Pack Templates and Sample Forms

Legal anchor and framework boundary. [Article 26](#). Templates are illustrative implementation aids and must be adapted to applicable law, sector and organisational context.

C.1 Purpose of This Appendix

This appendix provides practical templates and sample forms for building an Article 26 evidence pack.

The purpose is to help deployers preserve evidence that a high-risk AI system was used under valid control conditions.

The templates are designed to support:

- system-level governance;
- workflow-level control;
- case-level reconstruction;
- human oversight evidence;
- input data evidence;
- context evidence;
- monitoring evidence;
- incident response evidence;
- board and audit readiness.

These templates are not legal advice.

They are operational governance tools.

They should be adapted to the organization's sector, risk profile, internal governance model and applicable legal requirements.

The central question behind every template is:

Can the deployer prove that the AI-supported workflow was allowed to continue under valid, monitored, human-controllable and reconstructable conditions when its output influenced consequence?

C.2 Evidence Pack Index Template

Every high-risk AI system should have an evidence pack index.

The index helps reviewers, auditors, governance teams and senior management locate the evidence needed to assess control.

Template C.2 — Evidence Pack Index

AI System Name:

Insert system name.

Provider:

Insert provider name.

Internal System Owner:

Insert owner.

Business Process Owner:

Insert business owner.

Governance Owner:

Insert governance / risk / compliance owner.

Evidence Pack Version:

Insert version.

Date Created:

Insert date.

Last Updated:

Insert date.

Next Review Date:

Insert date.

Evidence Item	Location / Link	Owner	Last Updated	Status
AI System Record				Complete / Partial / Missing
Provider Instructions for Use				Complete / Partial / Missing
Intended Purpose and Actual Use				Complete / Partial / Missing
Map				
Risk Classification				Complete / Partial /

Evidence Item	Location / Link	Owner	Last Updated	Status
Workflow Map				Missing
				Complete /
				Partial /
Human Oversight Model				Missing
				Complete /
				Partial /
Human Oversight Training Records				Missing
				Complete /
				Partial /
Input Data Source Map				Missing
				Complete /
				Partial /
Input Data Freshness Rules				Missing
				Complete /
				Partial /
Context Management Record				Missing
				Complete /
				Partial /
Output Classification Model				Missing
				Complete /
				Partial /
Authority and Approval Matrix				Missing
				Complete /
				Partial /
Runtime Monitoring Plan				Missing
				Complete /
				Partial /
Alert and Escalation Rules				Missing
				Complete /
				Partial /
Log Retention Plan				Missing
				Complete /
				Partial /
Sample Decision Trails				Complete /
				Partial /

Evidence Item	Location / Link	Owner	Last Updated	Status
Incident				Missing
Response				Complete /
Procedure				Partial /
Provider				Missing
Contact Route				Complete /
				Partial /
				Missing
Workplace /				Complete /
Affected-				Partial /
Person				Missing
Information				
Assessment				
Reassessment				Complete /
Records				Partial /
				Missing
Board /				Complete /
Governance				Partial /
Review				Missing
Records				

Evidence Pack Status

Overall Evidence Pack Status:

Green / Amber / Red / Grey.

Summary of Gaps:

Describe missing or weak evidence.

Immediate Action Required:

Yes / No.

Action Owner:

Insert owner.

C.3 AI System Record Template

The AI System Record is the foundation of the evidence pack.

It identifies the system, its purpose, ownership and deployment context.

Template C.3 — AI System Record

AI System Name:

Insert system name.

Provider / Vendor:

Insert provider.

System Description:

Briefly describe what the system does.

System Type:

Classifier / scoring system / ranking system / recommendation system / LLM / agentic system / decision-support system / monitoring system / other.

System Version:

Insert version.

Model Version:

Insert model version where applicable.

Deployment Environment:

Sandbox / pilot / controlled production / full production / suspended / retired.

Deployment Date:

Insert date.

Internal Owner:

Insert owner.

Technical Owner:

Insert owner.

Business Process Owner:

Insert owner.

Governance Owner:

Insert owner.

Provider-Defined Intended Purpose:

Describe provider intended purpose.

Internal Approved Use:

Describe approved internal use.

Actual Use:

Describe actual use in practice.

High-Risk Classification:

High-risk / high-consequence / not high-risk / pending assessment.

Reason for Classification:

Explain briefly.

Affected Workflow:

Describe workflow.

Affected Persons or Entities:

Workers / candidates / customers / patients / citizens / students / suppliers / clients / internal users / other.

Output Type:

Summary / score / ranking / recommendation / alert / draft / decision support / workflow trigger / tool action / other.

Output Authority Level:

Draft-only / advisory / decision support / controlled trigger / not authorized for consequence.

Downstream Systems:

List systems receiving or acting on output.

Known Limitations:

List limitations.

Prohibited Uses:

List prohibited uses.

Reassessment Triggers:

List triggers.

Current Governance State:

ALLOW / HOLD / ESCALATE / REFUSE / SUSPENDED.

C.4 Intended Purpose and Actual Use Map

This template shows whether actual use remains within approved purpose.

Template C.4 — Intended Purpose and Actual Use Map

Item

Description

Provider Intended Purpose

Internal Approved Use

Item	Description
Actual Current Use	
Approved User Groups	
Actual User Groups	
Approved Data Sources	
Actual Data Sources	
Approved Output Role	
Actual Output Role	
Approved Consequence Level	
Actual Consequence Level	
Approved Downstream Systems	
Actual Downstream Systems	
Prohibited Uses	
Scope Drift Detected?	Yes / No / Unknown
Reassessment Required?	Yes / No
Governance State	ALLOW / HOLD / ESCALATE / REFUSE

Use Assessment

Does actual use match approved use?

Yes / No / Partial / Unknown.

If no or partial, describe gap:

Insert explanation.

Decision:

ALLOW / HOLD / ESCALATE / REFUSE.

Decision Reason:

Insert reason.

Owner:

Insert owner.

Date:

Insert date.

C.5 Instructions-to-Control Mapping Form

Provider instructions should become internal controls.

This template maps documentation to operation.

Template C.5 — Instructions-to-Control Mapping

Provider

Instruction / Limitation	Operational Meaning	Internal Control	Evidence Produced	Owner	Status
					Active / Partial / Missing
					Active / Partial / Missing
					Active / Partial / Missing

Example Entries

Provider

Instruction / Limitation	Operational Meaning	Internal Control	Evidence Produced	Owner	Status
Use only for decision support	Output cannot make final decision	Human review before consequenc e	Review timestamp and reviewer record	Process Owner	Active
Use current source data	Input must meet freshness threshold	Freshness check before output use	Data freshness log	Data Owner	Active
User must understand limitations	Reviewer requires training	System- specific training module	Training completion record	Governance Owner	Active

Mapping Review

Instructions Reviewed By:

Insert person / role.

Date Reviewed:

Insert date.

Instructions Version:

Insert version.

Controls Implemented?

Yes / No / Partial.

Gaps Identified:

Insert gaps.

Action Required:

Insert action.

C.6 Human Oversight Evidence Form

This template records whether human oversight is meaningful.

Template C.6 — Human Oversight Evidence Form

AI System:

Insert system name.

Workflow:

Insert workflow.

Oversight Role:

Insert role.

Named Reviewer or Responsible Role:

Insert name / role.

Backup Reviewer:

Insert backup.

Reviewer Competence Confirmed:

Yes / No / Partial.

Competence Evidence:

Role description / qualification / domain experience / assessment / other.

Reviewer Training Completed:

Yes / No / Partial.

Training Evidence:

Training record / refresher record / practical assessment / other.

Reviewer Authority Confirmed:

Yes / No / Partial.

Authority Evidence:

Approval matrix / delegation record / workflow permission / other.

Reviewer Support Confirmed:

Yes / No / Partial.

Support Evidence:

Evidence access / interface support / time allocation / escalation route / workload metrics.

Can Reviewer Override?

Yes / No.

Can Reviewer Hold Workflow?

Yes / No.

Can Reviewer Escalate?

Yes / No.

Can Reviewer Refuse Continuation?

Yes / No.

Review Occurs Before Consequence?

Yes / No / Not applicable.

Evidence Available to Reviewer:

Source data / AI output / warnings / freshness state / policy version / authority state / downstream consequence.

Oversight Quality Monitoring:

Approval rate / override rate / escalation rate / review time / backlog / audit sample.

Current Oversight State:

ALLOW / HOLD / ESCALATE / REFUSE.

Reason:

Insert reason.

C.7 Case-Level Human Review Record

This form is used for specific AI-supported cases.

Template C.7 — Case-Level Human Review Record

Case ID:

Insert case ID.

AI System:

Insert system.

Workflow:

Insert workflow.

AI Output ID:

Insert output ID.

Output Type:

Summary / score / ranking / recommendation / alert / draft / decision support / other.

Consequence Level:

Low / moderate / high / critical.

Reviewer Name / Role:

Insert reviewer.

Review Timestamp:

Insert timestamp.

Consequence Timestamp:

Insert timestamp.

Review Before Consequence?

Yes / No.

Reviewer Had Authority?

Yes / No / Partial / Unknown.

Evidence Shown to Reviewer:

List evidence.

Warnings Shown:

List warnings.

Input Data Freshness Status:

Current / stale / unknown / not applicable.

Context Status:

Inspectable / partial / not inspectable / not applicable.

Reviewer Action:

Approve / modify / reject / request more evidence / hold / escalate / refuse.

Reviewer Comment:

Insert short reason.

Override Recorded?

Yes / No / Not applicable.

Escalation Triggered?

Yes / No.

Final State:

ALLOW / HOLD / ESCALATE / REFUSE.

Evidence Location:

Insert location.

C.8 Input Data Evidence Form

This template records input data suitability.

Template C.8 — Input Data Evidence Form

AI System:

Insert system.

Workflow:

Insert workflow.

Data Owner:

Insert owner.

Data Source	Source Type	Owner	Authoritative?	Last Updated	Retrieved At	Freshness Threshold	Status
	Internal / external / user-entered		Yes / No / Unknown				Current / stale / unknown

Data Source	Source Type	Owner	Authoritative?	Last Updated	Retrieved At	Freshness Threshold	Status
	/ third-party						

Data Quality Assessment

Required Fields Present?

Yes / No / Partial.

Missing Required Fields:

List fields.

Data Relevant to Intended Purpose?

Yes / No / Partial / Unknown.

Data Sufficiently Representative Where Relevant?

Yes / No / Partial / Not assessed / Not applicable.

Known Bias or Limitation:

Describe.

User-Entered Data Included?

Yes / No.

User-Entered Data Verified?

Yes / No / Partial / Not applicable.

Conflicting Sources Detected?

Yes / No.

Data Warnings Displayed?

Yes / No.

Input Data Governance State:

ALLOW / HOLD / ESCALATE / REFUSE.

Reason:

Insert reason.

C.9 Context Evidence Form for LLM and Agentic Systems

This template records what context the system received.

Template C.9 — Context Evidence Form

Context Bundle ID:

Insert ID.

AI System:

Insert system.

Workflow:

Insert workflow.

Case ID:

Insert case ID where applicable.

Prompt / Instruction Type:

System instruction / user prompt / workflow instruction / agent plan / tool instruction / other.

Context Sources Included:

Source	Type	Version	Last Updated	Retrieved At	Status
	Policy / record / document / database / tool output / summary				Current / stale / unknown

Policy Version Included:

Insert policy version.

Provider Instruction Included in Context?

Yes / No / Partial / Not applicable.

Authority State Included?

Yes / No / Partial.

Human Oversight Requirement Included?

Yes / No / Partial.

