VIANORDIS POSITION PAPER 01 · SUPERSEDED

The AI Control Gap

Why model catalogues, policies and API gateways stop short once AI acts inside the enterprise

Architecture Profile Edition

SUPERSEDED — ARCHIVE COPY
This edition is retained for reference only. It was replaced on 26 August 2026 by a three-document set: **POSITION PAPER 01 V0.5.4**, the **GAIP CANDIDATE SPECIFICATION 0.4** and the **ADOPTION & MEASUREMENT GUIDE 0.1.4**. Several statements here were later corrected — among them the Cyber Resilience Act vulnerability-handling reference, the attribution of the PDP/PEP vocabulary, the DORA concentration-risk article, and the GAIP-0.1 conformance levels, which have since been withdrawn. Do not cite this edition as current.

Version	Draft v0.4
Status	Superseded archive edition
Date	26 August 2026
Author	Yves Frédéric N'Soussoula, Founder & CEO
Publisher	GoSec Cloud OÜ, Tallinn, Estonia
Licence	Creative Commons Attribution 4.0 International
Web	www.vianordis.eu

Open Review. This document is published for criticism. Corrections of fact are applied in the next edition and listed; substantive critique is credited unless the contributor prefers otherwise. Contact: www.vianordis.eu/en/contact · Law as at 26 August 2026. Not legal advice.

Contents

> **Publisher's disclosure.** This paper is published by GoSec Cloud OÜ (Tallinn, Estonia), which develops and operates **Vianordis**, an implementation of the architecture class described here. We have a commercial interest in the argument. We have tried to write a paper that survives being read by someone who does not. Chapter 3 states what this architecture does *not* solve, Chapter 5 states what already exists and does part of the job, and Annex C states exactly what is built and what is not. If you find those sections insufficient, tell us — the contact route is at the end.
>
> **Legal status.** Law as at **14 August 2026**. This is a strategic and technical position paper, not legal advice. Deadlines, interpretations and secondary sources change; the linked official texts govern. Nothing here replaces legal classification of a specific AI system, a data protection impact assessment, a conformity assessment, an information security review, or sector-specific analysis.

On one page

What changed. AI no longer only produces text. It reads mailboxes, joins records across systems, prepares transactions, operates applications and triggers actions. The output of a language model has become a business event.

Where the gap is. Many organisations now control *which model* may be used. This paper's central hypothesis is that materially fewer can demonstrate, for a specific action, *who authorised it, on whose behalf, under which mandate, under which policy version, from which sources its decisive parameters were derived, and whether the thing approved under a given system state is the thing that executed.* Access is not authority, and approval is not a guarantee that the underlying facts were true or still current.

Why this year. Three obligations are already live and one enters its first reporting phase next month:

- **§ 38(2) BSIG (Germany, in force since 6 December 2025)** — management bodies of in-scope essential and important entities have implementation and supervision duties; a culpable breach can create personal liability toward their own organisation under the company-law rules applicable to the entity.
- **DORA (since 17 January 2025)** — the management body defines, approves, oversees and is accountable for ICT risk arrangements, and the financial entity remains responsible when ICT services are outsourced.
- **Cyber Resilience Act — Article 14 from 11 September 2026** — manufacturers of in-scope products with digital elements must report actively exploited vulnerabilities on a 24 h / 72 h / final-report-within-14-days-after-a-fix-is-available sequence, and severe incidents on a 24 h / 72 h / one-month sequence. This affects many suppliers of distributable, embedded and on-premises software, but not every pure SaaS provider.
- **EU AI Act** — broadly applicable since 2 August 2026; the core high-risk obligations now apply from **2 December 2027** for Article 6(2) / Annex III systems and **2 August 2028** for Article 6(1) / Annex I systems following the Digital Omnibus on AI. That is implementation time, not a reason to postpone architecture work.

What to do. Inventory AI *actions*, not only AI models. Pilot one bounded tier 3 or tier 4 process. Give it a distinct agent identity, a scoped mandate, a versioned policy decision, a governed action envelope, a bound approval, an execution receipt and a correlated evidence record. Then measure reconstruction time, rejection of stale approvals, bypass coverage and operational cost before extending the pattern.

Three questions for IT leadership.

1. For an action an AI agent took last week, can we reconstruct the human or organisational principal, purpose, policy version and source of every impact-determining parameter — from records, without asking the team that built it?
2. If a person approved a payment, is the approval bound not only to amount, payee and invoice, but also to the material system state under which it was approved — and is it invalidated when that state changes?
3. Is it technically impossible for a compromised business application to reach a tier 4 or tier 5 target outside the enforcement point, and has the safe-failure behaviour been tested?

What v0.4 adds. This edition advances from an architectural position to a candidate implementation profile. Annex G defines the **Governed Action Interchange Profile (GAIP-0.1)**: the minimum artefacts, invariants, processing states and verification rules needed to carry authority from intent to execution. Annex H defines a measurement protocol so that the paper's thesis can be tested rather than repeated.

What this paper does not claim. That regulation mandates any particular product. That a control plane makes an organisation compliant. That signatures prove a decision was correct. That an approval remains valid after the world changes. That any architecture defends against a fully compromised administrative domain or against an agent acting with valid authority on attacker-supplied facts. See Chapter 3.

Document status, audience and scope

Audience

This paper is written for people who will have to answer for an AI-initiated action after it happened:

- executive boards and managing directors;
- CIOs, CTOs, CISOs and CDOs;
- IT and enterprise architects;
- data protection, compliance and risk owners;
- owners of AI platforms and automation;
- works councils and employee representatives (see §3.5, which is written for them specifically);
- developers, maintainers and open-source contributors.

Core thesis

> **Enterprises do not fail to scale AI only because of model quality or regulation. They fail because organisational authority and technical execution are not connected.**

This thesis is falsifiable, and Chapter 10 states four measurements that would confirm, weaken or falsify material parts of it. We do not yet have those measurements at scale. Saying so is part of the argument.

Evidence status — read this before Chapter 1

This is a **position paper**, and its central claim is currently supported by architecture and reasoning, not by published field data. We are explicit about that for two reasons. First, an unmarked assertion dressed as a finding is exactly the kind of governance artefact this paper argues against. Second, the measurements in §10.5 and Annex H are the mechanisms we are asking the industry to help gather.

Where this paper cites law, it cites the official text. Where it describes Vianordis, it states implementation status. Market-wide statements are framed as hypotheses unless supported by published evidence. Readers who find the lack of field data unsatisfying are reading it correctly: the purpose of §10.5 and Annex H is to turn the claim into a measurable research programme rather than to disguise inference as fact.

How normative language is used in v0.4

The main body remains explanatory and non-normative. Annex G uses **MUST**, **MUST NOT**, **SHOULD**, **SHOULD NOT** and **MAY** to define a candidate technical profile. Those words describe proposed conformance requirements for GAIP-0.1. They are not statements of law, do not imply certification, and do not make GAIP an adopted standard. The profile is published for open review and is expected to change.

Executive summary

The enterprise conversation about AI is changing shape. The first phase was chatbots, text generation and personal assistants. The next is systems and agents that reach into data, prepare business transactions, operate applications and trigger actions.

The decisive question moves with it:

> **Not only: which AI may we use? But: who authorises the AI when it acts on our behalf?**

Many organisations now have AI policies, approved model catalogues, data protection rules, API gateways and technical logging. All of these are necessary. Individually, none of them closes the chain between a business intention and the action that actually executed.

A valid API key proves that a system was technically able to reach an application. It does not prove that the specific action was organisationally mandated, purpose-bound, proportionate, and — where required — approved by a competent human.

> **Access is not authority.**

From that separation comes the **AI control gap**. Organisations may be able to identify the model or technical account that triggered a call. The hypothesis tested by this paper is that, for a material share of actions, they cannot demonstrate from existing records:

- which natural or legal person stood behind it;
- on whose behalf the agent acted;
- what mandate it actually held;
- which data and systems were permissible for that purpose;
- which policy version applied at the moment of decision;
- whether human approval was required;
- whether the approved action is the action that executed;
- and what verifiable event chain remains.

This paper calls AI-assisted execution that cannot be reconstructed **Shadow Execution**. It can arise inside a fully approved environment, with an approved model, run by an approved team. The problem is not the model. It is the missing link between identity, mandate, permission, policy, approval, execution and evidence.

Shadow Execution is not a new phenomenon. It is the confused-deputy and delegation problem, known since the 1980s, and it applies equally to RPA bots, cron jobs and CI/CD pipelines. What is new is the scale: agents generate delegation chains at a frequency and branching depth that makes manual reconstruction impractical.

Regulation raises the cost of not solving it — though not in the way most summaries suggest. The AI Act's high-risk obligations were **deferred** by the Digital Omnibus on AI to December 2027 and August 2028. The obligations that bite *today* are elsewhere: personal management liability under German § 38(2) BSIG, management-body accountability under DORA, and the Cyber Resilience Act's 24-hour vulnerability reporting clock from 11 September 2026. Chapter 6 sets out all four, with dates.

Governed Intelligence is the operating state in which an AI-assisted action does not execute merely because it is technically reachable, but passes an enforceable, reviewable and revocable control chain:

> **Intent → Identity → Mandate → Context → Policy → Approval → Execution → Evidence**

Governance then stops being reconstructed only afterwards in reports. It becomes a precondition of the action.

Version 0.4 makes that proposition more concrete. The chain is carried in a **Governed Action Envelope** that binds the proposed action, parameter provenance, mandate, policy decision, approval, expiry and material execution preconditions. At commit time the enforcement point revalidates revocation and state. If the target state has changed, the approval is stale and the transaction must be re-evaluated or re-approved. The resulting execution receipt is joined with the decision record in the evidence layer.

This does not make distributed transactions perfectly atomic. It makes the residual gap explicit: an action can be approved correctly and still become invalid before execution. That time-of-check/time-of-use problem is addressed in §3.6 and §7.7.

Chapter 5 is the chapter most vendor papers omit: the relationship of this architecture to prior art. Policy decision points and policy enforcement points come from XACML and NIST SP 800-207. Delegation semantics exist in OAuth 2.0 Token Exchange (RFC 8693), macaroons and GNAP. Workload identity exists in SPIFFE/SPIRE. Tamper-evident logs come from Certificate Transparency. Supply-chain integrity comes from SLSA, in-toto and Sigstore. None of this is new, and pretending otherwise would be the fastest way to lose a technical reader. The contribution claimed here is narrower: a candidate composition profile that binds *purpose*, *risk limit and approval obligation* to a delegation, carries the provenance of every decision parameter, binds approval to material execution preconditions, and produces a single correlated decision-and-execution evidence record.

Chapter 3 states what none of this fixes — including the case that matters most: if a model derives action parameters from an attacker-controlled document, a correctly bound approval binds to falsified values. The chain is formally intact and materially worthless. Any architecture that does not say this out loud should not be trusted with the rest.

Vianordis is our implementation of this architecture class: a model-independent control layer for people, agents, applications and business processes. It is neither a language model nor a wrapper around one. As of this edition it is in **Controlled Beta**, by invitation, with no public source and no completed certification. Annex C lists all 25 applications, their status, and what has not been done yet.

Technical openness alone is not enough. A critical control architecture also needs institutional stewardship that protects code, brand, licence model and purpose from unilateral capture. The intended vehicle is an Estonian foundation (*sihtasutus*), the **OpenSovereign Alliance Foundation**. As of this edition it is **planned; no cutover has been completed**. Chapter 11 states what that means for anyone considering a contribution or a licence today.

---

1. When an answer becomes an action

1.1 Governance mostly stopped at model access

The first wave of enterprise AI governance was about tool selection:

- May this AI service be used at all?
- What data may be entered into a model?
- Which vendor satisfies our data protection requirements?
- Which models are centrally approved?
- May staff use personal accounts?
- Which outputs must be labelled?

These questions remain relevant. They are questions about **access to AI**.

Agents add a second layer. An agent can read mail and documents, join CRM, ERP and HR records, create tasks and appointments, prepare purchase orders, create accounts and permissions, change configurations, publish content, prepare payments and contract steps, and drive external systems through APIs.

At that point a linguistic output has become a business event.

> **A model answers a question. An agent changes a state.**

Architecturally, this is the whole difference.

1.2 A prompt is not a mandate

Consider: *"Check this invoice and pay it if everything is correct."*

The sentence expresses intent. It does not answer:

- Who is entitled to issue this instruction?
- Does that person represent only their own cost centre, or the whole company?
- May the agent only prepare a payment, or also release it?
- What amount threshold applies?
- Which suppliers may be processed?
- Is dual approval required?
- Which data may be transmitted to the selected model?
- What happens if the agent interprets the instruction differently?
- How is it ensured that the action executed after approval is the action approved?

Natural language expresses intent. It is not by itself sufficient technical authorisation.

1.3 What existing controls actually cover

The table below is deliberately written in the strongest reasonable form of each existing control. Earlier drafts of this paper understated them, which made the argument easier to write and easier to dismiss.

Existing control	What the mechanism actually covers today	What is typically still outside its model
Model catalogue	Which models and vendors are approved, under which data classes	What a specific action may do in a specific business transaction
IAM and SSO, including delegation flows such as OAuth 2.0 Token Exchange (RFC 8693), PIM/JIT elevation, workload identity federation and Continuous Access Evaluation	Authentication, delegated technical authorisation, actor claims, session revocation	Purpose binding, monetary and risk limits, and revocation of a single *assignment* rather than a session
Roles and groups	Which application and which coarse operations are reachable	Which purpose the entitlement was granted for, and which single action it justifies
API gateway	Scopes, claims, actor claims, rate limits, external PDP calls	Whether the resulting business act was organisationally mandated and proportionate
Data loss prevention	Which data classes may leave which boundary	Whether use of that data is lawful *for this purpose*
Prompt and output filters	Which inputs and outputs are unwanted	Which real-world action may result from an accepted output
Workflow approval (SAP, Coupa, Workday and comparable)	Approval bound to a document object inside that system	Approval bound to an action executed by an agent *across* systems, including parameter provenance
SIEM and logging	Which events were recorded, per system	Which policy version and which mandate justified the act, correlated across systems
GRC documentation, ISO/IEC 42001, ISO/IEC 27001	Which rules, controls and management processes were decided and audited	Whether those rules were technically enforced at the moment of execution

None of these controls is redundant. Several of them do more than this paper's earlier drafts credited. The gap is not that they are weak; it is that they were designed for different questions, sit in different systems, and do not compose into one authorisation and evidence chain.