Limitations Included?

Yes / No / Partial.

Freshness Metadata Included?

Yes / No / Partial.

Context Compaction Used?

Yes / No.

Compaction Method:

Summary / memory / retrieval compression / other.

Governance-Critical Elements Preserved:

Source references / warnings / uncertainty / freshness / authority / pending checks / refusal conditions.

Potentially Lost Context:

Describe.

Context Inspectable?

Yes / No / Partial.

Context Governance State:

ALLOW / HOLD / ESCALATE / REFUSE.

Reason:

Insert reason.

C.10 AI Output Evidence Form

This template records the AI-generated output and its allowed role.

Template C.10 — AI Output Evidence Form

Output ID:

Insert ID.

AI System:

Insert system.

Case ID / Workflow ID:

Insert ID.

Output Timestamp:

Insert timestamp.

Output Type:

Summary / score / ranking / recommendation / alert / draft / decision support / workflow trigger / tool action.

Output Content Location:

Insert location.

Sources Cited:

Insert source references.

Confidence / Uncertainty Indicator:

Insert where available.

Warnings or Limitations Displayed:

Insert.

Contradictions Detected?

Yes / No / Unknown.

Unsupported Assumptions Detected?

Yes / No / Unknown.

Output Authority Level:

Draft-only / advisory / decision support / controlled trigger / not authorized.

May Influence Downstream Process?

Yes / No / Unknown.

Downstream Process:

Insert where applicable.

Human Review Required?

Yes / No / Conditional.

Output Governance State:

ALLOW / HOLD / ESCALATE / REFUSE.

Reason:

Insert reason.

C.11 Authority and Approval Form

This template records whether the workflow had valid authority.

Template C.11 — Authority and Approval Form

AI System:

Insert system.

Workflow:

Insert workflow.

Case ID:

Insert case ID.

Consequence Level:

Low / moderate / high / critical.

Output Role:

Draft / advisory / recommendation / decision support / trigger / action.

Authority Holder:

Insert person / role / committee / system rule.

Authority Type:

Review / approval / override / escalation / execution / suspension / refusal.

Authority Evidence:

Approval matrix / delegation / role permission / policy / workflow rule.

Authority Valid at Time of Use?

Yes / No / Partial / Unknown.

Threshold Applied:

Budget / risk / clinical / employment / legal / public-service / security / other.

Threshold Exceeded?

Yes / No / Unknown.

Higher Approval Required?

Yes / No.

Downstream Action Authorized?

Yes / No / Not applicable.

Authority Governance State:

ALLOW / HOLD / ESCALATE / REFUSE.

Reason:

Insert reason.

C.12 Runtime Monitoring Evidence Form

This template records monitoring status and alerts.

Template C.12 — Runtime Monitoring Evidence Form

AI System:

Insert system.

Monitoring Period:

Insert period.

Monitoring Owner:

Insert owner.

Signal Category	Monitored?	Current Status	Alert Threshold	Current Alert
Technical Health	Yes / No	Green / Amber / Red		
Model / Version	Yes / No	Green / Amber / Red		
Input Data	Yes / No	Green / Amber / Red		
Context	Yes / No	Green / Amber / Red		
Human Oversight	Yes / No	Green / Amber / Red		
Authority	Yes / No	Green / Amber / Red		
Scope / Intended Purpose	Yes / No	Green / Amber / Red		
Output Quality	Yes / No	Green / Amber / Red		
Downstream Effect	Yes / No	Green / Amber / Red		
Affected-Person Signals	Yes / No	Green / Amber / Red		
Incident / Near Miss	Yes / No	Green / Amber / Red		
Evidence Integrity	Yes / No	Green / Amber / Red		

Open Alerts:

List alerts.

Systems / Workflows in HOLD:

List.

Systems / Workflows in ESCALATE:

List.

Systems / Workflows in REFUSE:

List.

Monitoring Governance State:

ALLOW / HOLD / ESCALATE / REFUSE.

Action Required:

Insert action.

C.13 Incident Triage Form

This template supports incident and serious-risk triage.

Template C.13 — Incident Triage Form

Incident ID:

Insert ID.

Date / Time Detected:

Insert timestamp.

Detected By:

User / monitoring / reviewer / affected person / provider / audit / other.

AI System:

Insert system.

Workflow Affected:

Insert workflow.

Case ID(s):

Insert case IDs.

Incident Type:

Technical malfunction / data issue / context issue / human oversight issue / authority issue / output quality issue / downstream automation issue / affected-person complaint / serious risk / near miss / other.

Description:

Describe what happened.

Immediate Consequence:

Describe known consequence.

Potential Consequence:

Describe possible consequence.

Affected Persons or Entities:

Workers / candidates / customers / patients / citizens / students / suppliers / clients / other.

System Still Active?

Yes / No.

Immediate HOLD Required?

Yes / No.

Suspension Required?

Yes / No / Unknown.

Evidence Preserved?

Yes / No / Partial.

Provider Contact Required?

Yes / No / Unknown.

Legal / Compliance Review Required?

Yes / No / Unknown.

Serious Incident Assessment Required?

Yes / No / Unknown.

Initial Governance State:

ALLOW / HOLD / ESCALATE / REFUSE / SUSPEND.

Triage Owner:

Insert owner.

Next Review Time:

Insert time.

C.14 Evidence Preservation Checklist

Use this checklist when an incident, near miss, complaint or serious risk signal occurs.

Template C.14 — Evidence Preservation Checklist

Evidence Item	Preserved?	Location	Owner
AI system name and version	Yes / No		

Evidence Item	Preserved?	Location	Owner
Model version / fallback model	Yes / No		
Provider status	Yes / No		
User prompt / instruction	Yes / No		
System prompt / workflow instruction	Yes / No		
Context bundle	Yes / No		
Retrieved sources	Yes / No		
Input data condition	Yes / No		
Data freshness timestamps	Yes / No		
AI output	Yes / No		
Tool calls / agent steps	Yes / No		
Human review record	Yes / No		
Authority state	Yes / No		
Monitoring alerts	Yes / No		
Downstream action logs	Yes / No		
External communication	Yes / No		
Affected cases	Yes / No		
Error messages	Yes / No		
Screenshots where useful	Yes / No		
Provider communication	Yes / No		

Evidence Item	Preserved?	Location	Owner
Corrective action record	Yes / No		

Evidence Preservation Complete?
Yes / No / Partial.

If Partial, Explain Gap:
Insert explanation.

Evidence Integrity Protection Applied?
Yes / No.

Owner:
Insert owner.

C.15 Suspension and Return-to-Service Form

This template records suspension and controlled restart decisions.

Template C.15 — Suspension and Return-to-Service Form

AI System:

Insert system.

Workflow Affected:

Insert workflow.

Suspension Type:

Full suspension / partial suspension / reduced authority / manual fallback.

Suspension Start Time:

Insert timestamp.

Suspension Approved By:

Insert person / role.

Reason for Suspension:

Describe.

Affected Users / Workflows / Cases:

List.

Immediate Controls Applied:

Disable system / block tool / pause workflow / manual review / provider contact / other.

Evidence Preserved?

Yes / No / Partial.

Root Cause Identified?

Yes / No / Partial.

Corrective Action Completed?

Yes / No / Partial.

Testing Completed?

Yes / No / Not applicable.

Human Oversight Restored?

Yes / No / Not applicable.

Input Data Validity Restored?

Yes / No / Not applicable.

Monitoring Updated?

Yes / No / Not applicable.

Users Informed?

Yes / No / Not applicable.

Return-to-Service Conditions:

List conditions.

Return-to-Service Approved By:

Insert person / role.

Return-to-Service Time:

Insert timestamp.

Residual Risk:

Describe.

Next Review Date:

Insert date.

C.16 Affected-Person Information Record

This template records information duties and related communication.

Template C.16 — Affected-Person Information Record

AI System:

Insert system.

Workflow:

Insert workflow.

Affected Person Group:

Worker / candidate / customer / patient / citizen / student / supplier / other.

AI Role in Workflow:

Summarization / scoring / ranking / recommendation / routing / decision support / monitoring / other.

Information Duty Assessed?

Yes / No.

Information Required?

Yes / No / Unknown.

Notice Provided?

Yes / No.

Notice Date:

Insert date.

Notice Version:

Insert version.

Notice Content Location:

Insert location.

Does Notice Match Actual Use?

Yes / No / Partial / Unknown.

Human Review Route Explained?

Yes / No / Not applicable.

Challenge / Question Route Explained?

Yes / No / Not applicable.

Worker Representative Informed Where Required?

Yes / No / Not applicable.

Complaints / Challenges Received?

Yes / No.

Complaint Record Location:

Insert location.

Information Governance State:

ALLOW / HOLD / ESCALATE / REFUSE.

Reason:

Insert reason.

C.17 Reassessment Trigger Form

This template records when a system needs reassessment.

Template C.17 — Reassessment Trigger Form

AI System:

Insert system.

Trigger Date:

Insert date.

Trigger Type:

Provider instruction update / model version change / fallback event / new workflow / new data source / new affected-person group / downstream integration / incident / near miss / complaint / authority change / monitoring alert / policy change / other.

Description of Trigger:

Insert description.

Affected Workflow:

Insert workflow.

Potential Consequence Change:

Low / moderate / high / critical / unknown.

Current Use Paused?

Yes / No.

HOLD Applied?

Yes / No.

Escalation Required?

Yes / No.

Reassessment Owner:

Insert owner.

Evidence Needed:

List evidence.

Decision Deadline:

Insert date.

Reassessment Outcome:

Continue / continue with conditions / reduce authority / hold / suspend / refuse.

Decision Reason:

Insert reason.

Next Review:

Insert date.

C.18 Sample Completed Form: Procurement AI

Sample C.18 — Supplier Recommendation Evidence Record

AI System:

Supplier Risk Assistant.

Workflow:

Supplier approval.

Case ID:

PROC-2026-014.

Output Type:

Supplier risk recommendation.

Consequence Level:

High.

Provider Intended Purpose:

Supplier risk summarization and advisory recommendation.

Internal Approved Use:

Advisory supplier risk summary for procurement reviewer.

Actual Use:

Supplier recommendation prepared for approval workflow.

Input Data Status:

Financial data current. Sanctions check current. Cybersecurity review pending.

Human Oversight:

Procurement risk manager reviewed output before final approval.

Reviewer Authority:

Authority for advisory review. Final approval requires procurement committee.

Monitoring Alert:

Cybersecurity review pending.

Downstream Consequence:

Supplier could be shortlisted, but not approved.

Governance State:

HOLD.

Reason:

Final supplier approval cannot proceed until cybersecurity review is complete.

Action:

Escalated to information security team.

Evidence Preserved:

AI output, supplier data, freshness record, reviewer note, escalation record.

Next Trigger:

Resume after cybersecurity review.

C.19 Sample Completed Form: HR AI

Sample C.19 — Candidate Ranking Evidence Record

AI System:

Candidate Screening Assistant.

Workflow:

Recruitment shortlist.

Case ID:

HR-2026-032.

Output Type:

Candidate ranking.

Consequence Level:

High.

Provider Intended Purpose:

Candidate CV summarization.

Internal Approved Use:

CV summary for recruiter review.

Actual Use:

Ranking used to prioritize shortlist.

Input Data Status:

Current.

Human Oversight:

Recruiter reviewed top-ranked candidates.

Reviewer Authority:

Recruiter may shortlist candidates but must review source CVs.

Monitoring Alert:

Actual use drift from summarization to ranking.

Downstream Consequence:

Candidates may be excluded from interview.

Governance State:

ESCALATE.

Reason:

Ranking use exceeds approved output authority.

Action:

Ranking use paused pending HR governance review.

Evidence Preserved:

AI ranking, source CVs, reviewer record, scope-drift alert, escalation record.

Next Trigger:

Governance decision on ranking approval and controls.

C.20 Sample Completed Form: Clinical AI

Sample C.20 — Clinical Summary Evidence Record

AI System:

Clinical Summary Assistant.

Workflow:

Patient record review.

Case ID:

CLIN-2026-009.

Output Type:

Clinical summary.

Consequence Level:

Critical.

Provider Intended Purpose:

Clinical summarization for clinician review.

Internal Approved Use:

Summary only; not diagnosis or treatment decision.

Actual Use:

Summary prepared for clinician before care pathway decision.

Input Data Status:

Stale. Latest lab results missing.

Human Oversight:

Clinician review required.

Reviewer Authority:

Clinician has final clinical responsibility.

Monitoring Alert:

Missing latest lab data.

Downstream Consequence:

Summary may influence care pathway.

Governance State:

HOLD / REFUSE for clinical reliance.

Reason:

Summary cannot be treated as complete without latest labs.

Action:

Retrieve updated labs and regenerate summary.

Evidence Preserved:

Original summary, missing-data warning, clinician note, refresh request.

Next Trigger:

Updated summary after latest labs included and clinician review completed.

C.21 Evidence Quality Rating Form

This form helps assess whether the evidence is strong enough.

Template C.21 — Evidence Quality Rating

AI System:

Insert system.

Workflow:

Insert workflow.

Evidence Domain	Rating	Notes
System Identity	0 / 1 / 2 / 3 / 4 / 5	
Intended Purpose	0 / 1 / 2 / 3 / 4 / 5	
Human Oversight	0 / 1 / 2 / 3 / 4 / 5	
Input Data	0 / 1 / 2 / 3 / 4 / 5	
Context	0 / 1 / 2 / 3 / 4 / 5 / N/A	
Output Record	0 / 1 / 2 / 3 / 4 / 5	
Authority	0 / 1 / 2 / 3 / 4 / 5	
Monitoring	0 / 1 / 2 / 3 / 4 / 5	
Logs / Decision Trail	0 / 1 / 2 / 3 / 4 / 5	
Escalation	0 / 1 / 2 / 3 / 4 / 5	
Information Duties	0 / 1 / 2 / 3 / 4 / 5 / N/A	

Rating Scale

0 — No Evidence

No evidence available.

1 — Assertion

Control is claimed but not evidenced.

2 — Documented Process

Policy or procedure exists.

3 — Operational Record

Control operated and produced records.

4 — Reconstructable Evidence

Decision trail can be reconstructed.

5 — Runtime-Assured Evidence

Control is monitored and connected to ALLOW / HOLD / ESCALATE / REFUSE states.

Overall Evidence Quality:

0 / 1 / 2 / 3 / 4 / 5.

Critical Gaps:

Insert gaps.

Decision:

ALLOW / HOLD / ESCALATE / REFUSE.

C.22 Appendix C Summary

This appendix provides practical evidence pack templates and sample forms for Article 26 deployer governance.

The templates help organizations document:

- AI system identity;
- intended purpose and actual use;
- provider instructions mapped to internal controls;
- human oversight competence, training, authority and support;
- input data condition;
- LLM and agentic context;
- AI output authority;
- approval and authority state;
- runtime monitoring;
- incident triage;
- evidence preservation;
- suspension and return to service;
- affected-person information;
- reassessment triggers;
- evidence quality.

The appendix also provides sample completed forms for procurement, HR and clinical workflows.

The central purpose is to make evidence of control practical.

A deployer should not only know that an AI output was produced.

It should know:

- what system produced it;
- what data and context shaped it;

- whether the use was approved;
- whether human oversight operated;
- whether authority was valid;
- whether monitoring showed alerts;
- whether the output became consequence;
- and why the workflow was allowed, held, escalated or refused.

That is the practical function of an Article 26 evidence pack.

Appendix D

Glossary of Key Terms

Legal anchor and framework boundary. [Article 3](#). Terms taken from the Regulation should be read according to the official definitions; framework terms are identified as operational concepts.

D.1 Purpose of This Glossary

This glossary defines key terms used throughout this study.

The purpose is not to replace legal definitions in Regulation (EU) 2024/1689 or sector-specific regulatory guidance.

The purpose is to provide a practical operational vocabulary for Article 26 deployer governance.

Many AI governance failures occur because organizations use the same words differently.

Terms such as “oversight,” “monitoring,” “control,” “risk,” “decision support,” “human review,” “evidence,” and “accountability” can sound clear while meaning different things to legal, compliance, technology, business and audit teams.

This glossary supports a shared language.

It is designed for deployers, governance teams, board members, internal auditors, compliance officers, risk managers, technical teams, human overseers and operational users.

D.2 Key Terms

Accountability

The ability to identify who or what was responsible for a system, workflow, decision, review, escalation or control action.

In this study, accountability requires evidence.

A person or function may be named as accountable, but accountability is weak if the organization cannot reconstruct what happened, who reviewed it, what authority applied and why continuation was allowed.

Actual Use

The way an AI system is used in practice.

Actual use may differ from provider-defined intended purpose or internal approved use.

This is one of the most important governance risks.

A system approved for summarization may be used for ranking.

A system approved for drafting may be used for external communication.

A system approved as advisory may become a decision trigger.

Actual use must be monitored.

Affected Person

A natural person whose rights, opportunities, access, treatment, eligibility, employment, healthcare, education, public-service pathway or other significant interest may be affected by an AI-supported workflow.

Examples include:

- job candidates;
- workers;
- patients;
- citizens;
- students;
- customers;
- service users;
- benefit applicants.

Affected-person information, challenge routes and case reconstruction may be required depending on the context.

Agentic AI

An AI system or AI-enabled workflow that can perform multi-step tasks with some degree of autonomy.

Agentic AI may:

- plan;
- retrieve data;
- call tools;
- generate recommendations;
- update systems;
- trigger workflows;
- route tasks;
- send messages;
- continue across several steps.

Agentic AI changes deployer governance because the risk may occur across transitions, not only in the final output.

AI Output

The result produced by an AI system.

An AI output may include:

- classification;
- score;
- ranking;
- summary;
- recommendation;
- prediction;
- generated content;
- alert;
- draft;
- decision-support note;
- tool call;
- action plan;
- workflow trigger.

In this study, an output becomes governance-relevant when it influences consequence.

AI System Register

An internal inventory of AI systems used by the organization.

A mature register should include:

- system name;
- provider;
- owner;
- intended purpose;
- actual use;
- risk classification;
- affected workflow;
- affected persons;
- model or system version;
- deployment status;
- reassessment date.

A register is necessary, but not sufficient. It must connect to evidence of control.

ALLOW

A runtime governance state meaning that current control conditions support continuation.

ALLOW should be based on evidence, not assumption.

Typical ALLOW conditions include:

- use is within intended purpose;
 - input data is suitable;
 - human oversight is completed where required;
 - authority is valid;
 - monitoring shows no unresolved critical alert;
 - logs support reconstruction;
 - downstream consequence is controlled.
-

Approval

A formal or operational permission for a workflow, decision, use case or action to continue.

Approval is not always permanent.

Approval should be understood as valid only under defined conditions.

If those conditions change, approval should be reassessed.

Article 26

The article of the EU AI Act that sets out obligations for deployers of high-risk AI systems.

In this study, Article 26 is interpreted operationally as a runtime governance model covering:

- use according to instructions;
 - human oversight;
 - input data responsibility;
 - monitoring;
 - log retention;
 - risk and incident response;
 - workplace information;
 - affected-person information;
 - public authority and related obligations where applicable.
-

Authority

The legitimate power to approve, reject, escalate, override, execute, suspend or refuse an AI-supported workflow.

Authority may belong to:

- a human reviewer;
- a manager;
- a committee;
- a legal or compliance function;
- a clinician;
- a public official;
- a system rule;
- a policy threshold.

Authority must be current and matched to the consequence level.

Authority Continuity

The idea that authority must remain valid when conditions change.

Authority continuity may be broken by:

- model fallback;
- output becoming a downstream trigger;
- workflow expansion;
- role change;
- approval expiry;
- policy change;
- consequence increase.

Fallback should not preserve authority silently.

Authority Drift

A situation where the practical authority of an AI output increases over time without formal reassessment.

Examples:

an advisory recommendation becomes routinely followed;

a summary becomes the basis for final decision;

a score becomes a threshold;

a draft becomes external communication;

a ranking becomes candidate exclusion.

Authority drift is a major runtime governance risk.

Automation Bias

A tendency for humans to over-rely on AI outputs because they appear objective, structured, confident or efficient.

Automation bias may increase when:

reviewers are under time pressure;

source evidence is hard to access;

the interface nudges approval;

override is discouraged;

AI outputs are usually correct;
review becomes routine.

Automation bias weakens human oversight.

Board Briefing

A concise executive-level report showing AI governance status, control maturity, unresolved risks and decisions required.

A strong board briefing should show:

high-risk AI systems;
high-consequence workflows;
systems in ALLOW / HOLD / ESCALATE / REFUSE;
evidence gaps;
incidents and near misses;
remediation actions;
decisions required.

Case-Level Evidence

Evidence relating to a specific AI-supported workflow instance, decision, recommendation, output or affected-person case.

Case-level evidence may include:

input data condition;
AI output;
context record;
human review;
authority state;
monitoring alert;
downstream consequence;
final governance state.

Case-level evidence is essential for reconstructability.

Compliance

Meeting legal, regulatory, policy or internal requirements.

In this study, compliance is treated as necessary but not sufficient.

The strongest deployer governance moves from compliance documentation to evidence of control.

Consequence

The point at which an AI output begins to affect a person, process, right, opportunity, operational system, regulated workflow or institutional decision.

Examples of consequence include:

candidate shortlisted or rejected;
patient pathway influenced;
supplier approved;
citizen case routed;
customer message sent;
compliance case closed;
official record updated;
workflow trigger activated.

Governance should focus on control before consequence.

Consequence Point

The specific workflow stage where AI output begins to matter operationally.

Examples:

before final approval;
before external communication;
before case closure;
before shortlist creation;
before clinical pathway influence;
before public-service routing;
before a tool executes a write action.

Human control should be placed before consequence points.

Context

The information supplied to an AI system or agent during operation.

Context may include:

- user prompt;
- system instruction;
- documents;
- retrieved records;
- policies;
- tool outputs;
- previous conversation history;
- summaries;
- workflow state;
- authority state;
- human oversight requirements.

Context shapes what the AI system can reason about and therefore functions as a control surface.

Context Compaction

The process of compressing long context into summaries or shorter representations.

Context compaction can reduce cost and improve performance, but it can also remove governance-critical information such as:

- warnings;
- uncertainty;
- source references;
- timestamps;
- missing evidence;
- authority limits;
- pending checks;
- refusal conditions.

High-risk workflows should preserve governance-critical context during compaction.

Context Hygiene

The discipline of keeping AI context relevant, current, authorized and useful.

Good context hygiene avoids:

- stale facts;
- duplicate records;
- irrelevant history;
- outdated policy;
- unverified sources;
- conflicting context;
- old summaries;
- unauthorized material.

Context hygiene is not only a token-cost issue. It is a governance issue.

Context Record

A record showing what context was supplied to an AI system.

A context record may include:

- context bundle ID;
- source documents;
- retrieved records;
- policy version;
- timestamps;
- freshness metadata;
- compaction status;
- limitations;
- authority state;
- human oversight requirement.

Context records support reconstructability.