1.4 An approval is not yet a transaction commitment

The action described to an approver is evaluated at a point in time. The target system executes later. Between those moments, the relevant state may change:

- a supplier may be blocked;
- a bank account may be replaced;
- a purchase order may be closed or amended;
- an entitlement may be revoked;
- a sanctions-screening result may change;
- a policy or risk threshold may be updated;
- another process may consume the budget or resource that the action assumed was available.

This is a classic **time-of-check/time-of-use (TOCTOU)** problem. Binding an approval only to the visible parameters is therefore insufficient. A governed action needs two distinct commitments:

1. an **action commitment** — a digest of the operation and the parameters the approver saw; and
2. a **precondition commitment** — a digest or version reference for the material state that made the operation permissible.

At execution time the action broker must verify both. Where the target system supports conditional writes, optimistic concurrency, version identifiers or compare-and-set semantics, the broker should use them. Where it does not, the adapter must re-read the material state immediately before execution and either re-evaluate the policy or reject the action as stale.

The distinction matters because a perfectly signed approval can authorise an action that is no longer permissible. **Cryptographic binding proves equivalence to what was approved; it does not prove that the approval is still fresh.**

---

2. From Shadow AI to Shadow Execution

2.1 Shadow AI describes only part of the risk

Shadow AI usually means the use of unapproved AI services: personal accounts, unknown browser extensions, unvetted models. It is partly addressable through model catalogues, network rules, contracts, training and central access.

The next class of risk arises **inside** the approved environment. An officially sanctioned agent can:

- operate with an over-privileged service account;
- inherit the requesting user's entitlements and keep them;
- chain tools whose combined effect was never assessed;
- carry an instruction beyond its original purpose;
- pass data to a different model or provider;
- apply an approval to an action other than the one presented;
- keep acting after a role change or a leaver event;
- produce logs across systems that cannot be correlated afterwards.

Shadow AI and Shadow Execution are **not mutually exclusive**, and one is not simply the successor of the other. Shadow AI is a possible cause of Shadow Execution, not a necessary one. Unapproved tools produce unreconstructable actions; so do approved ones.

2.2 Definition

The definition below is deliberately written so that it can be evaluated by an auditor **without reference to any particular product or vendor**:

> **Shadow Execution** is an AI-assisted action for which an auditor cannot, within a defined time budget and from the organisation's existing records, reconstruct (a) the human or organisational principal on whose behalf it was taken, (b) the purpose and scope it was permitted under, (c) which rule set the organisation can show was actually applied at the moment of decision, and (d) whether any approval on record can be tied, by any means, to the values that actually executed.

Shadow Execution does not imply that an action was unlawful or harmful. It means the organisation cannot establish its legitimacy or its course of events.

The risk is therefore not only a wrong model answer. It is that a *correct or plausible* answer becomes an insufficiently authorised action.

On novelty. We make no claim to have discovered a new class of risk. This is the delegation and confused-deputy problem described by Norm Hardy in 1988, and it applies to RPA bots, scheduled jobs and CI/CD pipelines exactly as it applies to agents. Substitute "script-driven action" for "AI-assisted action" in the definition above and it remains true. What agents change is scale and opacity: delegation chains multiply, branch, and are generated at machine speed by a component whose reasoning is not directly inspectable. Manual reconstruction stops being practical somewhere on that curve. That is a quantitative change with qualitative consequences, and it is the honest version of the claim.

2.3 Observable forms

Each of the following is stated as an **observable configuration**, not a documented incident. We are not aware of published, attributable case studies for all of them, and we say so rather than implying otherwise.

Over-privileged technical identities. An agent uses a service account whose permissions were sized for integration convenience rather than for the individual task.

Unbounded delegation. A user signs in once; the agent receives a long-lived token and continues to act later, in a changed context.

Unbound approvals. A human confirms "the payment" without the approval being technically bound to amount, payee, invoice reference and the specific transaction.

Tool chains without combined assessment. Every individual tool is approved. Mail access plus document analysis plus ERP write plus external communication produces an effect that was never assessed as a whole.

Retrospective reconstruction. The organisation holds several logs but cannot reliably join which user instruction produced which model decision, which policy evaluation and which system action.

Policy drift. An agent was approved under an earlier policy. The rule changed; running mandates and agent instances were never re-evaluated.

Trust inheritance through integrations. An approved business application calls an agent that calls further services. The original approval is passed implicitly across system boundaries, with no delegation step visible or bounded.

2.4 Impact tiers: not every AI use needs the same depth of control

This taxonomy is introduced here rather than late in the paper, because Chapter 3 and Chapters 7 to 10 depend on it.

Tier	Effect class of the action	Example	Typical control
0	Inform	Summarise a document	Data and model policy
1	Recommend	Produce a proposed course of action	Labelling, sources, review
2	Prepare	Create a draft email, contract or booking	Mandate, draft marking
3	Execute reversibly	Create a calendar entry, update an internal ticket	Policy, bounded mandate, logging
4	Execute consequentially	Payment above a defined threshold, publication, permission change	Bound human approval
5	Critical or irreversible	Production shutdown, clinical or personnel decision	Strict oversight, multi-party approval, or prohibition

The tier is an attribute of **the action**, not of the model. It is also value-sensitive: a payment below a defined threshold, on a full three-way match, may reasonably sit at tier 3, while the same operation above that threshold sits at tier 4. Where an organisation draws that line is a policy decision, and §8.3 shows one worked example of it. The same model can operate at tier 0 for a document summary and at tier 5 when changing production parameters.

This is an internal architecture and control instrument. It is **not** the legal risk classification of the EU AI Act, and it should not be presented to an auditor as such.

2.5 Two dimensions of risk: impact and provenance assurance

The impact tier in §2.4 describes what an action can do. It does not describe how trustworthy the facts behind the action are. A second axis is needed.

This edition introduces **provenance assurance classes** for impact-determining parameters:

Class	Meaning	Typical example
P0 — independently verified	The value was checked through a second, organisationally independent path against an authoritative source	Bank account confirmed through supplier onboarding and call-back verification

Class	Meaning	Typical example
P1 — authoritative system state	The value was read from a designated system of record with an authenticated version or state reference	Approved supplier master record, current employee directory record
P2 — authenticated assertion	The value was entered or asserted by an authenticated person or service, but not independently verified	Cost centre supplied by the requester
P3 — derived from untrusted content	A model or parser derived the value from mail, documents, web content, tool descriptions or other attacker-influenceable input	IBAN extracted from an inbound invoice
P4 — mixed, unknown or untraceable	The value has multiple unresolved sources, an unknown derivation path, or no reconstructable provenance	Parameter assembled by a chained agent with incomplete telemetry

These classes are not legal categories. They are an engineering control for deciding when automation must stop.

A conservative default matrix is:

Action tier	P0–P1	P2	P3	P4
0–1	Permit under data/model policy	Permit with attribution	Permit with attribution	Permit only if the information purpose tolerates uncertainty
2	Permit	Permit with review	Draft only; visually mark provenance	Hold for clarification
3	Permit with normal controls	Permit within bounded thresholds	Verify impact-determining fields before execution	Deny or reduce to tier 2
4	Permit with bound approval and fresh preconditions	Require explicit approval and, where appropriate, second control	Do not execute until critical fields reach P0 or P1	Deny
5	High-assurance process and multi-party control	Exceptional process only	Prohibit automated execution	Prohibit

The matrix is intentionally stricter than many existing workflows. It is a candidate default, not a universal rule. An organisation may choose differently, but the exception should be explicit, versioned and visible in the evidence record.

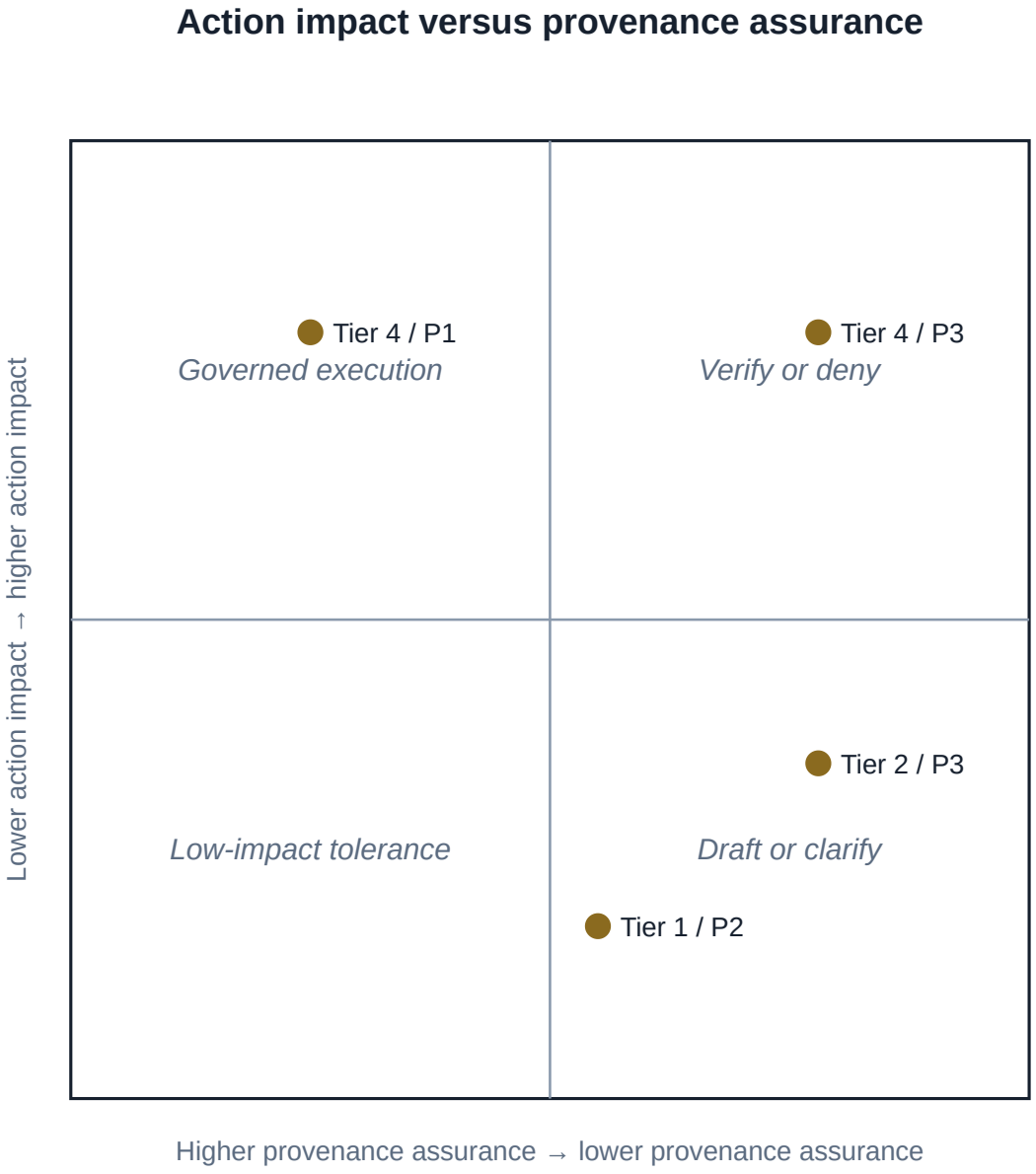

Figure 1. Impact and provenance are two axes, not one.

3. What this architecture does not solve

Most papers of this kind place their limitations at the end, where fewer people read them. They are here, before the architecture, because a reader who does not know the boundaries cannot evaluate the claim.

3.1 The general list

A control layer does not replace other governance, security and compliance work. In particular, it cannot automatically:

- determine the legal role of provider or deployer for a specific system;
- classify an AI system as high-risk or not high-risk in law;
- replace a data protection impact assessment;
- guarantee the quality or representativeness of training and input data;
- prevent discrimination or erroneous model outputs;
- replace subject-matter plausibility review;
- train staff;
- define organisational responsibility;
- replace an effective information security process;
- remove supply-chain dependency;
- replace a statutory conformity assessment or certification;
- protect a compromised organisation from the actions of its own highest-privileged administrators.

3.2 Input manipulation: the limit that matters most

This is the most important section of this paper.

A control chain authorises *an action with parameters*. It does not verify that those parameters describe reality.

Consider the worked example in Chapter 8. An agent reads an incoming invoice, derives supplier, bank details, invoice number and amount, and presents them to a human for approval. The approval is then cryptographically bound to exactly those values, and the action broker executes exactly that transaction.

Now assume the invoice document is attacker-controlled — which, for an inbound invoice mailbox, is the normal case rather than the exotic one. The document contains instructions or formatting designed to steer the extraction: a substituted IBAN, an injected directive, a manipulated line item. The model derives the falsified parameters. The human sees them rendered cleanly in an approval dialogue. The approval binds correctly. The broker executes exactly what was approved.

> **The control chain is formally intact and materially worthless.**

This is indirect prompt injection, and no amount of identity, mandate, policy or evidence architecture prevents it. Related failure modes with the same property: tool-description poisoning, where a malicious tool manifest steers selection; confused-deputy escalation through chained tools; and data exfiltration through a channel the agent is legitimately entitled to use.

What the architecture *can* contribute is narrower, and it should be stated narrowly:

Mitigation	What it actually achieves
Parameter provenance in the evidence record — every decision parameter carries its origin (model-derived from document X / user-entered / read from system of record)	Does not prevent manipulation; makes it reconstructable afterwards and reviewable at approval time

Mitigation	What it actually achieves
Out-of-band verification of critical fields — bank details checked against supplier master data, not against the document	Prevents the specific highest-value case, at the cost of a second data path
Provenance surfaced in the approval dialogue — the approver sees which values came from the untrusted document	Converts an invisible risk into a visible one; effectiveness depends on the approver actually reading it
Change thresholds on master data — any new or changed payee triggers a second, differently-sourced approval	Standard anti-fraud practice; predates AI and remains the strongest single control
Tier ceilings for untrusted input — actions whose parameters derive wholly from untrusted input are capped at tier 2 (prepare)	Reduces blast radius; reduces automation benefit

The honest summary: a runtime control plane reduces the number of ways an agent can act *without* authority. It does much less about an agent acting *with* authority on false premises. Claims that identity, policy and approval controls alone eliminate input-manipulation risk should therefore be treated with caution.

There is one genuine architectural consequence, and it is why this section exists rather than being a footnote: **evidence without parameter provenance is not evidence of much.** A record proving "person C approved amount X to payee Y" is close to useless in an injection incident. A record proving "amount X and payee Y were extracted by model M version N from document D, hash H, received from sender S, and were not verified against master data" is what an investigation needs. That requirement changes the evidence schema, and it is not in most designs.

3.3 The control plane is a single point of failure

A shared runtime control plane sits in the path of every governed business process. That is the point of it, and it is also its principal operational risk. If it is unavailable, either business stops or it does not — and either answer has a cost.

The design response is graceful degradation, not a claim of perfect availability:

- If a model becomes unavailable, another may be substituted where policy and data classification allow.
- If the **authorisation or policy component** fails, the system must not continue acting unconstrained. It should fall to a safe state: read-only access, draft mode, manual handling, or a full execution block for consequential actions.
- The tier taxonomy in §2.4 determines which actions block and which continue. That mapping must be decided in advance, written down, and tested — not discovered during an incident.

An organisation adopting this pattern should treat control-plane availability with the same seriousness it applies to its identity provider, because the failure characteristics are similar.

3.4 Cost, latency and brownfield reality

This paper describes a target architecture. It does not describe what it costs.

- **Latency.** Every governed action adds at least one policy evaluation and, for tiers 4 and 5, a human round trip. Policy evaluation can be single-digit milliseconds; human approval cannot. Processes with tight cycle times need their tier assignments chosen accordingly.
- **Implementation effort.** The adoption path in Chapter 9 has six phases and no stated durations, because honest durations depend on the estate. A bounded pilot may be achievable within weeks in some estates. Retrofitting hundreds of integrations is a different programme and should not be represented as a scaled version of the same task.
- **Brownfield.** Chapter 9 assumes a greenfield pilot. The harder question — how to route two hundred existing integrations through enforcement points without a two-year freeze — is not solved in this edition. The realistic answer involves prioritising by impact tier rather than by system, and accepting a long tail of tier 0–3 integrations that remain outside the enforcement path for some time. We would rather say that than imply a clean sweep.

The absence of field measurements remains a weakness. This edition therefore defines what must be measured without inventing values. For an automatically executed action, the machine-path latency budget is:

> **T_machine = T_identity + T_context + T_policy + T_state-check + T_broker + T_target + T_evidence**

For an approval-bound action:

> **T_end-to-end = T_machine + T_human + T_queue**

The second equation is usually dominated by human and queue time, not cryptography or policy evaluation. Annex H defines a reporting template for p50, p95 and p99 machine latency, approval latency, false-block rate, stale-approval rejection rate, exception rate and reconstruction cost. Until measured results are published, no performance claim should be inferred from this paper.

3.5 Employee representation, co-determination and the dual-use problem

This section exists because works councils are named in this paper's audience, and a paper that names them without addressing them is not addressing them.

The evidence architecture described in Chapter 7 produces a complete, personally attributable, cryptographically protected record of who instructed what, who approved what, and when. That is precisely what makes an action reconstructable. It is also, without any change to the design, an instrument capable of monitoring the conduct and performance of employees.

In Germany, a technical system that is *objectively suitable* for monitoring employee conduct or performance engages the works council's co-determination right under § 87(1) no. 6 BetrVG — regardless of whether the employer intends to use it that way. An evidence layer of the kind described here will meet that test. Comparable rules exist in other jurisdictions with statutory employee representation.

Under the GDPR, the same architecture raises purpose limitation and data minimisation questions in a sharper form than ordinary logging does, because the records are designed to be durable and non-repudiable — which is in tension with erasure and rectification rights. §7.8.1 describes the separation of payload data from integrity evidence that is intended to address this; whether it does so adequately is a question for a data protection impact assessment, not for a vendor's whitepaper.

What follows for anyone deploying this pattern:

1. **Involve employee representatives at design time, not at rollout.** The architecture decisions that matter — retention per record class, which fields are personally identifiable, who may query the evidence store, whether individual-level reporting is possible at all — are cheap to change early and expensive to change late.
2. **Separate the audit purpose from the performance-management purpose technically, not only in policy.** Access to the evidence store for incident reconstruction and access for any form of individual performance view should be distinct, separately authorised, and separately logged.
3. **Write the works agreement around the tier model.** Tier 0–3 actions generate operational records; tier 4–5 actions generate approval-bound evidence with named individuals. Those are different categories and deserve different treatment.
4. **Expect the question "can this be used to rank us?" and have a technical answer.** "We would not do that" is not one.

A works council reading Chapter 7 without this section would reasonably conclude that it describes surveillance infrastructure. It describes infrastructure that can be either, and the difference is a set of design decisions that should be made jointly.

3.6 State changes after approval: the TOCTOU limit

An approval can be perfectly bound to the action presented and still be unsafe to execute later. The state that justified the approval may have changed. This is not an AI-specific problem, but agents increase the number of delayed and chained actions for which it appears.

The architecture can reduce the risk by recording material preconditions, assigning the approval a short validity period, revalidating policy and revocation immediately before commit, and using conditional target-system operations where available. It cannot make an external system atomic when that system offers no transaction or concurrency primitive.

Where a target cannot support a conditional commit, the assurance claim must be lowered. A read-then-write adapter can detect many changes but still contains a race. Compensating transactions can repair some outcomes but do not make an irreversible action unhappen. The evidence record must therefore state the execution mode used: atomic conditional commit, locked transaction, immediate re-read, or best-effort with compensation.

3.7 Signatures, time and long-term evidence have a lifecycle

A valid signature demonstrates that a holder of a signing key signed a particular representation. It does not by itself prove that the signer's clock was correct, that the key was uncompromised at the relevant time, that the certificate was valid then, or that the cryptographic algorithm will remain trustworthy for the retention period.

Long-lived evidence therefore needs more than a current signature:

- a defined and monitored time source;
- a trustworthy timestamp where the timing assertion matters;
- signing-key separation, rotation and revocation;
- preservation of certificate chains, revocation information and verification metadata;
- algorithm agility and re-sealing before an algorithm or key becomes unsuitable;
- documented handling of key compromise;
- independent checkpoints for evidence that must survive compromise of the operating domain.

RFC 3161 provides a time-stamp protocol for proving that a datum existed before a particular time. RFC 4998 defines Evidence Record Syntax for preserving existence and integrity evidence over long archival periods. These are building blocks, not a complete deployment design. A v0.4 evidence implementation must state which time, signature and renewal profile it uses, and must avoid the broad claim of non-repudiation where the operational prerequisites are not met.

3.8 What the control plane does answer

> **How can organisational authority and technical execution be connected so that AI-assisted actions can be controlled, bounded and evidenced?**

It is one component of a broader governance and security architecture. What becomes indispensable is not a product labelled "runtime control plane" — it is the architectural capability to connect identity, mandate, policy, approval, execution and evidence at runtime.

---

4. Governed Intelligence: a working definition

4.1 From governance documents to executable governance

Governance in most organisations is a body of documents: policies, role descriptions, approval processes, risk assessments, training material, procedures, audit reports. These remain necessary. Their effect is bounded when the underlying rules cannot be evaluated or enforced during technical execution.

> **A policy describes what should happen. A control architecture decides whether it may happen. Evidence shows what did happen.**

Governed Intelligence connects the three.

4.2 Definition

> **Governed Intelligence** is an operating state in which an AI-assisted action passes an enforceable, reviewable and revocable control chain of intent, identity, mandate, context, policy, approval, execution and evidence.

The chain:

> **Intent → Identity → Mandate → Context → Policy → Approval → Execution → Evidence**

This is the canonical statement of the chain. It appears in full only here and in the executive summary; everywhere else the paper refers to "the control chain (§4.2)".

4.3 The eight elements

1. Intent — what is supposed to happen

The original instruction is captured as a structured intention: not only the prompt, but the intended business transaction — summarise information, produce a draft, update data, prepare a purchase order, initiate a payment, change a configuration.

Who structures the intent, and does that put the model in the authority path?

In practice, a language model usually performs the structuring. Left unaddressed, this contradicts §7.9 ("the model must not decide what it is permitted to do"). The resolution is explicit:

- Model-assisted structuring produces a **proposal**, never an authority artefact.
- The proposal is validated against a predefined intent schema; anything outside the schema is rejected rather than interpreted.
- For tier 3 and above, the structured intent is presented to the principal in human-readable form for confirmation before it becomes the basis of a policy decision.
- The evidence record marks the intent's **origin**: model-generated, user-confirmed, or system-generated.

The original wording may be retained alongside, but must not be the sole machine-readable basis of an authority decision.

2. Identity — who or what is acting

Every relevant entity needs a distinct identity: natural person, business role, application, service, agent, agent instance, model, external provider.

An agent must not appear as an anonymous process behind a general service account. This is workload identity as understood by SPIFFE/SPIRE, extended to agent instances and their parent-child relationships (see Chapter 5).

3. Mandate — on whose behalf and with what authority

The mandate links an agent identity to a legitimate principal. It defines the principal, the permitted purpose, the scope of authority, permitted systems and tools, time bounds, monetary or risk limits, delegation rules and revocation.

> A technical entitlement answers: *can this system reach that?* A mandate answers: *may it take this action, for this principal, for this purpose?*

Mandates should be as short-lived and as narrowly scoped as practical.

4. Context — under what conditions

Context may include tenant and legal entity, business unit and cost centre, data classification, affected data subjects, geographic region, model or provider location, target system, risk class, transaction value, current security posture, and prior decisions in the same process.

The same agent may hold different rights in different contexts.

5. Policy — which rules apply now

The policy engine evaluates intent, identity, mandate and context against versioned rules. Possible outcomes: permit; deny; reduce data scope; require a different model; enable additional logging; require human approval; require a second approval; escalate to a specialist.

Every decision must be traceable to the specific policy version in force at decision time.

6. Approval — which human authorisation is required

Not every action needs manual approval; the requirement must be risk-based against the tier model in §2.4. An agent may read information and produce a draft, while publication, payment or an entitlement change requires confirmation.

What matters is that the approval is **bound** to the specific action: defined content, recipient, amount, time and target resource — per §3.2, to the **provenance** of each parameter, and per §3.6, to the material execution preconditions and an expiry. If a material parameter or precondition changes, the approval is stale and must not be reused.

7. Execution — what actually runs

The action is not executed through unbounded, permanently stored credentials, but through controlled execution points: action broker, tool gateway, policy enforcement point, short-lived tokens, transaction-bound credentials, permitted command profiles, narrow API scopes.

Execution must be constrained to the approved action. Immediately before commit, the enforcement point must also revalidate mandate status, policy freshness, approval expiry and material target-state preconditions.

8. Evidence — what can be proven later

An evidence record links instruction, identities, mandate, context, policy decision, approvals, executed action, parameter provenance, precondition state, result, decision and execution timestamps, cryptographic profile and relevant versions.

Evidence is more than a log. A log shows a technical event. Evidence establishes the relationship between organisational authority and technical action.

4.4 Ten control invariants

The eight elements describe the chain. The following invariants describe the minimum properties that must remain true regardless of product choice or deployment style. Annex G turns them into candidate conformance requirements.

ID	Invariant	Failure that it prevents or exposes
GI-1 Principal attribution	Every governed action resolves to a human or organisational principal and a distinct acting workload	Anonymous service-account action
GI-2 Authority separation	Business applications and models cannot mint, widen or self-approve their own mandates	Self-escalation and circular approval
GI-3 No high-impact bypass	Tier 4 and 5 target operations have no privileged path outside the enforcement point, except a separately controlled break-glass path	Compromised application bypass
GI-4 Proposal is not authority	Model-produced intents and parameters remain proposals until schema validation and required confirmation	Model entering the authority path
GI-5 Approval equivalence	The approval is cryptographically or transactionally bound to the exact action and provenance shown to the approver	Approval reused for a different action
GI-6 State freshness	Material preconditions are checked at commit; stale approvals are rejected or re-evaluated	TOCTOU execution under changed state
GI-7 Revocation before commit	Mandate, credential, policy and approval revocation are checked at the last practical point before execution	Action continuing after leaver event or incident
GI-8 Evidence independence	The application that benefits from an action cannot silently rewrite the complete decision-and-execution history	Self-authored audit trail
GI-9 Tenant and key isolation	A tenant compromise does not confer another tenant's authority, keys or evidence	Cross-tenant blast radius
GI-10 Safe failure	Loss of an authority component moves each impact tier to a pre-declared safe state that is tested	Fail-open behaviour discovered during incident

A system does not need to be a single product to satisfy these invariants. A composed estate can conform if the boundaries and proofs are real. Conversely, a platform marketed as a control plane does not conform merely because it contains components with these names.

5. Relationship to prior art

Nearly every mechanism in this paper exists already, under a different name, in a standard or product that predates it. Stating that plainly is not a weakness in the argument; concealing it would be.