Control

A measure that prevents, detects, limits, escalates or corrects risk.

Controls may be:

technical;
procedural;
human;
organizational;
legal;
monitoring-based;
evidence-based.

In this study, a control is mature only when it produces evidence.

Control Before Consequence

The principle that governance checks, human review, authority validation and evidence assessment should occur before AI output affects real-world outcomes.

Control after consequence may support audit or correction, but it is weaker than preventive control.

Control Condition

A condition that must be true for an AI-supported workflow to continue.

Examples include:

input data is fresh;
reviewer is trained;
authority is valid;
use is within intended purpose;
logs are complete;
provider instructions are followed;
no unresolved critical alert exists.

If a control condition fails, the workflow may need to HOLD, ESCALATE or REFUSE.

Control Surface

The set of places where governance can influence an AI-supported workflow.

Control surfaces include:

data sources;

context;
prompts;
model routing;
tool access;
human review;
authority thresholds;
monitoring alerts;
logs;
downstream integrations;
escalation procedures.

Agentic AI expands the control surface.

Decision Trail

A reconstructable chain of evidence showing how an AI-supported output moved toward consequence.

A decision trail should connect:

AI system;
input data;
context;
output;
human review;
authority;
monitoring;
escalation;
downstream consequence;
final decision.

Technical logs alone may not be enough. The trail must show control.

Deployer

An organization or person using an AI system under its authority, except where the AI system is used in the course of a personal non-professional activity.

In this study, the deployer is the organization that places the AI system into operational context.

The deployer controls actual use.

Deployment Approval

A point-in-time decision that an AI system may be used under defined conditions.

Deployment approval is not permanent permission.

If actual use, data, workflow, model, provider, authority or consequence changes, approval may require reassessment.

Downstream Consequence

A consequence created when an AI output is consumed by another system, person or workflow.

Examples:

recommendation becomes task;
score becomes threshold;
summary becomes official record;
draft becomes external communication;
ranking affects shortlist;
alert triggers investigation.

Downstream consequences must be mapped.

Downstream Effect Map

A map showing where AI outputs travel and what they influence.

It should identify:

receiving systems;
users;
workflow triggers;
official records;
external communications;
affected persons;
regulated processes.

This map helps detect when advisory outputs become operational triggers.

ESCALATE

A runtime governance state meaning the issue requires review by a higher authority, specialist function, governance owner, provider or responsible decision-maker.

ESCALATE is appropriate when:

- use may be outside scope;
- authority is unclear;
- serious risk may exist;
- affected-person complaint is received;
- provider input is required;
- human oversight failure occurs;
- fallback changes authority conditions.

Evidence Architecture

The connected structure of records that allows an organization to prove AI governance controls operated.

Evidence architecture connects:

- system identity;
- intended purpose;
- actual use;
- provider instructions;
- input data;
- context;
- output;
- human oversight;
- authority;
- monitoring;
- escalation;
- downstream consequence;
- final decision.

Evidence architecture turns compliance into operational proof.

Evidence Chain

The connected sequence of evidence needed to reconstruct why an AI-supported workflow was allowed to continue.

An evidence chain may include:

- system record;
- input data record;
- output record;
- review record;
- authority record;
- monitoring record;
- consequence record.

Fragmented evidence is weaker than connected evidence.

Evidence of Control

Proof that a control operated when it mattered.

Evidence of control is stronger than:

- policy;
- intention;
- general assurance;
- checklist completion;
- verbal explanation.

Examples include:

- review before consequence;
- stale data triggered HOLD;
- fallback reduced authority;
- escalation record;
- refusal log;
- reconstructable decision trail.

Evidence Pack

A structured collection of records showing that an AI system is governed.

An Article 26 evidence pack may include:

AI system record;
provider instructions;
use map;
human oversight records;
data source map;
monitoring plan;
logs;
escalation procedure;
sample decision trails;
affected-person information records;
reassessment records.

Explainability

The ability to provide understandable information about how or why an AI output was produced.

In this study, explainability is useful but not enough.

High-risk governance also requires inspectability, evidence and control.

Fallback

Use of an alternative model, system, provider, workflow or manual process when the primary system is unavailable or unsuitable.

Fallback is not neutral.

A fallback model may not have the same validation, authority, logging, evidence quality or oversight assumptions as the primary model.

Fallback should be governed.

Fallback Authority Reduction

A control that reduces the authority of an AI output when fallback occurs.

For example:

full recommendation becomes draft-only;

automatic workflow trigger becomes human-reviewed;
high-consequence decision support is blocked;
output requires escalation.

This prevents fallback from silently preserving authority.

Fundamental Rights Impact Assessment

An assessment required in certain cases to evaluate potential impacts of high-risk AI systems on fundamental rights.

In this study, the FRIA is treated as part of the broader evidence architecture where applicable.

It should connect risks to operational controls and runtime monitoring.

Governance Signal

A runtime signal indicating whether AI use remains within valid control conditions.

Examples:

stale input data;
missing human review;
authority threshold exceeded;
model fallback;
context conflict;
scope drift;
missing logs;
affected-person complaint;
downstream trigger detected.

Governance signals should trigger action.

Grey Status

A governance status meaning unknown, unassessed or insufficient evidence.

Grey should not be treated as safe.

Unknown is a risk state.

High-Consequence AI System

An AI system that may significantly affect people, operations, legal obligations, regulated processes, safety, rights, access, opportunity, public trust or organizational risk, even if it is not formally classified as high-risk.

This study uses high-consequence as a practical governance concept.

High-Risk AI System

An AI system falling within high-risk categories under the EU AI Act.

High-risk AI systems require stronger governance, including deployer obligations under Article 26.

Organizations should also apply similar discipline to high-consequence systems where appropriate.

HOLD

A runtime governance state meaning that workflow continuation should pause because required control conditions are missing, stale, incomplete, uncertain or degraded.

HOLD is not failure.

It is controlled uncertainty.

Examples:

input data stale;
reviewer unavailable;
authority unclear;
context incomplete;
log gap detected.

Human-in-Control

A stronger form of human oversight where the human has practical ability to understand, challenge, delay, escalate, override or refuse an AI-supported workflow before consequence attaches.

Human-in-control requires:

competence;
training;
authority;
evidence access;
timing;
support;
documentation;
monitoring.

Human-in-the-Loop

A weaker and broader concept meaning that a human is present somewhere in the AI-supported process.

Human-in-the-loop does not necessarily mean the human has control.

A person may be in the loop but lack evidence, authority, time or intervention power.

Human Oversight

Human involvement designed to prevent or minimize risk and ensure appropriate use of the AI system.

In this study, human oversight is meaningful only if the human has:

competence;
training;
authority;
support;
evidence;
timing;
ability to intervene.

Incident

An event where an AI system or AI-supported workflow causes, contributes to or may contribute to harm, unsafe operation, unlawful consequence, rights impact, loss of control, serious error or governance failure.

Incidents require triage, evidence preservation and appropriate escalation.

Incident Response

The process for detecting, triaging, preserving evidence, escalating, correcting, suspending and reviewing AI-related incidents or near misses.

Incident response must be prepared before serious events occur.

Input Data

Data supplied to an AI system for processing.

Input data may include:

- structured records;
- documents;
- user-entered information;
- external data;
- internal system records;
- retrieved context;
- real-time signals.

Where input data is under deployer control, relevance, suitability, freshness and representativeness may be critical.

Input Data Freshness

The degree to which input data is current enough for the workflow and consequence level.

Freshness depends on context.

A sanctions check may need to be current at decision time.

A policy version must be current.

A patient record may require recent labs.

A supplier risk profile may require recent financial or cybersecurity data.

Inspectability

The ability to inspect enough of the AI-supported workflow to evaluate whether the output is supported, bounded and allowed to influence consequence.

Inspectable elements may include:

- data sources;
- context;
- reasoning support;
- rules;
- constraints;
- warnings;
- authority;
- human review;
- logs.

Inspectability is stronger than a fluent explanation.

Intended Purpose

The purpose for which the provider designed and supplied the AI system.

Deployers must compare intended purpose with internal approved use and actual use.

Use outside intended purpose may require escalation or refusal.

Instructions for Use

Provider-supplied information explaining how the system should be used, including intended purpose, limitations, user requirements, input requirements, output interpretation and operational conditions.

Instructions for use must be translated into internal controls.

Log

A record automatically or manually generated during system operation.

Logs may record:

system events;
inputs;
outputs;
tool calls;
user actions;
timestamps;
errors;
reviews;
downstream actions.

For governance, logs should support decision trails.

Log Integrity

The ability to trust that logs are complete, accurate, protected from alteration and retained appropriately.

Log integrity is essential for audit, investigation and reconstructability.

Model Dependency

Reliance on a specific model, provider, version or routing configuration.

Model dependency becomes a governance issue when model substitution may affect authority, validation, output quality, refusal behavior, data handling or logging.

Model Substitution

Replacement of one model with another.

Substitution may occur through:

fallback;
provider change;
version update;
routing decision;
cost optimization;
regional availability;
outage response.

Model substitution should trigger reassessment of authority and admissibility.

Monitoring

The process of observing system and governance conditions during operation.

Monitoring should include more than technical uptime.

It should cover:

- data;
- context;
- human oversight;
- actual use;
- authority;
- downstream effects;
- model fallback;
- logs;
- incidents;
- affected-person signals.

Near Miss

An event that could have caused harm, loss of control, unfair outcome, unauthorized action or governance failure, but did not result in actual consequence or was caught in time.

Near misses are valuable governance signals.

They should be recorded and reviewed.

Operational Memory

The organization's ability to remember and reconstruct what happened in AI-supported workflows.

Operational memory is created through evidence architecture.

It protects against institutional forgetting when people, systems, policies or providers change.

Output Authority Level

The level of influence an AI output is permitted to have.

Examples:

draft-only;
advisory;
decision support;
controlled workflow trigger;
external action;
not authorized.

Output authority should be explicitly defined.

Policy Version

The specific version of an internal or external policy used in an AI-supported workflow.

Policy version matters because a correct answer based on an outdated policy may be governance-invalid.

Policy version should be visible in context and decision trails.

Provider

The entity that develops or places an AI system on the market or puts it into service under its own name or trademark.

In practical governance, provider information is essential for instructions, limitations, updates, incident communication and system documentation.

REFUSE

A runtime governance state meaning the workflow should not proceed under current conditions.

REFUSE is appropriate where:

use is prohibited;
authority is absent;

human oversight is impossible;
evidence cannot support output;
serious risk remains unresolved;
fallback is not approved;
downstream consequence cannot be controlled.

Refusal is a control feature, not a system failure.

Reassessment

A review of whether an AI system or workflow remains valid under current conditions.

Reassessment may be triggered by:

provider update;
model change;
fallback event;
actual use drift;
new workflow;
new data source;
incident;
complaint;
authority change;
monitoring alert;
downstream integration.

Reconstructability

The ability to reconstruct what happened in an AI-supported workflow.

A reconstructable workflow should show:

system used;
input data;
context;
output;
human review;
authority;
monitoring alerts;

escalation;
downstream consequence;
final decision.

Reconstructability is central to evidence of control.

Reasoning Layer

A technical or governance component that helps expose or enforce structured reasoning conditions.

A reasoning layer may include:

knowledge graph;
rule engine;
validation layer;
constraint system;
policy engine;
evidence checker;
contradiction detector.

Reasoning layers can support traceability, refusal and human oversight.

Red Flag

A signal that AI governance may be weak or failing.

Examples:

no system owner;
use outside intended purpose;
no meaningful human oversight;
stale data;
missing logs;
model fallback without authority rules;
advisory output becoming executable;
affected-person complaint;
decision trail unreconstructable.