5.1 What already exists, and what it covers

Capability in this paper	Existing standard or practice	What it already covers	What is typically outside its model
Policy Decision Point / Policy Enforcement Point (§7.4)	**XACML** (OASIS; v1.0 2003, v3.0 2013); **NIST SP 800-207** Zero Trust Architecture	The decision/enforcement split, PDP/PEP/PIP vocabulary, attribute-based evaluation	Purpose and monetary limits as first-class inputs; binding a decision to a *specific approved transaction*
Policy as code	**Open Policy Agent / Rego**, **AWS Cedar**	Versioned, testable, externalised authorisation logic	The core decision mechanism is reusable. Domain-specific mandate semantics, tenant lifecycle, approval binding, evidence correlation and operations still require a profile and integration
Authority before access (§7.4, §7.5)	**Zero Trust**, "never trust, always verify"	Continuous verification, no implicit network trust	Verification of *organisational* authority, as distinct from technical identity and posture
Mandate (§4.3.3)	**OAuth 2.0 Token Exchange (RFC 8693)** `act` / `may_act` claims; **macaroons** (Birgisson et al., 2014); **GNAP**; **UCAN**	Delegation semantics, actor chains, attenuable caveats, third-party discharge	Business purpose, risk ceiling, approval obligation and revocation status as attributes of the delegation itself
Least privilege, just-in-time access (§7.2)	**PAM/PIM** practice; JIT elevation in major IdPs	Time-bounded elevation, approval workflows, session recording	Binding elevation to a single business transaction rather than a session
Agent and workload identity (§7.2)	**SPIFFE / SPIRE** (since 2018); mTLS; SCIM for lifecycle	Cryptographic workload identity, attestation, short-lived SVIDs, federation	Agent *instance* identity with parent-child lineage and principal attribution
Session and entitlement revocation	**CAEP / Shared Signals**, OpenID Connect back-channel logout	Near-real-time session revocation across relying parties	Revocation of an in-flight *mandate* mid-process
Hash chaining, Merkle structures, witnesses, checkpoints (§7.8.2)	**Certificate Transparency (RFC 6962)**, Trillian, transparency logs generally	Append-only logs, inclusion and consistency proofs, independent witnessing	The core construction is reusable. Confidential enterprise evidence additionally needs access control, selective disclosure, retention rules and private or federated witness operation
Signed releases, SBOM, reproducible builds (§11.3)	**SLSA**, **in-toto**, **Sigstore**, CycloneDX/SPDX	Provenance attestation, artefact signing, dependency transparency	The building blocks are reusable. Product-specific build policy, key custody, release authority, incident response and verification evidence remain operational work
Canonical signed action artefact (§7.12)	**RFC 8785** JSON Canonicalization; **JWS (RFC 7515)**; **COSE (RFC 9052/9053)**	Deterministic representation and standard signature containers	The domain schema, authority semantics, provenance fields and processing state machine
Trusted time and long-term evidence (§7.8.4)	**RFC 3161**, **RFC 4998 / RFC 9169**	Time-stamp tokens and archive evidence records	Operational trust model, key lifecycle, privacy, renewal policy and transaction correlation
State and concurrency binding (§1.4, §7.7)	Conditional writes, optimistic concurrency, ETags / `If-Match`, idempotency keys, transactional and saga patterns	Detecting stale state, preventing duplicate commit, handling partial failure	A cross-system rule for which preconditions must be bound to a human approval and carried into evidence
Management system for AI governance	**ISO/IEC 42001**, **NIST AI RMF 1.0**	Organisational governance, risk management process, documentation, continual improvement	Runtime enforcement; both are management-system and process frameworks, not execution controls

Capability in this paper	Existing standard or practice	What it already covers	What is typically outside its model
Information security management	**ISO/IEC 27001**, SOC 2	Control framework, audit evidence, certification path	Per-transaction authority binding
Agent-to-tool integration	**Model Context Protocol (MCP)**	A standard transport and description format for tool access	Authority: MCP describes *how* to call a tool, not *whether this actor may*, for what purpose, under whose mandate. The MCP security discussion — tool-description poisoning, confused-deputy risk in server chaining, consent fatigue — is directly relevant to §3.2
AI trust, risk and security management	Vendor and analyst category commonly labelled **AI TRiSM**; AI gateways	Model access control, prompt/output filtering, usage visibility, cost control	The action layer: what the output is permitted to *do*

5.2 What is actually being claimed as new

Given the table above, the honest claim is narrow:

1. **Purpose, risk limit and approval obligation as attributes of the delegation itself**, rather than as policy evaluated separately from the token. RFC 8693 can carry arbitrary claims; we are not aware of a widely adopted profile that standardises business purpose, monetary or impact ceilings, approval obligations and assignment-level revocation together.
2. **Approval bound to parameters *and their provenance*** (§3.2). Document-object binding is standard in ERP workflow. Binding across systems, to an action assembled by an agent from multiple sources, with each source recorded, is not.
3. **One correlated evidence record spanning identity, mandate, policy version, model version, tool version, approval and execution result** — as a first-class artefact with its own schema and retention model, rather than as a join performed after the fact in a SIEM.
4. **A composition** of the above with proven components (OPA or Cedar for decisions, SPIFFE for workload identity, transparency-log constructions for evidence integrity, SLSA/Sigstore for supply chain) rather than a replacement for them.

Points 1 to 3 are, we believe, defensible contributions. Point 4 is deliberate: an organisation that already runs an identity provider, a policy engine, a service mesh and a SIEM has most of the parts. What it usually lacks is the mandate model, the parameter-provenance-aware approval binding and the unified evidence schema — and it is entirely legitimate to build those on top of what it has, rather than buying a platform.

5.3 What this means for a buying decision

A reader evaluating this architecture should ask, in order:

1. Can we express purpose, risk limit and approval obligation in our existing token or policy model? If yes, do that.
2. Can we bind an approval to parameters, provenance and material preconditions across systems with what we have? In many estates this requires custom cross-system integration — this is where implementation work concentrates.
3. Can we produce one correlated record per governed action, or are we joining logs after the fact? If joining, measure how long a reconstruction actually takes; that number is the business case, and §10.5 and Annex H explain how to measure it.
4. Only then: is this better bought as an integrated control plane, or assembled?

We would rather a reader answer "assembled" and solve the problem than adopt a product and not.

5.4 Why publish a composition profile

A composition profile is useful even when none of its cryptographic or authorisation primitives is novel. Without a profile, two implementations can both claim to use workload identity, policy as code and signed logs while disagreeing on the questions that determine interoperability and assurance:

- What exactly is signed?
- Which identity is the principal and which is the acting workload?
- How is a mandate represented and revoked?
- Which parameter sources are visible to an approver?
- Which target-state preconditions invalidate an approval?
- What constitutes the execution receipt?
- Which records must be independently verifiable?
- Which failures require a block rather than a warning?

Annex G therefore defines GAIP-0.1 as a **candidate interchange and verification profile**, not as a claim of invention. A conforming implementation may be Vianordis, a collection of existing enterprise components, or an independent project. The value of the profile is that the same action can be evaluated by the same questions.

---

6. Regulatory convergence becomes an architecture problem

6.1 No single regulation requires a control plane

Neither the AI Act nor the GDPR, NIS2 or DORA prescribes a Vianordis-like platform. The instruments pursue different objectives:

The AI Act addresses risks of specific AI systems and the roles along the AI value chain. The GDPR protects personal data and the rights of data subjects. NIS2 and the German BSI Act require appropriate cybersecurity and risk management from in-scope entities. DORA specifies digital operational resilience for the financial sector. The Data Act governs access to and sharing of data from connected products, and cloud switching. The Cyber Resilience Act governs security properties and vulnerability handling for products with digital elements. Sectoral rules add further requirements.

At the technical level these objectives keep meeting at the same points: identities and roles, data access, third parties, technical entitlements, approvals, logging, monitoring, risk decisions and demonstrability.

The convergence does not arise because the laws are alike. It arises because their implementation lands on the same IT systems and business processes. A shared technical control layer can satisfy several of them with fewer separate proofs — which is an efficiency argument, not a legal necessity.

6.2 EU AI Act: what applies when

Regulation (EU) 2024/1689 as amended by Regulation (EU) 2026/1744 (the Digital Omnibus on AI, adopted 8 July 2026, published in the Official Journal on 24 July 2026, in force since 27 July 2026).

6.2.1 Application timeline

Scope	Applies from
Chapters I and II — general provisions, prohibited practices (Art. 5), AI literacy (Art. 4)	2 February 2025
Chapter III Section 4, Chapter V (GPAI models), Chapters VII and XII, Art. 78	2 August 2025
General applicability; Art. 50 transparency; Chapter VI; governance and penalties	**2 August 2026**
Transitional deadline for Art. 50(2) machine-readable marking, for synthetic-content systems already placed on the market before 2 August 2026	2 December 2026
New Art. 5 prohibitions introduced by the Digital Omnibus (non-consensual intimate imagery, CSAM)	2 December 2026
Chapter III Sections 1–3 — the core high-risk obligations — for Art. 6(2) / Annex III systems	**2 December 2027**
Chapter III Sections 1–3 for Art. 6(1) / Annex I systems (safety components in regulated products)	**2 August 2028**
AI regulatory sandboxes (Art. 57) — Member State obligation to establish, deferred by one year; not an application date for obligations	2 August 2027

The deferral is unconditional. The Commission's November 2025 proposal had tied the dates to a finding on the availability of harmonised standards; that trigger is not in the final Article 113.

Two consequences for planning. First, the obligations most often cited as the reason to act now — logging, human oversight, deployer duties — do not apply until December 2027 at the earliest. Second, that is roughly one budget cycle plus one implementation cycle, which for an estate-wide architecture change is not generous.

6.2.2 What the high-risk regime will require

For systems in scope from 2027/2028, the AI Act provides among other things:

- **Art. 12** — the system must technically enable automatic recording of events over its lifetime. Note that Art. 12 is a design obligation; retention follows from Art. 19 and Art. 26(6).
- **Art. 14** — effective human oversight must be *enabled by the provider*. Under Art. 14(4), oversight persons must be able to correctly interpret the output, "to decide not to use the high-risk AI system or to otherwise disregard, override or reverse the output", and "to intervene or interrupt the system through a 'stop' button or a similar procedure". Art. 14(5) adds a four-eyes requirement for remote biometric identification.
- **Art. 19** — providers retain automatically generated logs under their control for **at least six months**, with a special rule for financial institutions.
- **Art. 26** — deployer obligations: use in accordance with the instructions, appropriate technical and organisational measures, assignment of human oversight to natural persons "who have the necessary competence, training and authority" (Art. 26(2)), monitoring of operation and reporting (Art. 26(5)), and log retention of at least six months (Art. 26(6)).
- **Art. 25** — role reallocation. A party that puts its name or trademark on a high-risk system, makes a substantial modification, or changes the intended purpose such that the system becomes high-risk, becomes the provider of the resulting system. Art. 25(4) contains a separate carve-out for tools and components supplied under a free and open-source licence.
- **Art. 27** — the fundamental rights impact assessment is retained, and may be satisfied by cross-reference to a DPIA to the extent the Art. 27 requirements are covered there.

Note the division of labour in Art. 14 and Art. 26: **enabling** oversight is a provider obligation; **assigning** it to competent people is a deployer obligation. Organisations that integrate, rebrand or substantially modify a system may find they hold both.

6.2.3 Changes introduced by the Digital Omnibus that are easy to miss

- **Art. 4 (AI literacy) weakened** — from "shall ensure … a sufficient level of AI literacy" to an obligation to take measures to support the development of AI literacy, with no requirement to guarantee a particular level. An obligation of means, not of result.
- **"Safety component" narrowed** (Art. 3(14) and Art. 6) — systems that concern only user assistance, performance optimisation, efficiency, automation, comfort or quality control are not safety components. This materially reduces the Annex I high-risk perimeter.
- **New Art. 5 prohibitions from 2 December 2026** — AI systems for generating or manipulating non-consensual intimate imagery and child sexual abuse material, including where this is "reasonably foreseeable" without substantial technical modification. Penalty band: 35 million euro or 7 %. This is the clearest instance in the Act of a requirement that can only be met by runtime safeguards rather than documentation.
- **New Art. 4a and adjusted Art. 10(5)** — processing of special categories of personal data for bias detection and correction, extended to providers and deployers of non-high-risk systems and models, subject to safeguards including pseudonymisation, access controls and deletion. A direct GDPR interface.
- **Art. 75 — enforcement shifts to the AI Office** for AI systems built on a GPAI model of the same provider and for systems integrated into VLOPs/VLOSEs, with extended investigation and fining powers.
- **Registration retained.** The proposal to remove the Art. 6(3) / Art. 49 registration duty for systems self-assessed as non-high-risk was **not** adopted; the duty remains, with Annex VIII Section B items 7 and 9 removed.
- **New "small mid-cap" category** — simplified quality management and technical documentation, priority sandbox access, and capped penalties.
- **Machinery Regulation (EU) 2023/1230 moved** from Annex I Section A to Section B, with a delegated act due by 2 August 2028.

6.2.4 Penalties

Band	Amount	Scope
Art. 99(3)	up to **35 million euro or 7 %** of total worldwide annual turnover, **whichever is higher**	Non-compliance with the Art. 5 prohibitions
Art. 99(4)	up to **15 million euro or 3 %**, whichever is higher	Most other obligations of providers, deployers, importers, distributors and notified bodies
Art. 99(5)	up to **7.5 million euro or 1 %**, whichever is higher	Supplying incorrect, incomplete or misleading information to notified bodies or national authorities
Art. 99(6)	for SMEs and start-ups, each band is capped at the **lower** of the two figures	—
Art. 100	up to 1.5 million / 750,000 euro	EDPS fines on Union institutions
Art. 101	up to **3 % or 15 million euro**	Commission fines on GPAI model providers; applicable from 2 August 2026

The Digital Omnibus left the amounts and the tier structure unchanged. It did extend the reduced-cap treatment to the new small mid-cap category and adjusted which infringements fall into the Art. 99(4) band.

6.2.5 Open source under the AI Act

Art. 2(12) excludes AI systems released under free and open-source licences from the Regulation's scope, unless they are placed on the market or put into service as high-risk systems, or as systems falling under Art. 5 or Art. 50.

Three limits matter for Chapter 11 of this paper:

1. The exclusion covers **AI systems**, not **GPAI models**. Models fall under the narrower reliefs of Art. 53(2) and Art. 54(6), which fall away entirely where systemic risk is present.
2. Monetised or not-genuinely-open distributions are not covered, per the recitals. **This bears directly on any dual-licensing model, including ours** — whether an AGPLv3-plus-commercial construction still falls within Art. 2(12) is a real question and not one we consider settled.
3. Art. 25(4) provides a separate FOSS carve-out for value-chain suppliers.

6.3 Germany: NIS2 and the BSI Act — management liability is already live

The *Gesetz zur Umsetzung der NIS-2-Richtlinie und zur Regelung wesentlicher Grundzüge des Informationssicherheitsmanagements in der Bundesverwaltung* (NIS2UmsuCG) of 2 December 2025 (BGBl. 2025 I no. 301, issued 5 December 2025) brought the new BSI Act (BSIG 2025) into force on **6 December 2025**, with no general transition period for the substantive duties.

§ 30 BSIG requires essential and important entities to take appropriate, proportionate and effective technical and organisational risk management measures, proportionate to risk exposure, size, cost and potential damage, and to **document** them. § 30(2) lists ten areas: risk analysis and security concepts; incident handling; business continuity, backup and crisis management; supply chain security; security in acquisition, development and maintenance of systems; effectiveness assessment; training and cyber hygiene; cryptography and encryption; personnel security, access control and asset management; and multi-factor authentication, secured communications and emergency communications.

§ 38 BSIG is the provision that changes the conversation with a board:

- **§ 38(1)** — management bodies of essential and important entities are obliged to implement the § 30 risk management measures and to supervise their implementation.
- **§ 38(2)** — *management bodies that breach their duties under subsection 1 are liable to their entity for culpably caused damage, under the company-law rules applicable to the entity's legal form.*
- **§ 38(3)** — regular training obligation for the management body itself.

This is internal liability toward the entity, fault-based, measured by the standard applicable to the legal form (§ 43 GmbHG, § 93 AktG and equivalents). It is distinct from administrative fines against the entity under § 65 BSIG. It is in force now, unlike the AI Act's high-risk regime.

Staged deadlines that follow from the Act and are easy to miss:

Duty	Deadline
Registration (§ 33 BSIG)	within **three months** of first becoming in scope — for entities in scope at entry into force, generally **6 March 2026**
BSI portal for registration and reporting	live since 6 January 2026
Notification of changes	without undue delay, at the latest within two weeks
Incident reporting (§ 32)	**24 h** early warning / **72 h** report / **one month** final report
KRITIS evidence (§ 39(1))	first at a date set by the BSI, thereafter every three years; § 39(3) contains a transition rule for entities designated under the 2009 BSIG

The 24-hour early-warning clock is itself an argument for correlated evidence. In an agent-driven incident, the first 24 hours are spent determining what the agent did, on whose behalf and with what data. Organisations that have to join four log sources by hand to answer that will miss the window.

Note also that the BSIG consistently uses *informationstechnische Systeme* / *Informationstechnik*, not the EU-level "ICT" vocabulary used by DORA and NIS2 itself. A further amendment is pending: the draft CRA implementation act (BT-Drs. 21/6134 of 26 May 2026) designates the BSI as market surveillance and notifying authority.

6.4 DORA: responsibility survives outsourcing

Regulation (EU) 2022/2554 has applied to in-scope financial entities since **17 January 2025**.

Art. 5(1) requires an internal governance and control framework ensuring effective and prudent management of ICT risk. Art. 5(2) provides that the management body "shall define, approve, oversee and be responsible for the implementation of all arrangements related to the ICT risk management framework", and Art. 5(2)(a) assigns it **ultimate responsibility**. Art. 6(1) requires a "sound, comprehensive and well-documented ICT risk management framework"; Art. 6(4) requires an independent control function with separation along three-lines principles. Art. 16 provides a simplified framework for microenterprises and certain small entities — the general statement does not apply uniformly to all financial entities.

On third parties, Art. 28(1)(a) is the operative sentence: financial entities "shall at all times remain fully responsible for compliance with, and the discharge of, all obligations under this Regulation and applicable financial services law". Art. 28(1) requires integration of ICT third-party risk into the **Art. 6(1) framework** specifically; Art. 28(1)(b) applies proportionality; Art. 28(2) requires a documented strategy reviewed regularly by the management body; Art. 28(3) requires the register of information.

Applied to AI:

> **Procuring a model, a cloud platform or an agent service does not transfer enterprise responsibility to the supplier.**

An organisation therefore needs more than a list of its AI suppliers. It needs a controllable view of the actions, data flows, dependencies and entitlements those suppliers enable.

6.5 GDPR: purpose and minimisation have to hold at execution time

The GDPR applies unchanged. Note for completeness: the Digital Omnibus proposal on data and privacy of 19 November 2025 — which is a **different instrument** from the AI Omnibus discussed above — remains in the legislative process as at August 2026. Proposed changes to pseudonymisation, legitimate interest as a basis for AI training, records of processing and breach notification timelines are **not** current law, and should not be planned against as if they were.

Two provisions matter most here.

Art. 25(1) requires the controller to implement appropriate technical and organisational measures both "at the time of the determination of the means for processing" and "at the time of the processing itself" — which is narrower than a continuous-supervision duty, and worth quoting precisely rather than paraphrasing.

Art. 25(2) — data protection by default — is the more directly applicable provision for agents, and is frequently overlooked. It requires that, by default, only personal data necessary for each specific purpose are processed, in terms of amount collected, extent of processing, storage period and accessibility. For an agent architecture this reads almost as a specification: default entitlements, default retention, default logging scope and default data breadth all fall under it. The least-privilege and default-deny principles in §7.2 and §7.5 are, in the personal-data case, an implementation of Art. 25(2).

Alongside these, records of processing activities under Art. 30 may be required — the sub-250-employee exemption in Art. 30(5) is defeated in practice by its own exceptions where AI processing is involved — and a data protection impact assessment under Art. 35 will frequently be triggered, typically via Art. 35(3)(a) where systematic and extensive evaluation or profiling is present.

An agent with technical access to a large corpus should therefore not be permitted to use all of it for every task. The control architecture must be able to take purpose, context and data classification into account at decision time.

6.6 Two instruments the earlier draft omitted

6.6.1 Cyber Resilience Act — Regulation (EU) 2024/2847

The CRA applies to "products with digital elements" placed on the EU market. Scope is narrower than often assumed: **pure software-as-a-service is generally outside it** and is addressed by NIS2 instead. What is caught is distributable or on-premises software, and remote data processing solutions that are integral to a product with digital elements.

The date that matters immediately: from **11 September 2026**, manufacturers must report **actively exploited vulnerabilities** on a 24-hour early-warning / 72-hour notification / final-report-no-later-than-14-days-after-a-corrective-or-mitigating-measure-is-available sequence. Severe incidents follow a 24-hour / 72-hour / one-month sequence. Notified bodies have been operating since 11 June 2026, and the Regulation applies in full from **11 December 2027**.

For this paper's argument the CRA matters twice over: it imposes on suppliers of installable control-plane software exactly the supply-chain and vulnerability-handling discipline described in §11.3, and it gives enterprise buyers a lawful basis to demand it. **To the extent we distribute installable components, or operate a remote data processing solution integral to one, it applies to us**; which components qualify is under assessment, and the status is stated in Annex C.6.

6.6.2 Data Act — Regulation (EU) 2023/2854

Applicable since 12 September 2025, with design obligations for new connected products from 12 September 2026 and cloud-switching provisions — including the removal of switching charges — from 12 January 2027.

This is the instrument that turns the portability argument in §7.10 from a preference into a legal entitlement. An organisation arguing internally for provider-independent identities, policies and evidence now has a regulation behind it rather than only an architectural principle.

6.6.3 Worth watching: eIDAS 2.0

Regulation (EU) 2024/1183 requires Member States to make EU Digital Identity Wallets available by end-2026, with acceptance obligations for certain private service providers from 2027. For the "who is actually acting" question in §4.3.2, this is the most relevant identity development in the EU and should be tracked by anyone designing an agent identity model now.

6.7 Shared technical objectives

Regulatory objective	Possible technical support
Accountability	Distinct identities for people, services and agents
Access control	Short-lived, purpose-bound entitlements
Purpose limitation	Mandates with a stated purpose and permitted context
Human oversight	Risk-based approvals, escalation and interruption
Traceability	Correlated event and decision records
Resilience	Controlled execution, repeatability, blocking and restart mechanisms
Third-party governance	Model and provider abstraction with documented data paths
Demonstrability	Versioned policies, bound approvals, evidence records
Data minimisation	Context-dependent limitation of the information supplied
Supply chain security	Traceable components, dependencies and signed releases
Vulnerability handling	Coordinated disclosure, SBOM, reporting readiness within 24 h

This mapping does not create compliance. It shows where a shared architecture can provide reusable controls. Annex A maps the same capabilities against individual instruments and management-system standards.

7. Architecture of a runtime control plane

The architectural principles that appeared as a separate chapter in v0.2 are integrated into the relevant sections here, to remove a redundancy readers of that draft found tiring.

7.1 Reference flow

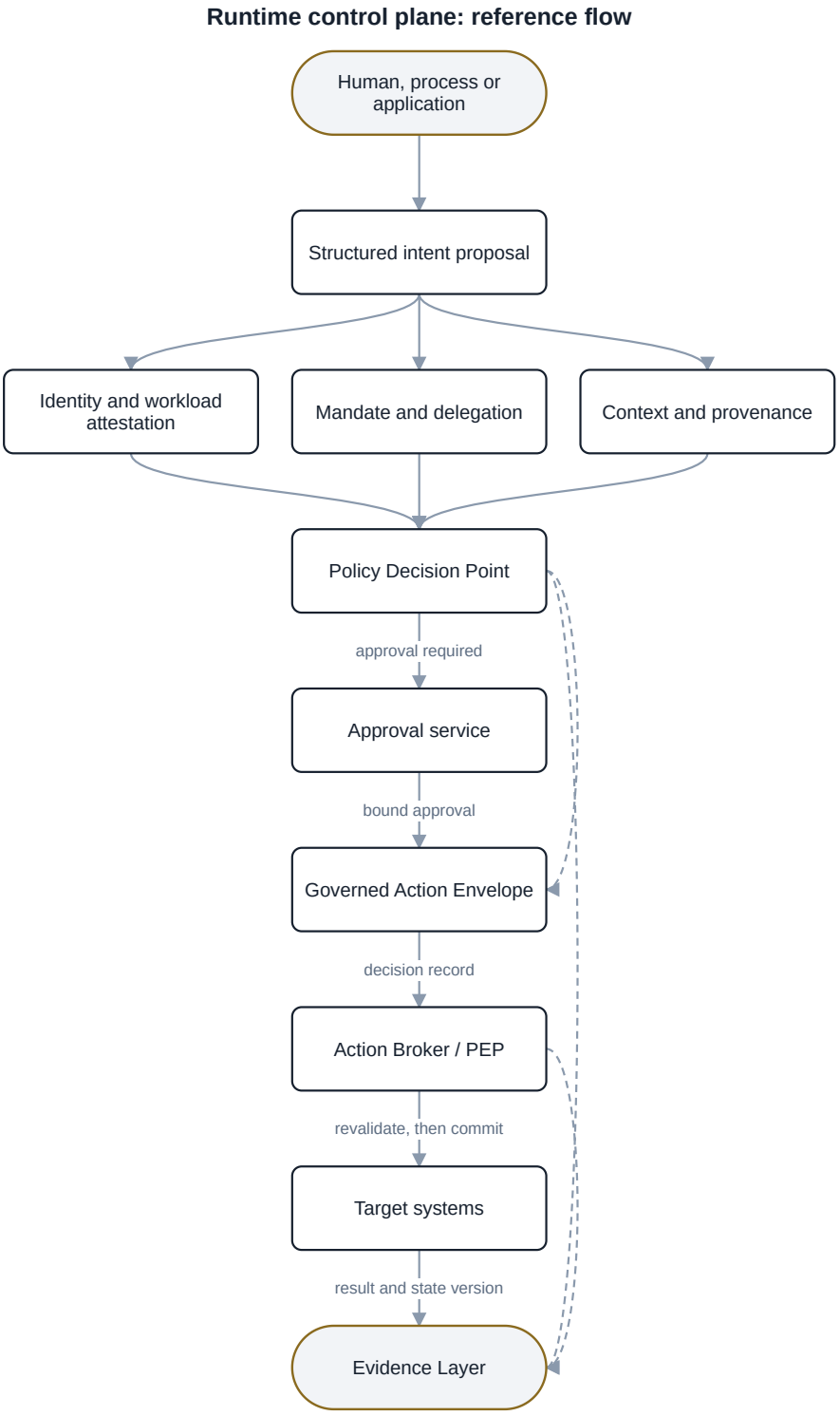

Figure 2. Authority, approval, execution and evidence are four separate paths.

The diagram draws four paths separately:

1. **authority path** — identity, mandate, context and policy produce a decision;
2. **approval path** — a human approves a specific action, provenance and precondition set where required;
3. **execution path** — the broker revalidates and commits through a controlled target adapter;
4. **evidence path** — the decision record and execution receipt are independently joined and protected.

Payload remains in the system of record wherever practical. The evidence layer receives references, digests and the minimum metadata needed to verify the chain. Revocation, incident and policy-change events feed back into the control plane.

The Mermaid source is intentionally included in the document so that publishing systems can render it as SVG without making the figure an opaque image.

7.2 Identity for people and machines — and least privilege

Classical identity systems concentrate on people, applications and technical services. An agent-capable architecture additionally needs distinct agent identities; identities for individual agent instances; parent-child relationships between agents; attribution to a human or organisational principal; documented ownership; status and lifecycle; certificates or cryptographic attestation; and immediate revocation.

An agent must not be trusted merely because it runs inside the corporate network.

Entitlements follow least privilege and just-in-time principles: time-bounded rights, task-specific scopes, transaction-bound tokens, automatic withdrawal at task end. In the personal-data case this is not only good practice but an implementation of GDPR Art. 25(2).

This layer should be built on SPIFFE/SPIRE or an equivalent workload identity system rather than reinvented (see §5.1).

7.3 Mandate and authority service

The mandate service separates long-lived organisational roles from short-lived agent assignments.

Field (EN)	German equivalent	Meaning
Mandate ID	Mandats-ID	Unique reference
Principal	Auftraggeber	Person or organisation on whose behalf action is taken
Agent Identity	Agentenidentität	The mandated agent
Purpose	Zweck	Business purpose
Allowed Actions	Zulässige Aktionen	Permitted action types
Systems and Tools	Zielsysteme und Werkzeuge	Permitted targets
Data Scope	Datenbereich	Permitted data domains
Risk Limit	Risikogrenze	Monetary, criticality or impact ceiling
Validity	Gültigkeit	Start and expiry
Delegation	Delegation	Permitted or prohibited sub-delegation
Approval Rules	Freigaberegeln	Required approvals
Revocation Status	Widerrufsstatus	Current revocation state

Mandates should be as short-lived and narrowly scoped as practical. The transport can be an existing one — RFC 8693 token exchange carries arbitrary claims — but Purpose, Risk Limit and Approval Rules must be part of the delegation artefact, not evaluated separately from it.

Terminology note: throughout this document, "Mandate" denotes the singular control-chain element. Readers of the German edition should note that German "Mandate" is a plural form; the German edition uses "Mandat" in running text.

7.4 Policy decision and policy enforcement

A complete architecture separates the **Policy Decision Point**, which decides what is permitted, from the **Policy Enforcement Point**, which enforces the decision technically. This vocabulary is XACML's and NIST SP 800-207's; the implementation should use an established policy engine such as OPA or Cedar rather than a bespoke one.

A decision of the form "this action is permitted only after approval" loses its protective effect if the agent can reach the target system by an alternative route. Relevant execution paths must therefore run through controlled enforcement points.

This separation must not exist merely logically inside the same compromisable business application. For consequential processes, decision and enforcement functions need their own technical identities, separate keys and clearly defined trust boundaries.

Policies must be **versioned**, and every decision traceable to the version in force at decision time. That is what later distinguishes correct application of a then-current rule from faulty execution and from a subsequently changed corporate policy.

7.5 Default deny for real-world actions

Where identity, mandate, context or policy is not unambiguous, a consequential action should not execute by default. The system may deny, switch to draft mode, reduce data access, or involve a human.

The tier model in §2.4 sets where this applies. Tier 0 and 1 defaults that deny will make the system unusable; tier 4 and 5 defaults that permit will eventually make it dangerous.