Red flags require review.

Runtime Governance

The continuous process of determining whether an AI-supported workflow remains allowed to continue under current conditions.

Runtime governance includes:

- observing conditions;
- comparing them with approved controls;
- deciding ALLOW / HOLD / ESCALATE / REFUSE;
- acting;
- recording;
- reassessing.

This study treats Article 26 as a runtime governance model.

Runtime Signal

A signal observed during operation that may indicate whether control conditions remain valid.

Examples:

- data stale;
- reviewer unavailable;
- model fallback;
- scope drift;
- context conflict;
- output contradiction;
- missing log;
- complaint received;
- downstream automation detected.

Runtime signals should connect to governance states.

Scope Drift

A situation where actual use gradually moves beyond approved use.

Examples:

- summarization becomes ranking;

drafting becomes sending;
advisory output becomes decision trigger;
internal use becomes external communication;
low-risk workflow becomes high-consequence workflow.

Scope drift requires reassessment.

Serious Incident

A significant incident category under the AI Act framework.

For deployer governance, any possible serious incident should trigger rapid triage, evidence preservation, escalation and provider or authority assessment where required.

Serious Risk

A signal that use of an AI system may create significant risk to health, safety, fundamental rights, legal obligations, affected persons or operational control.

Serious risk may require HOLD, ESCALATE, REFUSE or suspension.

Silent Continuation

A governance failure where an AI-supported workflow continues even though the conditions required for valid continuation have degraded or disappeared.

Examples:

stale data used;
reviewer absent;
fallback model unapproved;
output outside intended purpose;
authority expired;
logs missing;
downstream automation changed.

Silent continuation is one of the main risks this study addresses.

Stale Evidence

Information that remains visible but is no longer current enough to support the consequence attached to the AI output.

Stale evidence may include:

- old policy;
- expired approval;
- outdated supplier risk record;
- old patient data;
- previous account status;
- old eligibility rule.

Stale evidence should not carry current authority.

Suspension

A temporary stop, restriction or reduction of AI system use due to risk, incident, control failure or uncertainty.

Suspension may be:

- full;
- partial;
- workflow-specific;
- feature-specific;
- reduced-authority mode;
- manual fallback.

Return to service should require evidence.

Tool Call

An action by an AI system or agent to use an external function, API, database, system or service.

Tool calls may include:

- reading records;
- retrieving documents;
- updating systems;

sending messages;
creating tasks;
triggering workflows.

Tool calls expand the AI control surface.

Traceability

The ability to trace an AI-supported output back to its sources, context, rules, human review, authority and consequence.

Traceability supports oversight, audit, affected-person review and incident response.

Trustworthy AI

AI use that is valid, monitored, human-controllable, reconstructable and accountable.

In this study, trustworthy AI does not mean blind trust.

It means trust based on evidence.

Use Case

A specific way an AI system is used inside an organization.

A use case should define:

workflow;
users;
data;
output;
consequence;
human oversight;
authority;
monitoring;
evidence.

Use-case specificity is essential for deployer governance.

Workflow

A sequence of operational steps in which an AI system is used.

Examples:

recruitment screening;
supplier approval;
clinical summarization;
compliance case review;
public-service eligibility;
customer support response;
financial risk assessment.

AI governance must be mapped at workflow level, not only system level.

Workflow Execution

The movement of a workflow from input to consequence.

Agentic AI may participate in workflow execution by planning, tool use, routing, updating, recommending or triggering actions.

Workflow execution requires transition-level control.

D.3 Glossary Summary

This glossary provides a practical vocabulary for Article 26 deployer governance.

The central terms are connected by one core idea:

AI governance is not only about the system.

It is about controlled use.

Controlled use requires:

valid purpose;
suitable input data;
inspectable context;
meaningful human oversight;
valid authority;
monitoring;

logs;
escalation;
refusal where necessary;
reconstructable evidence.

The most important operational sentence remains:

A deployer should be able to prove that a high-risk AI system was used under valid, monitored, human-controllable and reconstructable conditions when its output influenced consequence.

This glossary supports that interpretation.

Appendix E

90-Day Implementation Roadmap

Legal anchor and framework boundary. [Article 26](#). The 90-day roadmap is a governance implementation sequence, not a statutory deadline.

E.1 Purpose of This Roadmap

This roadmap provides a practical 90-day implementation path for Article 26 deployer governance.

It is designed for organizations that need to move from general AI governance intention toward operational evidence of control.

The roadmap assumes that the organization may already have some AI policies, system inventories, vendor documents, risk processes or compliance procedures.

However, it also assumes that those elements may not yet be connected into a runtime governance model.

The goal of the first 90 days is not perfection.

The goal is to establish a working foundation.

By the end of 90 days, the organization should be able to show:

- which high-risk or high-consequence AI systems are in scope;
- who owns them;
- what workflows they affect;
- what outputs they produce;
- where consequence attaches;
- what human oversight exists;
- what input data controls exist;
- what monitoring signals matter;
- what logs are available;
- where evidence gaps exist;
- which workflows should ALLOW, HOLD, ESCALATE or REFUSE;
- what remediation actions are required.

The roadmap is divided into four phases:

- Days 1–15: Mobilize and identify.
- Days 16–30: Map and classify.

- Days 31–60: Build control evidence.
- Days 61–90: Test, remediate and report.

This roadmap should be adapted to organization size, sector, AI maturity and risk level.

A small organization may complete a simplified version.

A regulated organization may need a more formal implementation programme.

E.2 90-Day Outcome Targets

By Day 90, the organization should aim to have the following minimum outputs.

Governance Outputs

- AI deployer governance owner identified.
- Cross-functional implementation group established.
- High-risk and high-consequence AI systems identified.
- AI system register updated.
- Use-case owners assigned.
- Article 26 readiness matrix completed for priority systems.
- Board or executive summary prepared.

Control Outputs

- Intended purpose and actual use mapped.
- Provider instructions collected.
- Instructions-to-controls mapping started.
- Human oversight roles defined.
- Human oversight competence and training gaps identified.
- Input data source map created.
- Data freshness and missing-data rules drafted.
- Authority and approval map drafted.
- Downstream effect map started.

Evidence Outputs

- Evidence pack index created.
- Log inventory created.
- Sample decision trails tested.
- Evidence gaps identified.
- ALLOW / HOLD / ESCALATE / REFUSE rules drafted.

- Incident triage form prepared.
- Provider contact route documented.
- Reassessment triggers defined.

Operational Outputs

- Priority remediation plan created.
- Monitoring signal catalogue drafted.
- One incident response drill or tabletop exercise completed.
- At least three case samples tested for reconstructability.
- Systems requiring HOLD, ESCALATE or REFUSE identified.
- 90-day report completed.

The Day 90 target is not to declare every AI system mature.

The target is to know which systems are controlled, which are partially controlled, which are unknown and which should not continue under current conditions.

E.3 Phase 1 — Days 1–15: Mobilize and Identify

The first phase establishes ownership and visibility.

The organization cannot govern what it has not identified.

Objective

Create the implementation structure and identify AI systems that may fall within high-risk or high-consequence deployer governance.

Key Actions

1. Appoint an AI Deployer Governance Owner

Assign a person or function responsible for coordinating Article 26 deployer readiness.

Possible owners include:

- compliance;
- risk;
- legal;
- technology governance;
- AI governance office;
- data governance;
- operational risk;

- internal control function.

The owner does not need to perform every task.

But they must coordinate the evidence architecture.

2. Form a Cross-Functional Working Group

The working group should include relevant representatives from:

- legal;
- compliance;
- risk;
- technology;
- data;
- cybersecurity;
- HR;
- procurement;
- operations;
- internal audit;
- business units;
- clinical, financial, public-sector or sector-specific owners where relevant.

The goal is to connect legal interpretation to operational control.

3. Create an Initial AI System Inventory

Start with all known AI systems, including:

vendor AI systems;
internally built models;
embedded AI in enterprise software;
LLM tools;
agentic workflows;
HR tools;
procurement tools;
compliance tools;
customer support tools;
clinical or public-service tools;
workflow automation using AI outputs.

Do not wait for perfect classification.

List first.

Classify next.

4. Identify Priority Systems

Prioritize systems that may affect:

employment;
healthcare;
financial services;
public services;
education;
procurement;
access to essential services;
safety;
legal or regulated processes;
worker treatment;
affected natural persons;
downstream automated workflows.

These systems should be reviewed first.

5. Establish a Temporary Control Rule

Until assessment is complete, define a temporary rule:

No high-consequence AI output should be used as a final decision, external communication, automated trigger or official record unless a responsible human has reviewed it and the decision trail can be reconstructed.

This prevents uncontrolled expansion during the 90-day review.

Phase 1 Deliverables

By Day 15, produce:

AI deployer governance owner record.
Working group membership list.
Initial AI system inventory.
Priority system list.
Temporary control rule.
Initial board or executive awareness note.

Phase 1 Success Test

Can the organization name its priority AI systems and the people responsible for reviewing them?

If not, continue Phase 1 until ownership and visibility are established.

E.4 Phase 2 — Days 16–30: Map and Classify

The second phase maps how priority AI systems are actually used.

The focus is workflow reality, not only vendor description.

Objective

Classify priority AI systems and map intended purpose, actual use, affected workflows, outputs, data, humans, authority and consequence.

Key Actions

1. Collect Provider Instructions

For each priority system, collect:

- instructions for use;
- intended purpose;
- limitations;
- user requirements;
- input data requirements;
- output interpretation guidance;
- monitoring guidance;
- logging capability;
- incident contact route;
- update process;
- known failure modes.

If provider instructions are missing, record the gap.

2. Map Intended Purpose Versus Actual Use

Create a table showing:

- provider intended purpose;
- internal approved use;
- actual use;

- user groups;
- affected workflows;
- output type;
- consequence level;
- downstream systems;
- prohibited uses.

This is one of the most important 90-day tasks.

Many governance risks appear when actual use differs from intended or approved use.

3. Identify Consequence Points

For each workflow, identify where AI output begins to matter.

Examples:

- shortlist created;
- recommendation routed;
- case prioritized;
- official record updated;
- external message sent;
- clinical pathway influenced;
- supplier approved;
- compliance case closed.

These are the points where control must exist before consequence.

4. Classify Risk and Consequence

Classify each priority system as:

- high-risk;
- high-consequence;
- medium-risk;
- low-risk;
- unknown / pending.

Unknown should not be treated as safe.

Grey is a risk state.

5. Identify Human Oversight Model

For each system, ask:

- who reviews the output;
- whether they are trained;
- whether they have authority;
- whether they can access evidence;
- whether they can override;
- whether they can hold or escalate;
- whether review occurs before consequence.

6. Identify Input Data Sources

Map:

- source systems;
- data owners;
- source reliability;
- freshness requirements;
- missing-data rules;
- representativeness concerns where relevant;
- user-entered data;
- external data.

7. Map Downstream Effects

Identify where AI outputs travel.

Ask:

- who sees the output;
- whether another system consumes it;
- whether it becomes a structured field;
- whether it triggers tasks;
- whether it updates records;
- whether it reaches affected persons;
- whether it influences regulated processes.

Phase 2 Deliverables

By Day 30, produce:

- Provider instructions collection.
- Intended purpose and actual use maps.
- Priority workflow maps.
- Consequence point map.
- Initial risk and consequence classification.

Human oversight map.
 Input data source map.
 Downstream effect map.
 Initial gap list.

Phase 2 Success Test

Can the organization explain how each priority AI system is actually used and where its output begins to create consequence?

If not, the system is not yet governable.

E.5 Phase 3 — Days 31–60: Build Control Evidence

The third phase builds practical evidence structures.

The organization now moves from mapping to control.

Objective

Create evidence packs, control rules, monitoring signals and decision-state logic for priority AI systems.

Key Actions

1. Create Evidence Pack Indexes

For each priority system, create an evidence pack index.

At minimum, include:

- system record;
- provider instructions;
- intended purpose and actual use map;
- human oversight model;
- input data source map;
- authority map;
- monitoring plan;
- logging plan;
- incident route;
- sample decision trail;
- reassessment triggers.

The goal is to know where evidence lives.

2. Build Instructions-to-Control Mapping

For each important provider instruction, define:

operational meaning;
internal control;
evidence produced;
owner;
status.

Example:

Provider instruction: human review required before final decision.

Internal control: workflow cannot move to final state without reviewer approval.

Evidence: review timestamp, reviewer role, action record.

3. Define Human Oversight Evidence

For each priority workflow, document:

reviewer role;
competence criteria;
training status;
authority;
evidence access;
override ability;
escalation route;
review timing;
workload concerns.

Any gaps should become remediation items.

4. Define Input Data Control Rules

For each data source, define:

freshness threshold;
completeness requirement;
missing-data action;
source reliability;
conflict handling;
data owner;
HOLD conditions.

Example:

Supplier sanctions check older than defined threshold → HOLD.

Clinical record missing latest labs → HOLD.

Policy version unknown → HOLD.

5. Define Authority and Boundary Rules

For each workflow, define:

- what the AI output may do;
- what it may not do;
- who can approve;
- what thresholds apply;
- when escalation is required;
- when refusal is required;
- whether downstream automation is allowed.

This prevents advisory outputs from silently becoming execution signals.

6. Create Monitoring Signal Catalogue

Create a catalogue of runtime signals, including:

- technical health;
- model version;
- model fallback;
- input data freshness;
- context integrity;
- human oversight availability;
- authority state;
- actual use drift;
- downstream effect;
- output quality;
- affected-person complaints;
- log integrity.

Each signal should have an owner and action.

7. Define ALLOW / HOLD / ESCALATE / REFUSE Rules

For each priority workflow, define what conditions produce each state.

Example:

ALLOW: data fresh, reviewer trained, authority valid, logs complete.

HOLD: data stale, reviewer unavailable, context incomplete.

ESCALATE: use outside approved scope, authority threshold exceeded, affected-person complaint.

REFUSE: prohibited use, required oversight impossible, fallback not approved, serious risk unresolved.

8. Build Log and Decision Trail Requirements

Define what must be logged:

system ID;
 model version;
 input data condition;
 context bundle where relevant;
 AI output;
 human review;
 authority;
 monitoring alerts;
 downstream consequence;
 final state.

Then test whether current systems can produce this evidence.

9. Create Incident Triage and Evidence Preservation Forms

Use Appendix C forms as the base.

At minimum, define:

who triages;
 who preserves evidence;
 who contacts provider;
 who can suspend use;
 who approves return to service;
 how affected cases are identified.

Phase 3 Deliverables

By Day 60, produce:

Evidence pack index for each priority system.

Instructions-to-controls mapping.

Human oversight evidence form.

Input data control rules.

Authority and boundary rules.

Monitoring signal catalogue.

ALLOW / HOLD / ESCALATE / REFUSE decision tables.

Log and decision trail requirements.

Incident triage form.

Evidence preservation checklist.

Remediation backlog.

Phase 3 Success Test

Can the organization identify what evidence proves control for each priority AI workflow?

If not, controls may exist as intention but not as evidence.

E.6 Phase 4 — Days 61–90: Test, Remediate and Report

The fourth phase tests whether the evidence architecture works.

This is where the organization moves from design to assurance.

Objective

Test priority workflows, identify gaps, remediate critical issues and report readiness to leadership.

Key Actions

1. Perform Case-Level Sampling

Select case samples for each priority system.

Include:

ordinary cases;

high-consequence cases;

cases with human review;

cases with override;

cases with escalation;

cases with missing data warnings;

cases involving downstream effects;

recent cases after any workflow change.

For each sample, ask:

- what system was used;
- what data was used;
- was data fresh;
- what output was produced;
- who reviewed it;
- did review occur before consequence;
- what authority applied;
- were monitoring alerts active;
- what downstream consequence occurred;
- can the decision trail be reconstructed?

2. Rate Evidence Quality

Use a simple scale:

- 0 — no evidence.
- 1 — assertion.
- 2 — documented process.
- 3 — operational record.
- 4 — reconstructable decision trail.
- 5 — runtime-assured control.

For high-consequence workflows, aim for Level 4 in core domains:

- input data;
- human oversight;
- authority;
- logs;
- downstream consequence.

3. Identify Systems in HOLD, ESCALATE or REFUSE

Not every system will be ready by Day 90.

That is acceptable if the state is visible.

The organization should identify:

- systems allowed to continue;
- systems allowed only with conditions;
- systems held pending remediation;

systems escalated for governance decision;
systems refused or suspended under current conditions.

4. Remediate Critical Gaps

Critical gaps may include:

no owner;
no provider instructions;
use outside intended purpose;
no meaningful human oversight;
stale data used in high-consequence workflow;
no authority map;
no decision trail;
downstream automation uncontrolled;
no incident route.

Remediation should have owners and due dates.

5. Run One Incident Response Drill

Test a practical scenario.

Recommended scenarios:

stale data in high-risk workflow;
model fallback event;
human reviewer unavailable;
downstream automation discovered;
affected-person complaint;
missing logs.

Record:

time to detect;
time to escalate;
evidence preserved;
decision made;
gaps identified;
remediation actions.

6. Prepare Board or Executive Summary

Prepare a short leadership report.

Include:

- systems reviewed;
- high-risk / high-consequence systems;
- readiness status;
- key risks;
- systems in HOLD / ESCALATE / REFUSE;
- incidents or near misses;
- evidence quality;
- remediation plan;
- decisions required.

7. Create 90-Day Remediation Roadmap

The first 90 days will identify issues that cannot all be fixed immediately.

Create a post-90-day roadmap for:

- log improvements;
- monitoring automation;
- human oversight training;
- provider documentation gaps;
- authority mapping;
- downstream integration controls;
- context management;
- incident response maturity;
- audit readiness.

Phase 4 Deliverables

By Day 90, produce:

- Case sampling results.
- Evidence quality ratings.
- Updated readiness matrix.
- List of systems by governance state.
- Critical remediation plan.
- Incident response drill record.
- Board / executive summary.
- Post-90-day remediation roadmap.
- Updated evidence pack indexes.
- Final 90-day implementation report.

Phase 4 Success Test

Can the organization show which priority AI workflows are controlled, which are partially controlled, which are unknown and which should not continue under current conditions?

If yes, the 90-day programme has succeeded.

E.7 90-Day Timeline Summary

Time Period	Focus	Main Deliverables
Days 1–15	Mobilize and identify	Governance owner, working group, initial AI inventory, priority list
Days 16–30	Map and classify	Intended purpose map, actual use map, workflow map, consequence points, input data map
Days 31–60	Build control evidence	Evidence packs, oversight model, input rules, monitoring signals, decision-state logic
Days 61–90	Test and report	Case sampling, evidence ratings, remediation plan, incident drill, board summary

E.8 Priority System Triage Model

Not every AI system should receive the same attention in the first 90 days.

Use the following triage model.

Priority 1 — Immediate Review

Systems that:

are high-risk;

affect natural persons;

influence employment, healthcare, finance, public services or education;

trigger downstream actions;

- are agentic or tool-connected;
- affect safety or fundamental rights;
- lack clear ownership;
- have known incidents or complaints.

Priority 2 — Review Within 90 Days

Systems that:

- support internal decisions;
- generate recommendations;
- process sensitive data;
- influence procurement, compliance or operational risk;
- are in pilot but may move to production;
- rely on third-party provider models;
- use LLM context or retrieval.

Priority 3 — Review After 90 Days

Systems that:

- are low-consequence;
- support internal drafting only;
- do not affect natural persons;
- do not trigger workflows;
- do not use sensitive data;
- have clear human review and no downstream consequence.

Priority 3 systems should still be registered.

They simply may not require the deepest review in the first 90 days.

E.9 Minimum Day-30 Checklist

By Day 30, the organization should be able to answer:

- Which AI systems are in use?
- Which systems are high-risk or high-consequence?
- Who owns each priority system?
- What provider instructions exist?
- What is each system's intended purpose?
- What is the actual use?

Which workflows are affected?
 What outputs are produced?
 Where does consequence attach?
 Who performs human oversight?
 What input data is used?
 What downstream systems receive AI output?
 Which systems have unclear scope?
 Which systems require immediate HOLD or ESCALATE?
 What are the biggest evidence gaps?

If these questions cannot be answered, the organization should not report strong Article 26 readiness.

E.10 Minimum Day-60 Checklist

By Day 60, the organization should be able to show:

Evidence pack index for each priority system.
 Instructions-to-controls mapping.
 Human oversight role and authority.
 Training gap list.
 Input data source map.
 Data freshness rules.
 Authority and approval map.
 Downstream effect map.
 Monitoring signal catalogue.
 Log inventory.
 Decision trail requirements.
 ALLOW / HOLD / ESCALATE / REFUSE rules.
 Incident triage form.
 Evidence preservation checklist.
 Remediation backlog.

If these items exist only as drafts, the organization should mark readiness as partial.

E.11 Minimum Day-90 Checklist

By Day 90, the organization should be able to show:

Case samples tested.
Decision trails assessed.
Evidence quality ratings completed.
Systems grouped by ALLOW / HOLD / ESCALATE / REFUSE.
Critical gaps identified.
Owners assigned to remediation.
One incident response drill completed.
Provider contact routes documented.
Board summary prepared.
Post-90-day roadmap created.

The Day 90 report should not hide gaps.
It should make gaps visible and actionable.

E.12 Common 90-Day Failure Patterns

Organizations should watch for common implementation failures.

1. Inventory Without Workflow Mapping

The organization lists AI systems but does not map where outputs influence consequence.

Result:

The register exists, but governance is shallow.

2. Provider Instructions Stored But Not Operationalized

Instructions are saved in a folder but not translated into procedures, training, monitoring or controls.

Result:

Compliance documentation exists, but Article 26 use control is weak.

3. Human Oversight Named But Not Tested

A reviewer role exists, but reviewers lack training, authority, evidence access or time.

Result:

Human-in-the-loop exists, but human-in-control does not.

4. Monitoring Focuses Only on Uptime

Dashboards show technical status, but not data freshness, actual use, oversight, authority or downstream effects.

Result:

The system can be green while governance is failing.

5. Logs Exist But Decision Trails Fail

Logs show activity, but cannot connect input, output, human review, authority and consequence.

Result:

The organization cannot reconstruct control.

6. No Failure States

Every workflow defaults to continuation because HOLD, ESCALATE and REFUSE states are not defined.

Result:

Silent continuation risk remains high.

7. Board Reporting Is Too Positive

Leadership receives adoption metrics but not control gaps.

Result:

Executive oversight is weak.

8. No Owner for Remediation

Gaps are identified but not assigned.

Result:

Findings remain open and operational risk persists.

E.13 Post-90-Day Maturity Path

The first 90 days should create the foundation.

After Day 90, the organization should mature the programme.

Months 4–6

Focus:

- automate monitoring signals;
- improve decision trail logging;
- complete training;
- remediate critical gaps;
- test more cases;
- strengthen provider contracts;
- implement fallback authority rules.