7.6 Human oversight as a technical function

Human oversight must not be merely an organisational note. An effective technical implementation needs: an intelligible presentation of the planned action; presentation of the data and assumptions used, **including the provenance of each parameter** (§3.2); risk indications; the ability to modify, reject or escalate; binding of the approval to the specific transaction; secure identification of the approving person; documented competence and authority; and an interrupt or stop capability.

From 2 December 2027 and 2 August 2028 respectively, the AI Act will require for applicable high-risk systems that human oversight is technically enabled and that persons can, among other things, correctly interpret outputs, disregard, override or reverse them, and interrupt the system through a stop button or similar procedure (Art. 14(4)). Deployers must assign that oversight to competent, trained and authorised persons (Art. 26(2)).

An approval must be bound to the concrete action. Confirmation of "transfer 1,200 euro to supplier A for invoice 4711" must not be reusable for "transfer 12,000 euro to supplier B". If a material element changes, the approval is void.

7.7 Action broker instead of standing full access

The action broker is the controlled transition from a decision to a real system change. It should accept only approved action types; validate mandate and policy decision; use short-lived credentials; check parameters and provenance against the approval; verify expiry and revocation; re-read or conditionally bind material target state; prevent repeats and replay; capture results; handle errors and partial states; and complete the execution receipt.

The objective is not to rebuild every application. Existing systems are connected through standardised adapters, APIs or MCP interfaces — with the caveat from §5.1 that MCP describes how to call a tool, not whether this actor may.

7.7.1 Commit-time state validation

For each action type the adapter must identify **material preconditions**. Examples: supplier-master version, purchase-order status, account ownership, current entitlement, current policy digest, available balance, target object version. The governed action envelope carries those preconditions and their permitted freshness window.

At commit the broker follows this order:

1. reject an expired envelope or approval;
2. re-check mandate, credential and policy revocation;
3. fetch or validate the material target-state versions;
4. re-run the policy decision if a relevant state or policy changed;
5. execute with an idempotency key and, where supported, a conditional operation such as compare-and-set or `If-Match`;
6. receive a target-system receipt containing the committed state or version;
7. emit the execution receipt and close, compensate or escalate the transaction.

An adapter that cannot perform steps 3 to 5 must declare a lower assurance mode in the evidence record.

7.7.2 Replay, idempotency and partial failure

Every governed action requires a globally unique transaction identifier and an idempotency key. A repeated request with the same key must return the existing outcome or fail closed; it must not create a second payment, account or publication.

Where multiple target systems are involved and no distributed transaction exists, the process must declare its compensation model. The evidence record distinguishes `committed`, `partially_committed`, `compensated`, `compensation_failed` and `indeterminate`. A clean audit trail must not hide an incomplete business state.

7.8 Evidence layer

The evidence layer correlates events across system boundaries.

Evidence field	Example
Transaction ID	Global transaction identifier
Tenant ID	Affected organisation
Principal Identity	Instructing person or role
Agent Identity	Acting agent and instance
Mandate ID and version	Mandate in force
Envelope profile and version	GAIP or other schema used
Intent	Structured business purpose, with origin marking
Model and version	Model used
Tool and version	Tool used
Data Classification	Data classes processed
Parameter provenance	**For each decision parameter: source (document hash, system of record, user entry), and whether independently verified**
Policy Decision	Permit, deny or escalate
Policy version	Rule set in force
Approval Evidence	Approving person, bound action, provenance, preconditions and expiry
Action Digest	Hash or reference of the approved action

Evidence field	Example
Precondition Digest	Material target-state versions or hashes used for approval and commit
Decision Record ID	Signed policy and approval decision reference
Execution Receipt	Broker and target-system receipt, including idempotency key and committed version
Model Evidence	Provider, deployment or endpoint, advertised model, request ID and reproducibility status
Time Data	Decision, approval and execution timestamps, clock or timestamp source
Cryptographic Profile	Canonicalisation, signature, timestamp and checkpoint profile
Outcome	Success, failure, rolled back
Incident Link	Link to deviations or incidents

The parameter provenance row is new in this edition and follows directly from §3.2. Without it, an evidence record survives an injection incident intact and tells the investigator nothing useful.

7.8.1 Separating payload data from evidence

Not all raw data needs to sit permanently in an immutable record. For personal or confidential data in particular, mutable and erasable payload must be separated from long-lived integrity proof.

One workable approach: leave payload data in its system of record; store only necessary metadata, references and cryptographic digests in the evidence layer; apply retention periods per data type; permit erasure and rectification of the payload; and still be able to demonstrate the integrity of the event chain.

A digest does not prove the legal content or the correctness of the referenced data. It can demonstrate whether an exactly referenced state has changed since the digest was created.

This design is intended to address the tension identified in §3.5 between durable evidence and data subject rights. Whether it does so adequately in a given deployment is a DPIA question.

7.8.2 Cryptographic tamper-evidence

Evidence records should be serialised canonically, hashed, and linked to preceding entries or to periodic Merkle structures. Related records or completed evidence periods can then be signed with a separately protected signing key. Key management must sit outside the business applications — a dedicated key management service or HSM. Signed checkpoints can additionally be anchored in an independent transparency log or with a separate witness service.

The construction to adopt here is Certificate Transparency's (RFC 6962), not a novel one.

These mechanisms do not prevent a compromised or highly privileged actor from altering or deleting records. They ensure that subsequent manipulation, missing entries and reordering become **detectable**. Append-only storage, immutable retention policies, separate replicas and independent checkpoints further reduce the risk that a single compromised service can silently rewrite the chain.

> **The goal is not the unrealistic claim of an "unalterable log", but defensible cryptographic tamper-evidence.**

7.8.3 Tiered assurance

Not every transaction needs the same evidential strength.

Assurance level	Mechanism	Example
Basic	Correlated, access-controlled logs with a central time source	Internal, reversible information action (tier 0–1)
Elevated	Canonical records, hash chaining and signatures	Approval-bound business process (tier 2–3)

Assurance level	Mechanism	Example
High	Append-only / WORM, separate replicas, signed checkpoints	Critical entitlement or financial action (tier 4)
Externally verifiable	Independent witness or transparency log	Evidence of particular regulatory or societal significance (tier 5)

The level chosen must match risk, retention requirements, data protection constraints and expected audit demand.

7.8.4 Trusted time and cryptographic lifecycle

A local application timestamp is an observation, not independent proof of time. The evidence profile must identify the time source for every material event and the maximum tolerated skew. Higher-assurance actions should combine monitored secure time synchronisation with an RFC 3161 time-stamp token or an equivalent qualified or independently operated service for signed checkpoints.

The evidence system also needs a cryptographic lifecycle:

- key identifiers and certificate chains retained with the evidence;
- keys generated and used inside an appropriate KMS or HSM boundary;
- rotation without losing the ability to verify historical records;
- revocation and a documented compromise procedure;
- periodic validation of checkpoints;
- algorithm agility and re-sealing before a hash or signature algorithm reaches end of life;
- preservation or renewal of long-term evidence using an ERS-like construction where the retention horizon warrants it.

A record whose key was later compromised is not automatically worthless, but its assurance depends on whether trustworthy evidence establishes that it existed before compromise.

7.8.5 Model and provider evidence

“Model and version” is not equally provable in every operating model.

For a self-hosted model, the record can contain an artefact digest, container or package digest, tokenizer version, runtime version and deployment configuration. For a provider-managed model, a reproducible artefact may not be available. The record should then distinguish:

- provider and legal service;
- endpoint or deployment identifier;
- advertised model alias;
- API version;
- provider request or trace identifier;
- provider-supplied system fingerprint or receipt, if available;
- relevant safety or tool configuration digest;
- **reproducibility status**: reproducible, provider-identifiable, alias-only, or unknown.

The evidence layer must not claim an exact model version when the provider only supplied a mutable alias.

7.8.6 Privacy-preserving access to evidence

A strongly correlated evidence store is also a high-value surveillance and breach target. Access must be purpose-bound and queryable without making unrestricted employee timelines the default interface.

At minimum:

- separate incident, audit, legal-hold and operational-support roles;
- field-level or view-level minimisation;
- query logging and alerting on bulk access;
- tenant-scoped encryption and keys;
- retention by record class and impact tier;
- pseudonymous operational views where named attribution is not required;
- controlled re-identification for an authorised investigation;
- export packages that disclose only the evidence needed for the stated purpose.

Cryptographic integrity does not justify unlimited retention or unlimited access.

7.9 Separating model from authority

The model must not decide what permissions it holds. A model may propose an action, derive parameters and explain risks. The authority decision must be made outside the model, by traceable rules.

Per §4.3.1, where a model structures the intent, the output is a proposal validated against a schema, marked with its origin, and — from tier 3 upward — confirmed by the principal.

Per §3.2, "derive parameters" is where the residual risk concentrates. Parameters derived from untrusted input carry that fact into the evidence record and, for tier 4 and above, require out-of-band verification of the fields that determine impact.

7.10 Portability, tenancy and sovereignty

Provider and model portability. The control architecture must not exist solely inside one AI provider's proprietary security model. Identities, mandates, policies and evidence must remain controllable independently of the selected model. Since the Data Act, cloud switching provisions give this a legal footing as well as an architectural one.

Tenant isolation. In a multi-tenant system, identities, policies, keys, evidence and data paths must be cleanly separated per tenant. A central control plane must not become a central, unsegmented trust anchor.

Sovereignty as control capability. Sovereignty is not only a server's geography. Technical sovereignty covers control over identities, keys, policies, data paths, provider substitution, operating models, interface and protocol standards, export capability, recovery and exit strategy.

A European hosting location without controllable identities, keys and dependencies is insufficient. Conversely, a portable architecture can integrate different European providers or self-operated components without surrendering its own authority logic.

7.11 Lifecycle, revocation and compromise containment

Governed Intelligence does not end at an agent's first approval. The lifecycle covers registration, ownership assignment, risk classification, review, activation, monitoring, policy re-evaluation, recertification, suspension, revocation and decommissioning.

Role changes, security incidents, new model versions, changed tools or amended policies can trigger re-evaluation.

The architecture must assume individual components will be compromised. The objective is not an infallible platform but bounded damage:

- a business application must not be able to issue mandates;
- an agent must not be able to widen its own scopes;
- a compromised model provider must not acquire technical authority;
- a compromised service account must not be able to silently rewrite evidence history;

- a compromised tenant component must not reach other tenants;
- a compromised administrator account must be bounded by key separation, four-eyes rules and independent evidence.

Annex B states the trust boundaries and the compromise scenarios this is intended to bound, including the input-manipulation case from §3.2.

7.12 The Governed Action Envelope

The control chain needs a transportable artefact. This edition calls it the **Governed Action Envelope**. It is the unit passed from policy decision and approval to the enforcement point, and the object against which the execution receipt is verified.

The envelope is not itself a credential and must not contain standing secrets. It is a signed, short-lived statement of what may be attempted under which conditions.

A simplified JSON representation:

```
{
  "profile": "GAIP-0.1",
  "transaction_id": "0191f2d1-7c5a-7c31-a15f-b15de9b72b85",
  "tenant_id": "tenant-42",
  "principal": {
    "id": "person:B",
    "organisation": "legal-entity:DE-01"
  },
  "actor": {
    "agent_id": "spiffe://tenant-42/agent/invoice-checker",
    "instance_id": "agent-instance:A-91"
  },
  "intent": {
    "type": "invoice.payment.execute",
    "schema": "urn:vianordis:intent:invoice-payment:1",
    "origin": "user-confirmed"
  },
  "mandate": {
    "id": "M-4711",
    "version": 3,
    "purpose": "settle approved supplier invoice",
    "risk_limit": {"currency": "EUR", "amount": 5000},
    "expires_at": "2026-08-14T13:35:00Z"
  },
  "parameters": [
    {
      "name": "amount",
      "value": "1200.00",
      "source": "document:sha256:...",
      "provenance_class": "P3",
      "verified_against": "erp:invoice:4711"
    },
    {
      "name": "payee_iban",
      "value_digest": "sha256:...",
      "source": "erp:supplier:V-238@version-14",
      "provenance_class": "P1"
    }
  ],
  "policy_decision": {
    "id": "D-882",
    "policy_digest": "sha256:...",
    "effect": "permit-with-approval"
  },
  "approval": {
    "approver": "person:C",
    "approved_action_digest": "sha256:...",
    "approved_precondition_digest": "sha256:...",
    "expires_at": "2026-08-14T13:35:00Z"
  },
  "preconditions": [
    {"resource": "erp:supplier:V-238", "expected_version": "14"},
    {"resource": "erp:purchase-order:9204", "expected_state": "open"}
```

```
  ],
  "execution_constraints": {
    "target": "erp:payment-api",
    "operation": "create-payment",
    "idempotency_key": "0191f2d1-7c5a-7c31-a15f-b15de9b72b85",
    "assurance_mode": "conditional-commit"
  }
}
```

The example is illustrative. Annex G defines the candidate field requirements and verification rules.

7.12.1 Canonicalisation and signature

The envelope must have a deterministic representation before hashing or signing. A JSON implementation may use RFC 8785 JCS; a CBOR implementation may use deterministic CBOR with COSE. JWS or COSE can carry the signature. The selected profile, algorithms, key identifier and critical headers must be explicit.

Signing the whole envelope is preferable to signing a set of loosely related tokens because it makes the object reviewed, approved and executed unambiguous. Sensitive parameter values may be represented by digests or encrypted disclosures where the enforcement point can resolve them securely.

7.12.2 Processing state machine

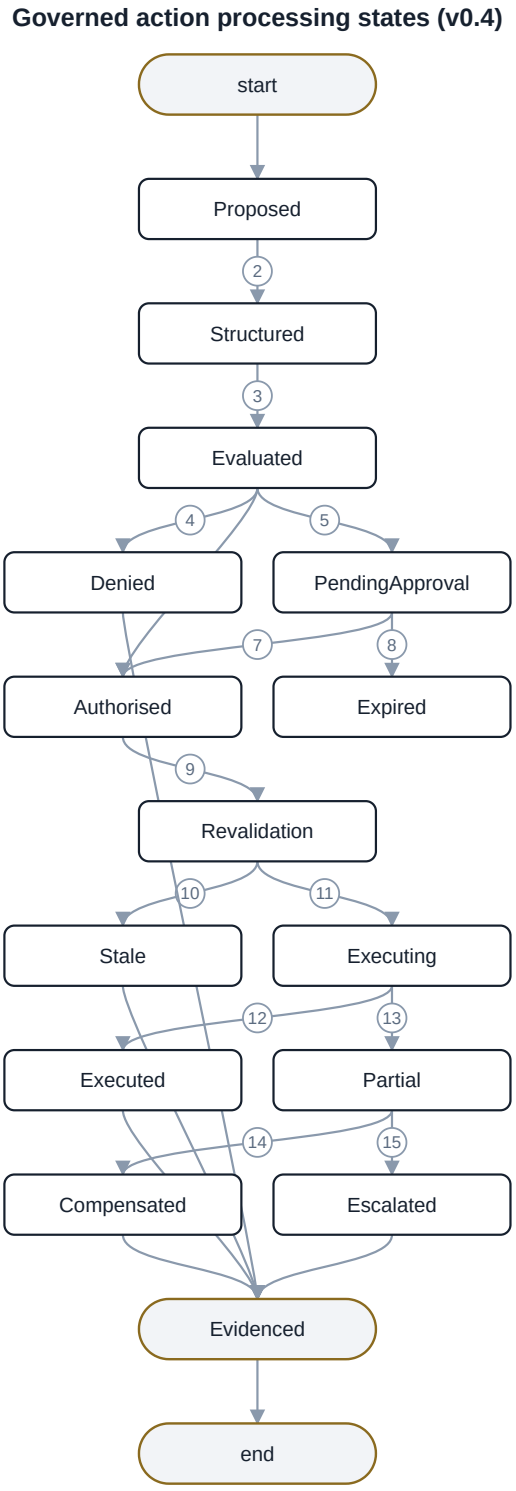

Figure 3. Processing state machine as published in v0.4.

#	Transition	Condition
2	`Proposed` → `Structured`	schema validation
3	`Structured` → `Evaluated`	identity, mandate, context, policy
4	`Evaluated` → `Denied`	deny
5	`Evaluated` → `PendingApproval`	approval required

#	Transition	Condition
6	`Evaluated` → `Authorised`	permit
7	`PendingApproval` → `Authorised`	bound approval
8	`PendingApproval` → `Expired`	timeout or state change
9	`Authorised` → `Revalidation`	broker receives envelope
10	`Revalidation` → `Stale`	revocation, expiry or precondition mismatch
11	`Revalidation` → `Executing`	conditions hold
12	`Executing` → `Executed`	target commit receipt
13	`Executing` → `Partial`	partial commit
14	`Partial` → `Compensated`	compensation succeeds
15	`Partial` → `Escalated`	compensation fails or state unknown
16	`Executed` → `Evidenced`	decision and receipt joined

No transition from `Proposed` or `Structured` to `Executing` is permitted. A model proposal cannot skip authority evaluation.

7.13 Brownfield enforcement modes

A mature estate cannot place every integration behind a hard enforcement point at once. The migration path should be explicit rather than hidden behind a binary “governed / ungoverned” label.

Mode	Behaviour	What can be claimed
M0 — Inventory	Action is mapped; no runtime telemetry added	Visibility only
M1 — Observe	Existing identity, model, application and target logs are correlated after the fact	Reconstruction coverage can be measured; no prevention claim
M2 — Shadow decision	PDP evaluates the action but cannot block it; differences from actual execution are recorded	Policy readiness and false-block analysis
M3 — Soft enforcement	Unclear or high-risk actions are reduced to draft, warned or routed to manual processing; some bypass remains	Partial runtime control
M4 — Hard enforcement	Relevant target credentials and network paths exist only behind the PEP; stale or unauthorised actions cannot commit	No-bypass claim for the declared action set

Movement between modes should be per action type and impact tier, not per application. A large ERP may have tier 1 actions in M1 and payment actions in M4 simultaneously.

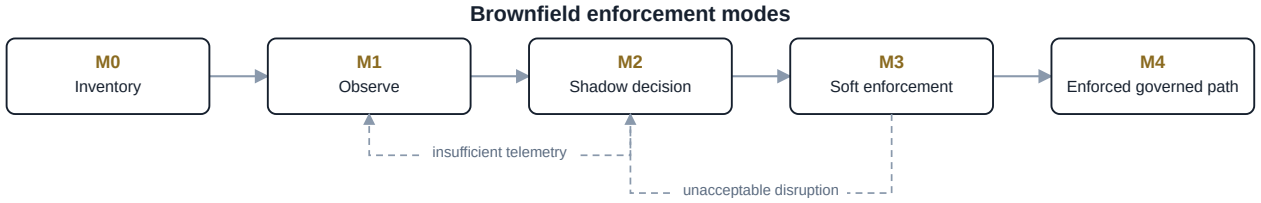

Figure 4. Movement between modes is per action type, not per application.

7.14 Safe-degradation matrix

The response to a control-plane outage must be declared per action tier before production use.

Tier	Default during identity/PDP outage	Default during evidence outage	Default during target-adapter uncertainty
0	Continue with local data/model policy where permitted	Buffer minimal records or mark unevidenced	Return information only
1	Continue or degrade to local recommendation	Buffer and reconcile	No external side effect
2	Draft only; no publication or commit	Store draft and block promotion	Draft only
3	Permit only pre-authorised reversible actions with short offline policy cache; otherwise manual	Queue if reversible; block if evidence is mandatory	Fail closed or compensate
4	Block or route to fully manual authorised process	Block execution	Block and escalate
5	Block; no offline autonomous mode	Block execution	Block, isolate and invoke incident procedure

Offline policy caches create their own freshness and revocation risk. Their maximum age and permitted action set must be part of the policy and evidence profile.

7.15 Break-glass paths

Some organisations need an emergency path that bypasses normal automation controls. A hidden administrator credential is not a break-glass design.

A defensible break-glass path requires:

- a narrowly defined emergency purpose;
- multi-party activation for tier 4 and 5 actions;
- separate credentials and key custody;
- short validity and automatic expiry;
- conspicuous real-time alerting;
- independent evidence that the path was invoked;
- mandatory post-event review;
- regular tests that do not expose the credential to routine operators.

Break-glass use does not conform to the normal transaction chain. It is a declared exception whose purpose is continuity under exceptional conditions, not a permanent bypass disguised as resilience.

8. Worked example: AI-assisted invoice processing

This example is illustrative and constructed. It is not a customer case study, and no claim is made that the figures or failure modes are drawn from a documented incident.

8.1 The scenario

An organisation wants an AI agent that reads invoices from a mailbox, identifies supplier and invoice data, matches against the purchase order in the ERP, assesses discrepancies, prepares a payment, and executes payment for uncritical cases.

8.2 Typical integration without a shared control layer

The agent holds access to the invoice mailbox, read rights on purchase orders, write rights on creditor master data, access to the payment workflow, and a general service account.

The agent logs its model calls. The ERP logs the posting. The identity system logs the service account sign-in.

Questions that remain open:

- Which employee instructed the agent?
- Was that person entitled to authorise payments for this legal entity?
- Was the agent instructed only to check, or to pay?
- Which model version was used?
- Which data were transmitted to the model provider?
- Which policy version applied?
- Was a discrepancy detected?
- Which specific payment did a human approve?
- Does the executed transaction match the transaction presented?
- **Where did the bank details come from?**

8.3 Execution with Governed Intelligence

Step 1 — Intent. An entitled person or a scheduled business process creates the intent: *"Check invoice 4711 against purchase order 9204 and prepare payment."* The intent is structured against a schema; its origin is marked.

Step 2 — Identity. The agent uses its own identity and instance identity. The instructing person remains attached to the transaction as principal.

Step 3 — Mandate. The mandate permits: read the specific invoice; read the associated purchase order; check the supplier; produce a payment proposal. It does not yet permit payment.

Step 4 — Context. The control plane resolves tenant and legal entity, cost centre, invoice amount, data classification, target system and risk indicators.

Step 5 — Policy. For example: amounts up to 500 euro may be processed automatically on a full three-way match; amounts between 500 and 5,000 euro require approval; **new or changed bank details always require a second approval sourced independently of the invoice document**; data from this invoice may only be sent to a model approved for this data class; discrepancies block automatic payment.

Step 6 — Approval. The entitled person is shown invoice number, supplier, bank details, amount, purchase order, detected discrepancies and the planned action — **each field annotated with its provenance**: which values were extracted by the model from the document, which were read from supplier master data, and which were independently verified. Fields derived

solely from the untrusted document are visually distinguished. The approval is bound technically to exactly these values, their provenance, the supplier-master and purchase-order versions used for the decision, and a short expiry.

Step 7 — Execution. The action broker receives a short-lived, transaction-bound right to execute exactly this payment. Immediately before commit it revalidates mandate and policy status and requires the supplier and purchase-order versions to match the approved preconditions. The agent holds no standing payment access.

Step 8 — Evidence. The evidence record links principal, agent, mandate, policy version, model and tools, approval, transaction parameters, parameter provenance, approved and committed precondition state, idempotency key, target receipt and ERP result. Depending on assurance level it is hash-chained, signed and stored in a separate evidence repository.

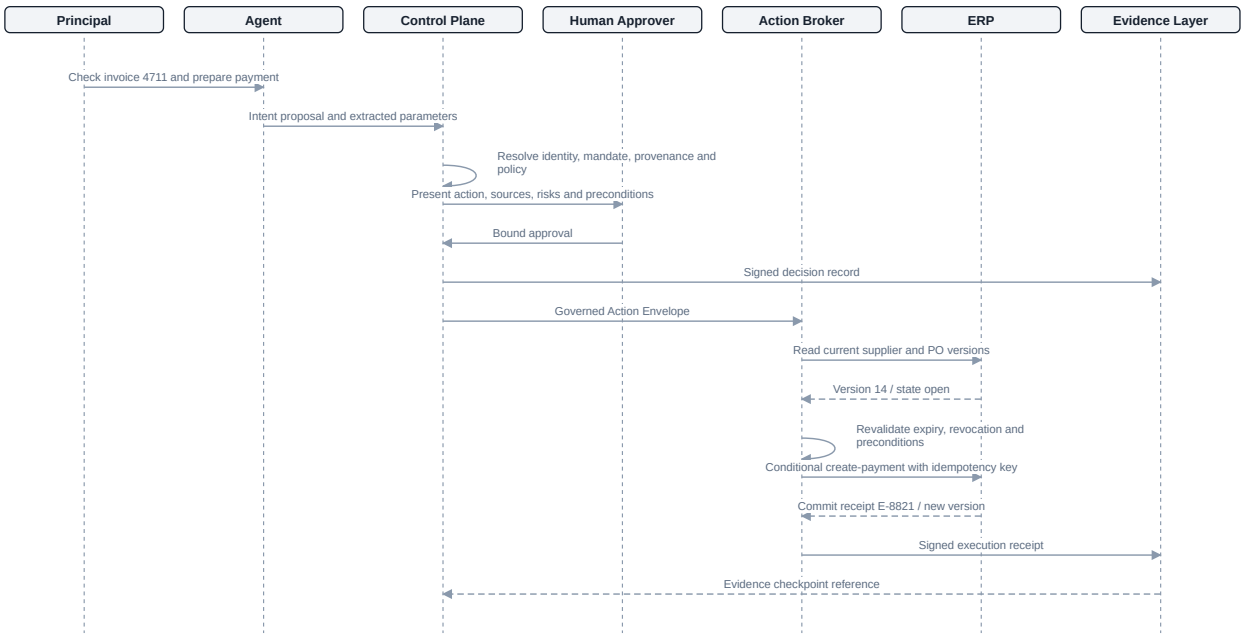

Figure 5. A governed invoice payment, end to end.

8.4 The difference at audit

Without a shared control layer:

> "The service account created a payment at 14:32. The agent log shows a model response a few seconds earlier."

With Governed Intelligence:

> "Agent A, instance A-91, acted on behalf of person B under mandate M-4711. Policy P-27 required approval because of the amount. Person C approved at 14:31 an action bound to amount, payee and invoice. The bank details in that action were extracted by model M v3 from document D (hash H) received from sender S, and were verified against supplier master record V-238 before presentation. The action broker executed exactly this action at 14:32 in the ERP. The ERP confirmed the transaction under reference E-8821. The signed evidence checkpoint shows the event chain has not been altered since."

The second record does not prove that every legal or commercial decision was correct. It makes authority, process, execution and subsequent integrity verification reconstructable — and, critically, it records where each decisive value came from.

8.5 What still fails in this example

Assume the attacker controls the inbound document and the supplier is new, so there is no master record to verify against. The model extracts the attacker's IBAN. Master-data verification returns "no prior record". The approval dialogue shows the IBAN as document-derived and unverified. If the approver approves anyway — a foreseeable outcome under time pressure or poor interface design — the chain executes an attacker-controlled payment with a perfect audit trail.

The controls that would have stopped it are not in the control chain at all: a supplier onboarding process that establishes bank details out of band before the first invoice, and a policy that refuses tier 4 execution where a payee is both new and document-derived. The architecture can *enforce* that policy. It cannot *invent* it.

This is what §3.2 means in a concrete case, and it is why we would rather show the failure than only the success.

8.6 What happens when state changes after approval

Assume the payment is approved at 14:31 against supplier record version 14 and an open purchase order. At 14:31:30 the fraud team blocks the supplier and the ERP advances the record to version 15. The broker attempts execution at 14:32.

A parameter-only design still sees the approved amount and IBAN and proceeds. A precondition-bound design detects the version mismatch. The envelope becomes stale. The broker records `precondition_mismatch`, does not call the payment operation, and returns the transaction for policy re-evaluation or human review.

If the ERP supports a conditional commit, the same version check can be enforced atomically by the target. If it only supports a read followed by an unconditional write, a race remains between the final read and commit. The adapter must disclose that it is operating in `immediate-reread` rather than `conditional-commit` mode, and the organisation should decide whether that assurance is sufficient for the tier.

This case is important because it shows the difference between three proofs:

1. **the approver saw and approved these values;**
2. **those values came from these sources;**
3. **the material state that made the action permissible still held when the target committed it.**

A mature evidence record needs all three.

---

9. An adoption path

Durations are deliberately omitted. Any figure presented without knowledge of identity, target-system and integration maturity would be an estimate disguised as a plan. The sequence below is designed to produce measured scope before schedule commitments.

9.0 Entry principle — no big-bang control-plane migration

The programme unit is an **action type**, not an application and not an AI model. An organisation should not wait until every system is integrated before governing its highest-impact actions. Nor should it route low-impact activity through high-assurance controls that make the platform unusable.

Start with the action map, select the highest combination of impact and feasible control, and move that action through M0 to M4 as defined in §7.13.

9.1 Phase 1 — Make AI actions visible

Inventory the actual AI-assisted actions, not only the models. For each use case, document: business purpose; users and owners; models used; data processed; connected systems; possible actions; impact tier per §2.4; existing approvals; existing logs; third parties.

The result is not a model list but an **AI action map**.