Months 7–12

Focus:

- expand to medium-risk systems;
- integrate AI governance into procurement;
- improve board reporting;
- run incident drills;
- create continuous assurance cycle;
- improve affected-person information processes;
- monitor actual use drift.

Year 2

Focus:

- runtime governance automation;
- centralized AI evidence platform;
- cross-system monitoring;
- advanced context governance;
- agentic AI transition controls;
- internal audit programme;
- sector-specific assurance;
- evidence-based trust reporting.

The roadmap should evolve from manual evidence collection toward integrated runtime governance.

Manual controls may be acceptable at the start.

High-consequence systems eventually need stronger automation and assurance.

E.14 90-Day Board Report Template

The Day 90 board report should be concise.

Template E.14 — 90-Day Article 26 Readiness Report

Reporting Date:

Insert date.

Prepared By:

Insert owner.

Overall Status:

Green / Amber / Red / Mixed.

Executive Summary

Briefly describe:

systems reviewed;

overall readiness;

strongest controls;

weakest controls;

decisions required.

Systems Reviewed

System	Use Case	Status	Key Issue
		ALLOW / HOLD / ESCALATE / REFUSE	

Key Findings

Insert finding.

Insert finding.

Insert finding.

Evidence Quality

Domain	Status	Notes
System Register	Green / Amber / Red	
Intended Purpose	Green / Amber / Red	

Domain	Status	Notes
Human Oversight	Green / Amber / Red	
Input Data	Green / Amber / Red	
Monitoring	Green / Amber / Red	
Logs / Decision Trails	Green / Amber / Red	
Escalation	Green / Amber / Red	

Systems Requiring Decision

System	Issue	Recommended Decision
		Continue / continue with conditions / hold / escalate / refuse

Remediation Plan

Action	Owner	Due Date	Priority
			High / Medium / Low

Board Decisions Requested

Insert decision.

Insert decision.

Insert decision.

Next Review

Insert date or trigger.

E.15 Appendix E Summary

The 90-Day Implementation Roadmap provides a practical path for organizations beginning Article 26 deployer governance.

The goal of the first 90 days is not perfect maturity.

The goal is visibility, ownership, workflow mapping, evidence structure, runtime decision logic and leadership awareness.

The roadmap is divided into four phases:

Days 1–15: Mobilize and identify.

Days 16–30: Map and classify.

Days 31–60: Build control evidence.

Days 61–90: Test, remediate and report.

By Day 90, an organization should know:

which AI systems are high-risk or high-consequence;

what they are used for;

where AI output influences consequence;

who controls them;

what evidence exists;

what evidence is missing;

which systems may continue;

which systems require HOLD, ESCALATE or REFUSE;

what remediation is needed.

The central principle is:

Do not wait for perfect governance before creating control visibility.

Start with the highest-consequence systems.

Map actual use.

Build evidence packs.

Test decision trails.

Define ALLOW, HOLD, ESCALATE and REFUSE.

Report honestly to leadership.

That is how deployers move from AI governance intention to operational evidence of control.

Appendix F

Final Checklist and Document Control Page

Legal anchor and framework boundary. [Article 26](#). The checklist operationalises the study's interpretation and does not replace legal, conformity or sector-specific assessment.

F.1 Purpose of This Appendix

This appendix provides a final readiness checklist and document control structure for the study.

Its purpose is to help the reader, reviewer, deployer, governance team, internal audit function or board-level audience assess whether the document is complete, usable and ready for formal circulation.

This appendix serves four functions:

- It provides a final Article 26 deployer governance checklist.
- It provides a document control page for versioning and ownership.
- It provides a publication readiness checklist.
- It provides a final implementation review page.

The study argues that Article 26 deployer governance should be understood as evidence-based runtime control.

This appendix helps confirm whether the organization has translated that idea into practical readiness.

F.2 Final Article 26 Deployer Governance Checklist

This checklist should be used before deploying, expanding or materially changing the use of a high-risk or high-consequence AI system.

Control Area	Question	Status	Evidence Location
System Identification	Is the AI system recorded in the internal AI system register?	Complete / Partial / Missing	
Ownership	Are business, technical and governance	Complete / Partial / Missing	

Control Area	Question	Status	Evidence Location
	owners assigned?		
Provider	Is the provider identified and documented?	Complete / Partial / Missing	
Instructions for Use	Are provider instructions available and current?	Complete / Partial / Missing	
Instructions Mapping	Have provider instructions been translated into internal controls?	Complete / Partial / Missing	
Intended Purpose	Is provider-defined intended purpose documented?	Complete / Partial / Missing	
Approved Use	Is internal approved use documented?	Complete / Partial / Missing	
Actual Use	Is actual use mapped and monitored?	Complete / Partial / Missing	
Scope Drift	Are use expansion and scope drift monitored?	Complete / Partial / Missing	
Risk Classification	Has high-risk or high-consequence status been assessed?	Complete / Partial / Missing	
Workflow Map	Is the affected workflow mapped?	Complete / Partial / Missing	
Consequence Point	Is the point where AI output influences consequence identified?	Complete / Partial / Missing	
Output Role	Is the AI output role defined?	Complete / Partial / Missing	

Control Area	Question	Status	Evidence Location
Output Authority	Is the output authority level defined?	Complete / Partial / Missing	
Input Data	Are input data sources mapped?	Complete / Partial / Missing	
Data Freshness	Are freshness thresholds defined?	Complete / Partial / Missing	
Data Completeness	Are missing-data rules defined?	Complete / Partial / Missing	
Representativeness	Is representativeness assessed where relevant?	Complete / Partial / Missing	
Context	Is context inspectable for LLM or agentic workflows?	Complete / Partial / Missing	
Human Oversight	Is human oversight assigned?	Complete / Partial / Missing	
Competence	Is reviewer competence evidenced?	Complete / Partial / Missing	
Training	Is system-specific reviewer training completed?	Complete / Partial / Missing	
Authority	Does the reviewer have authority to intervene?	Complete / Partial / Missing	
Support	Does the reviewer have evidence, time and escalation support?	Complete / Partial / Missing	
Review Timing	Does review occur before consequence attaches?	Complete / Partial / Missing	

Control Area	Question	Status	Evidence Location
Override	Can the reviewer override or modify the AI output?	Complete / Partial / Missing	
HOLD	Can the workflow pause when evidence is missing or stale?	Complete / Partial / Missing	
ESCALATE	Is escalation available for unresolved or higher-risk issues?	Complete / Partial / Missing	
REFUSE	Can the workflow refuse unsupported or unauthorized use?	Complete / Partial / Missing	
Monitoring	Are governance signals monitored, not only technical health?	Complete / Partial / Missing	
Alerts	Are alert thresholds and owners defined?	Complete / Partial / Missing	
Logs	Are relevant logs retained?	Complete / Partial / Missing	
Decision Trail	Can the workflow be reconstructed from input to consequence?	Complete / Partial / Missing	
Incident Response	Is AI incident triage defined?	Complete / Partial / Missing	
Evidence Preservation	Is evidence preservation ready for incidents and complaints?	Complete / Partial / Missing	
Provider Contact	Is the provider or distributor contact route documented?	Complete / Partial / Missing	
Suspension	Is suspension and	Complete / Partial /	

Control Area	Question	Status	Evidence Location
	return-to-service procedure defined?	Missing	
Workplace Information	Are workers and representatives informed where required?	Complete / Partial / Missing / N/A	
Affected Persons	Are affected natural persons informed where required?	Complete / Partial / Missing / N/A	
FRIA Link	Has fundamental rights impact assessment applicability been assessed where relevant?	Complete / Partial / Missing / N/A	
Public Authority Duties	Are public authority or EU database duties assessed where relevant?	Complete / Partial / Missing / N/A	
Reassessment	Are reassessment triggers defined?	Complete / Partial / Missing	
Board Reporting	Is leadership informed of control state and gaps?	Complete / Partial / Missing	
Internal Audit	Has internal review or case sampling been performed?	Complete / Partial / Missing	

F.3 Minimum Go / No-Go Checklist

Before a high-risk or high-consequence AI system moves into production or expanded use, the deployer should answer the following.

Go Conditions

The system may proceed where:

- The system is identified and owned.
- Provider instructions are available and operationalized.
- Intended purpose and actual use are aligned.
- Input data is suitable for the intended purpose.
- Human oversight is meaningful before consequence.
- Authority is valid.
- Monitoring covers governance signals.
- Logs support reconstruction.
- Escalation and incident response are ready.
- Information duties are addressed where required.
- Decision trails can be sampled.
- Reassessment triggers are defined.

Conditional Go

The system may proceed with restrictions where:

- The use is limited to pilot or controlled deployment.
- Human review is mandatory before consequence.
- Output is restricted to draft-only or advisory use.
- Downstream automation is disabled.
- Additional monitoring is active.
- Evidence gaps have named remediation owners.
- Leadership has approved the residual risk.
- A review date is scheduled.

No-Go Conditions

The system should not proceed where:

- Actual use is outside intended purpose.
- Required human oversight is not meaningful.
- Input data is stale or unsuitable for consequence.
- Authority is absent or unclear.
- Decision trails cannot be reconstructed.
- Monitoring cannot detect relevant control failure.
- Serious risk remains unresolved.
- Required information duties are unmet.

- Provider instructions are unavailable for high-risk use.
 - Fallback or model substitution is ungoverned.
 - Downstream consequence cannot be controlled.
 - The organization cannot explain why continuation is allowed.
-

F.4 Evidence Quality Final Check

Each priority AI system should be assessed using the following evidence quality scale.

Evidence Level	Meaning	Suitable for High-Consequence Use?
Level 0	No evidence exists.	No
Level 1	Control is asserted but not evidenced.	No
Level 2	Policy or procedure exists.	Not sufficient alone
Level 3	Operational records show the control operated.	Sometimes
Level 4	Decision trail is reconstructable.	Yes, where proportionate
Level 5	Control is monitored at runtime and connected to ALLOW / HOLD / ESCALATE / REFUSE.	Strongest

Final Evidence Question

For each priority system, ask:

Can we reconstruct a real case from AI input to AI output, human review, authority, monitoring, escalation where relevant, downstream consequence and final decision?

If the answer is no, the evidence pack is incomplete.

F.5 Final Document Control Page

Document Title

Article 26 as Runtime Governance: Evidence of Control, Human Oversight and Deployer Responsibility under the EU AI Act

Working Subtitle

From Compliance Documentation to Evidence-Based AI Control

Document Type

Professional study / governance paper / operational framework.

Version

V1.0 - Final Public Release.

Author

Jozsef Fodor.

Project / Brand

NextOne / NextOne Observatory.

Primary Subject

EU AI Act Article 26 deployer obligations for high-risk AI systems.

Secondary Subjects

Runtime governance, human oversight, evidence of control, AI monitoring, decision trails, agentic AI, fallback governance, context governance, hybrid AI, board oversight and internal audit.

Intended Audience

- AI deployers;
- compliance teams;
- risk teams;
- legal teams;
- internal audit;
- board members;
- technology leaders;
- AI governance professionals;
- regulated-sector operators;
- procurement, HR, healthcare, finance and public-sector governance teams.

Document Status

Final Public Release, approved by the author on 16 July 2026.

Confidentiality Status

Public.

Last Updated

16 July 2026.

Next Review

After any material legal, regulatory or guidance change.

Document Owner

Jozsef Fodor / NextOne Observatory.

Reviewers

Structured internal release audit and official-source verification; final author approval recorded on 16 July 2026.

Approval Status

Approved by the author for final public release on 16 July 2026.

Law and Guidance Reviewed As At

16 July 2026.

Legal Source Basis

Official text of Regulation (EU) 2024/1689, with current Commission implementation materials treated according to their legal status.

F.6 Version History

Version	Date	Author / Editor	Description of Change
V0.1	17 June 2026	Jozsef Fodor	Full draft assembly completed: 34 chapters plus Appendices A-F.
V0.9	20 June 2026	Jozsef Fodor	Public review draft released with canonical page and

Version	Date	Author / Editor	Description of Change
			preserved publication evidence.
V1.0	16 July 2026	Jozsef Fodor	Final Public Release approved; final PDF, legal-source status, release QA and immutable publication evidence completed.

F.7 Review Checklist Before Publication

Before publication, complete the following.

Structure Review

Item	Status
Title page complete	Complete
Abstract complete	Complete
Executive summary complete	Complete
Table of contents updated	Complete
Chapter numbering verified	Complete
Appendix numbering verified	Complete
Earlier “30 chapters” reference corrected	Complete
Final chapter count confirmed as 34 chapters plus Appendices A–F	Complete
Part headings consistent	Complete
Cross-references checked	Reviewed; final PDF page check at release

Legal and Source Review

Item	Status
EU AI Act references checked	Complete against official source record
Article 26 references checked	Complete
Article 13, 14, 27 and related references checked where used	Complete at article-anchor level
Legal claims reviewed carefully	Targeted source-control review complete; not a legal opinion
“Not legal advice” disclaimer included	Complete
Source list prepared	Complete
Quotations checked	No material long-form quotations used
Citations formatted consistently	Complete using chapter-level legal anchors

Content Review

Item	Status
Main thesis consistent throughout	Complete
ALLOW / HOLD / ESCALATE / REFUSE definitions consistent	Complete
Human-in-control concept consistent	Complete
Evidence of control concept consistent	Complete
Runtime governance concept consistent	Complete
Agentic AI sections aligned	Complete
Context governance sections aligned	Complete
Board and audit sections aligned	Complete
Appendices aligned with main chapters	Complete

Style Review

Item	Status
Tone professional and clear	Complete

Item	Status
Repetition reduced where appropriate	Preserved except release-control corrections
Definitions consistent	Complete
Sector examples balanced	Complete
Tables formatted consistently	Complete in source; final PDF QA at release
Long sections broken clearly	Complete
Final conclusion strong and clear	Complete

Publication Review

Item	Status
Document formatted as PDF	Complete - final V1.0 PDF generated
Page numbers added	Complete - verified in final V1.0 PDF
Header and footer added	Complete - verified in final V1.0 PDF
Cover page designed	Complete - final V1.0 cover
Version and document control included	Complete
Author bio added if desired	Not included
Contact information added if desired	Canonical page included
Distribution status confirmed	Final Public Release - author approved 16 July 2026

F.8 Final Implementation Checklist for Deployers

This checklist can be used after reading the study.

A deployer should ask:

System and Use

- Do we know all AI systems in use?
- Do we know which are high-risk or high-consequence?
- Do we know what each system is actually used for?
- Do we know where AI output influences consequence?

Controls

- Do we have meaningful human oversight?
- Do humans have competence, training, authority and support?
- Do we control input data relevance, freshness and completeness?
- Do we monitor actual use, data, oversight, authority and downstream effect?
- Do we have escalation and refusal paths?

Evidence

- Can we reconstruct a case?
- Do logs connect input, output, review and consequence?
- Do we preserve context for LLM or agentic systems?
- Do we record authority state?
- Do we preserve evidence during incidents?

Runtime Governance

Can workflows enter HOLD?

Can unresolved issues ESCALATE?

Can unsupported use REFUSE?

Do we know which systems are currently in each state?

Does the board see control status, not only AI adoption?

Reassessment

Do provider updates trigger review?

Do model changes trigger review?

Does actual use drift trigger review?

Do incidents and complaints trigger review?

Does downstream integration trigger review?

The final deployer question is:

Can we prove that AI-supported workflows remain under valid control conditions when they matter?

F.9 Final Reader Checklist

After reading this study, the reader should be able to explain:

Why Article 26 is not only a compliance checklist.

Why deployer governance depends on actual use.

Why human-in-the-loop is weaker than human-in-control.

Why input data and context are control surfaces.

Why monitoring must detect governance conditions.

Why logs must become decision trails.

Why fallback is not neutral.

Why agentic AI requires transition-level control.

Why evidence architecture is necessary.

Why ALLOW, HOLD, ESCALATE and REFUSE make governance operational.

Why boards need control metrics, not only AI adoption metrics.

Why trustworthy AI depends on evidence-based runtime control.

F.10 Final Statement

This study began with a simple concern:

AI governance often fails not because organizations lack principles, but because principles are not translated into operational control.

Article 26 gives deployers a practical challenge.

Article 26 requires deployers to use high-risk AI systems in accordance with instructions, assign competent and supported human oversight, monitor operation, keep controlled logs, and act on identified risks and serious incidents. The additional runtime-control architecture developed in this study is an operational interpretation.

The deeper message is that responsible AI use must be provable.

A deployer should be able to show not only that an AI system was deployed, but that it remained under valid control when its output began to matter.

The final operational sentence of this study is:

A deployer should be able to prove that a high-risk AI system was used under valid, monitored, human-controllable and reconstructable conditions when its output influenced consequence.

That is the foundation of evidence-based runtime AI governance.

Appendix F Summary

This appendix provides the final checklist and document control structure for the study.

It includes:

- final Article 26 deployer governance checklist;
- go / no-go checklist;
- evidence quality check;
- document control page;
- version history;
- publication readiness checklist;
- deployer implementation checklist;
- reader checklist;
- final statement.

With this appendix, the full draft structure is complete:

- 34 main chapters;
- Appendix A: Article 26 Clause-by-Clause Operational Mapping;
- Appendix B: ALLOW / HOLD / ESCALATE / REFUSE Decision Tables;
- Appendix C: Evidence Pack Templates and Sample Forms;
- Appendix D: Glossary of Key Terms;
- Appendix E: 90-Day Implementation Roadmap;
- Appendix F: Final Checklist and Document Control Page.

The editable master, release-control record and legal-source apparatus have been assembled for V1.0. Author approval was recorded on 16 July 2026. Final full-document visual QA, PDF generation and immutable evidence-package creation were completed for this public release.

Legal Sources, Guidance Status and Citation Record

V1.0

Purpose and interpretive boundary

This source record identifies the legal provisions and official implementation materials used to anchor the study. The official text of Regulation (EU) 2024/1689 is authoritative. Commission webpages, service-desk summaries, draft guidelines and policy announcements support navigation and context but do not replace the Regulation or any formally adopted amendment.

Law and guidance reviewed as at 16 July 2026. The study is an operational governance interpretation and does not provide legal advice or guarantee compliance.

Primary authoritative legal source

[Regulation \(EU\) 2024/1689 - Official Journal / EUR-Lex](#) - Regulation of 13 June 2024 laying down harmonised rules on artificial intelligence. This is the primary authority for all formal legal propositions in the study.

Core provisions used

[Article 3 - Definitions](#) - deployer, intended purpose, instructions for use, input data and serious incident definitions.

[Article 6 - Classification rules for high-risk AI systems](#) - high-risk classification and Annex III treatment.

[Article 12 - Record-keeping](#) - automatic event logging and traceability requirements for high-risk AI systems.

[Article 13 - Transparency and provision of information to deployers](#) - instructions for use, capabilities, limitations, human oversight measures and logging information.

[Article 14 - Human oversight](#) - effective natural-person oversight, automation-bias awareness, interpretation, override and interruption.

[Article 26 - Obligations of deployers of high-risk AI systems](#) - central legal basis for use according to instructions, human oversight, input data, monitoring, logs, notification, registration and cooperation.

[Article 27 - Fundamental rights impact assessment for high-risk AI systems](#) - FRIA duties where applicable.

[Article 49 - Registration](#) - registration obligations relevant to providers and certain deployers.

[Article 71 - EU database for high-risk AI systems listed in Annex III](#) - database context and access.

[Article 72 - Post-market monitoring by providers](#) - provider monitoring context referred to by Article 26(5).

[Article 73 - Reporting of serious incidents](#) - incident reporting framework and timelines.

[Article 79 - Procedure for AI systems presenting a risk](#) - risk concept referenced by Article 26(5).

[Article 86 - Right to explanation of individual decision-making](#) - individual explanation context where applicable.

[Article 113 - Entry into force and application](#) - application dates in the enacted Regulation, subject to any later formally adopted amendment.

Official Commission implementation materials

[AI Act regulatory framework](#) - official Commission overview of the AI Act and its risk-based framework.

[Guidelines for providers and deployers of AI high-risk systems](#) - as at 16 July 2026, the classification guidelines are draft and non-binding; the targeted consultation closes on 23 July 2026.

[Draft Commission guidelines on classification of high-risk AI systems](#) - draft guidance published 19 May 2026; used only as contextual classification material.

[Council final-adoption record for the Digital Omnibus on AI - final Council approval was given on 29 June 2026. The final act was signed on 8 July 2026; operative legal propositions in this study remain tied to formally published legal text and verified entry-into-force status.](#)

[Supporting implementation of the AI Act with clear guidelines](#) - Commission roadmap for further 2026 guidance, including deployer obligations, high-risk requirements, FRIA and incident reporting.

[Navigating the AI Act](#) - official implementation FAQ; useful context but not a substitute for the Regulation.

Source-status rules applied in this study

Binding legal text: described as a requirement or obligation only where anchored to the Regulation or another applicable legal instrument.

Official guidance: identified as guidance and treated according to whether it is final, draft, consultation-stage or non-binding.

Legislative status: proposals and political agreements are treated as developments; adopted acts are distinguished from formally published and applicable law.

NextOne Observatory framework concepts: ALLOW, HOLD, ESCALATE, REFUSE, evidence of control, continued validity, authority continuity, human-in-control and runtime governance are operational constructs unless the text expressly identifies a statutory source.

Sector examples and templates: illustrative and context-dependent; they may require additional sector, data-protection, employment, safety, procurement or professional review.

Citation method used in V1.0

The final public release uses chapter-level legal-anchor paragraphs and this consolidated source record. Legal anchors identify the relevant articles and distinguish statutory propositions from operational interpretation. A later academic edition may add conventional footnotes or endnotes without changing the substantive framework.

Verification record

Control	V1.0 status	Verification basis
Official Regulation identified	Complete	EUR-Lex Regulation (EU) 2024/1689
Articles 3, 6, 12, 13, 14 and 26 checked	Complete	Official-version text presented through the Commission AI Act Service Desk with link to EUR-Lex
Articles 27, 49, 71, 72, 73, 79, 86 and 113 anchored	Complete at article-reference level	Official AI Act Explorer article records
Draft high-risk guidance labelled correctly	Complete	Commission page: draft, non-binding, consultation until 23 July 2026

Digital Omnibus status bounded	Complete	Parliament approval 16 June, Council final approval 29 June and final-act signature 8 July 2026; operative claims remain tied to formally published text
Framework concepts distinguished from law	Complete	Front-matter disclaimer, chapter legal anchors and this source record
Professional legal opinion	Not performed	This document remains research and operational governance analysis, not legal advice

Suggested citation

Fodor, Jozsef. Article 26 as Runtime Governance: Evidence of Control, Human Oversight and Deployer Responsibility under the EU AI Act. NextOne Observatory, V1.0, 2026.

Canonical page

<https://nextone.ie/observatory/article26/>

Final source note

Readers should verify the current official legal text, any formally adopted amendments, implementing or delegated acts, harmonised standards and final Commission guidance before using this study for a compliance decision.