9.2 Phase 2 — Control one bounded process

Choose a clearly bounded pilot: invoice preparation, document approval, support ticket classification, user onboarding, contract drafting, internal publication.

The pilot receives an agent identity, a mandate, bounded tools, a versioned policy, defined approval points, a governed action envelope, an execution receipt and an evidence record.

Choose a pilot at tier 3 or 4. Tier 0–2 pilots often prove integration but not authority binding.

A pilot is not complete when the happy path works. Acceptance criteria should include:

- mandate revoked between approval and execution;
- policy changed after approval;
- target state changed after approval;
- repeated request with the same idempotency key;
- compromised or unavailable evidence producer;
- control-plane outage for every supported tier;
- parameter derived from untrusted input;
- attempted target-system bypass;
- partial commit and failed compensation;
- independent reconstruction by someone outside the implementation team.

9.3 Phase 3 — Reuse shared control functions

Standardise the reusable building blocks: identity schema, mandate format, policy interface, approval mechanism, Governed Action Envelope profile, action broker, execution receipt, evidence schema including parameter provenance and preconditions, tier and provenance definitions, and adapters for central systems.

9.4 Phase 4 — Cross provider and application boundaries

Extend the control layer to further models, applications and business units. Subject-matter responsibility stays decentralised while central minimum controls are enforced uniformly.

Sequence by impact tier, not by system. A tier 4 action in a peripheral system matters more than a tier 1 action in a core one.

9.5 Phase 5 — Continuous recertification

Re-evaluate agents and AI processes regularly or on events: new model version, new tool, changed data class, new regulatory requirement, changed corporate policy, security incident, changed business role, mandate expiry.

9.6 Phase 6 — Independent review and community feedback

For critical components, make architecture, threat model, evidence schema and reference policies available for independent review: open review of this paper, public architecture decision records, community security reviews, external penetration tests, auditor and industry workshops, published corrections and lessons learned.

9.7 Brownfield integration patterns

Different target systems require different control points. Four patterns cover most estates:

Pattern A — Credential choke point

Remove standing target credentials from the agent and business application. Only the broker can obtain a short-lived credential. This is the strongest path where the target supports scoped API access.

Pattern B — Network and API mediation

Place the target API behind a gateway, service mesh policy or network path that requires the governed envelope or a derived proof. This can provide M4 enforcement without changing the business application, but only if alternative network paths and credentials are actually removed.

Pattern C — Target-native workflow binding

Use the target system's own approval and transaction mechanisms, but inject the principal, mandate, provenance and decision references into the native object. This is often preferable for ERP and HR systems with mature workflow controls. The integration task is then cross-system binding and evidence correlation, not replacement.

Pattern D — Observe and reconcile

Where no enforcement point is initially possible, correlate actions after the fact, compute reconstruction coverage and compare a shadow policy decision with the actual outcome. This is M1 or M2, not runtime governance, but it creates the evidence needed to justify the next migration step.

The same programme may use all four patterns. What matters is that the assurance mode is declared per action type.

9.8 Establish a measurement baseline before expansion

Before the pilot changes the process, sample comparable existing transactions and record:

- whether principal, purpose and policy version can be reconstructed;
- time and staff effort to reconstruct;
- number of systems and people needed;
- proportion of approvals bound to the executed values;

- proportion of parameters with reconstructable provenance;
- existing duplicate, stale-state and exception rates;
- current human approval time;
- current operational and audit cost.

Repeat the measurement after the pilot. Without a baseline the organisation can demonstrate architecture but not value.

9.9 Exit criteria for scaled adoption

Scale only when the pilot has shown:

1. no known bypass for the declared high-impact action path;
2. stale approvals are rejected in test and production-like conditions;
3. an independent reviewer can reconstruct a random sample within the target time budget;
4. evidence verifies after key rotation and simulated component compromise;
5. safe degradation and break-glass behaviour are tested;
6. false blocks, approval delay and exception rates are operationally acceptable;
7. data protection and employee-representation controls are agreed where applicable;
8. the measured benefit justifies the additional operating dependency.

---

10. A maturity model for leadership

A numerical score is useful for showing progress, but dangerous if critical failures can be averaged away. Version 0.4 therefore combines a score with mandatory gates.

10.1 How to use this

Score one point per question answered **yes, with evidence a third party could verify**. “We think so” scores zero. Total 0–20.

Then apply the mandatory gates in §10.3. A high numerical score cannot compensate for a direct path around enforcement, self-issued authority or unprotected evidence keys.

10.2 The questions

A. Identity (4 points)

1. Does every production agent have a distinct identity, separate from any shared service account?
2. Can we determine, for a past action, on whose behalf the agent acted?
3. Are individual agent *instances* distinguishable, with parent-child lineage where agents call agents?
4. Can we revoke an agent's authority immediately, and have we tested it during an in-flight process?

B. Authority (4 points)

5. Is the agent's mandate separate from its technical credentials?
6. Are entitlements purpose-bound and time-bound rather than standing?
7. Does the mandate carry a monetary or impact ceiling and an explicit sub-delegation rule?
8. Can a business application or model issue mandates, widen its own scopes or approve its own request? (Score the point if the answer is **no**.)

C. Enforcement (5 points)

9. Can tier 4 or 5 actions reach target systems outside controlled execution points? (Point if **no**.)
10. Do we know which data are transmitted to which model and tool, per data class and transaction?
11. Are policies enforced technically at execution time, not only documented or evaluated in shadow mode?
12. Are human approvals bound to the concrete action, parameters, parameter provenance, material preconditions and expiry?
13. Does the broker reject revoked, replayed, expired or stale envelopes, and is the target operation idempotent or otherwise duplicate-safe?

D. Evidence (4 points)

14. Can we demonstrate which policy version and decision record applied to a past action?
15. Can someone outside the implementation team reconstruct the chain from instruction to committed system state within the declared time budget?
16. Is that chain cryptographically tamper-evident, including an independently verifiable checkpoint for the highest assurance class used?
17. Are keys, signing services, trusted-time services and the evidence store separated from ordinary business applications and service accounts?

E. Organisation (3 points)

18. Do management, IT, data protection, security and the business share one view of responsibility, and are employee representatives involved where the evidence design engages co-determination rights?
19. Are agents, models, tools and action adapters part of ICT risk management, supply-chain assessment and incident response?
20. Do we know which actions must block, degrade or use break-glass when a control component is unavailable, and have we tested each mode?

10.3 Mandatory gates and score caps

Gate	Requirement	Consequence if failed
G1 — Authority separation	Question 8 = yes	Maturity is capped at **Partial**
G2 — No high-impact bypass	Question 9 = yes for the action set claimed as governed	Maturity is capped at **Partial**
G3 — Transaction binding	Questions 12 and 13 = yes	Maturity is capped at **Governed in parts**
G4 — Independent reconstruction	Question 15 = yes	Maturity is capped at **Governed in parts**
G5 — Evidence and key separation	Questions 16 and 17 = yes for tier 4/5 evidence	Maturity is capped at **Governed in parts**
G6 — Safe failure	Question 20 = yes	Maturity is capped at **Governed in parts**

These gates apply only to the action scope being assessed. An organisation may be M4 and “Governed” for invoice payment while remaining M1 and “Partial” for infrastructure changes. A single enterprise-wide label is therefore less informative than an action-level maturity map.

10.4 Score bands

Score	Band	What it means
0–5	**Unmapped**	AI actions exist but are not governed as actions. The first task is an action map, not a platform decision.
6–10	**Partial**	Controls exist per system. They do not compose. Reconstruction is possible but slow, manual or dependent on the implementation team.
11–15	**Governed in parts**	A working control chain exists for some processes. The principal risk is uneven coverage, stale-state handling and ungoverned high-tier actions.
16–20	**Governed**	The chain holds for the declared action scope, all gates pass and the result is independently demonstrable. Residual risks remain, especially manipulated input, privileged-domain compromise and operational dependency.

A score of 20 does not mean safe or compliant. It means the architecture has made its remaining failures more explicit and testable.

10.5 Four measurements that test this paper's thesis

The core thesis is presented as falsifiable. These four measurements test different parts of it:

1. **Reconstruction coverage** — the proportion of AI-initiated transactions for which the human or organisational principal, purpose, policy version, decisive parameter provenance and execution result can be reconstructed from existing records.
2. **Time to reconstruct** — median and p95 elapsed time to reconstruct a complete action chain for a randomly selected transaction, measured by someone who did not build the system.
3. **Approval and state-binding rate** — the proportion of tier 4 and 5 approvals bound to the executed parameters, their provenance and the material preconditions; plus the proportion of stale approvals rejected before commit.
4. **Governance-constrained initiatives** — the proportion of AI initiatives stopped, deferred, reduced in autonomy or materially delayed for reasons recorded as authority, accountability, auditability or control integration rather than model quality, cost or business case.

Measurements 1 to 3 test whether the operational gap exists. Measurement 4 tests whether it is a binding constraint on scaling AI. The gap may be real while some other constraint matters more.

The thesis is weakened if organisations with mature conventional controls score highly on measurements 1 to 3 without a dedicated control-plane pattern, or if they fail to scale while measurement 4 remains near zero. Annex H defines the sampling and reporting method. We invite organisations to publish comparable results, with or without Vianordis.

10.6 Operational measures for a production deployment

The thesis metrics are not sufficient to operate the system. A production service should also track:

Measure	Why it matters
Governed-action coverage by tier	Shows whether the highest-impact actions are actually on the controlled path
Bypass exceptions	Reveals residual privileged routes and temporary waivers
Policy-decision p50/p95/p99	Quantifies machine-path overhead
Human approval p50/p95 and queue age	Shows where business latency accumulates
False-block and false-escalation rate	Detects policy that is technically correct but operationally unusable
Stale-approval rejection rate	Measures real TOCTOU exposure and the value of precondition binding
Break-glass frequency and review closure	Detects whether emergency access has become routine
Evidence verification success	Confirms that signatures, checkpoints, keys and retained validation material still work
Partial-commit and compensation-failure rate	Exposes target adapters that cannot provide the claimed transaction assurance
Cost per governed action	Makes the control architecture an economic decision rather than a compliance abstraction

11. Open source, licensing and institutional stewardship

11.1 Openness does not create compliance

Open source is not a regulatory exemption. Art. 2(12) AI Act excludes AI systems released under free and open-source licences from scope unless they are placed on the market or put into service as high-risk systems or as systems under Art. 5 or Art. 50. The exclusion is conditional, and §6.2.5 sets out three limits that apply to any dual-licensed offering — including this one.

Independently, the GDPR, the BSI Act, DORA, the CRA and sectoral rules continue to apply according to processing and deployment.

11.2 The value of open source is verifiability

For a control layer, open source offers specific advantages: policies and enforcement logic can be inspected; interfaces can be implemented independently; organisations can trace dependencies and data paths; security researchers can report weaknesses; deployment and provider substitution remain possible; evidence formats can develop as open standards; the community can contribute reference policies and adapters; and critical functions need not be accepted as an opaque black box.

Open source does not reduce operational complexity. Verifiability only becomes effective when source, builds, releases, dependencies and governance are traceable together.

11.3 Openness requires a secure supply chain

Trust in a specific release additionally requires: signed commits and releases; traceable build processes; software bills of materials; dependency and vulnerability management; documented review and approval processes; clear maintainer responsibility; reproducible or verifiable builds; coordinated vulnerability disclosure; and protection of release keys and build infrastructure.

The established frameworks here are SLSA, in-toto and Sigstore, with SBOM in CycloneDX or SPDX. From 11 September 2026 the Cyber Resilience Act makes the disclosure element a legal duty for manufacturers, with a 24-hour clock for actively exploited vulnerabilities (§6.6.1).

11.4 The Vianordis licence model, as it stands today

Current licensor. GoSec Cloud OÜ holds the rights and issues all licences.

Open licence. Vianordis code that is released publicly will be licensed under **GNU AGPLv3 (AGPL-3.0-only)**. Commercial use is permitted; use, modification, operation and redistribution remain subject to the licence conditions. To date, no component has been released publicly — see below.

Commercial licence. A separate paid commercial licence provides alternative rights: private modifications, closed hosted operation, OEM use, white-labelling and redistribution. It does not alter AGPLv3 grants already made to other recipients. Indicative annual fee bands are published at vianordis.eu/en/git, ranging from an internal-use tier to bespoke full-exploitation terms.

Current source status — stated plainly. As of Edition 02 / Revision 142 (6 August 2026), **no components have public source code**. The Core is listed as "Source: Not yet public". Applications are to be released progressively after validation gates. A paper arguing for verifiability should say when there is not yet anything to verify; this is that statement.

Contribution terms. The Forge is in testing and not publicly available. **Contribution guidance and the applicable agreements will be published before contributions open.** This is the open item that matters most to prospective contributors, and it deserves more than a placeholder, so:

The choice between a Contributor Licence Agreement with rights transfer, a CLA with a broad licence grant, and a DCO without one is not cosmetic — it determines whether the commercial licence path described above can include community contributions at all. Under Estonian law (Copyright Act, AuÕS), an author's moral rights (*isiklikud õigused*) stay with the author and cannot be assigned; only economic rights (*varalised õigused*) can be transferred or licensed. A CLA drafted on a US template will not map cleanly onto that. We will publish the chosen instrument, the reasoning, and the Estonian-law analysis behind it before opening contributions, rather than presenting contributors with a fait accompli.

The commitment we are willing to be held to: the open reference implementation must be able to demonstrate the central architecture class. The licence model is not intended to become a classic open-core arrangement in which essential security or governance functions remain proprietary. If a future edition of this paper contradicts that sentence, it should be quoted back at us.

What the community needs beyond source code: a public architecture, a documented threat model, open evidence schemas, reference policies, example mandates, standardised interfaces, security and review processes, traceable release decisions, and a transparent contribution process.

11.5 Institutional stewardship

The openness of a critical control architecture does not depend on its software licence alone. It also depends on who decides, over the long term, about source code, brand, licence terms and core architectural principles.

The intended structure. Long-term stewardship is to be vested in a purpose-bound European entity: the **OpenSovereign Alliance Foundation**, a planned Estonian *sihtasutus*, which after legal formation and IP transfer would hold or control the Vianordis source-code rights, administer the trademarks and issue both AGPL and commercial licences.

Where it actually stands. Foundation status is **planned; no cutover has been completed**. GoSec Cloud OÜ remains the responsible operating and licensing organisation. Existing GoSec licences remain valid after any transition, with no duplicate fees. No public-benefit or tax-privileged status is claimed merely from choosing the foundation form.

Safeguards that must exist in the constitutional documents, not only in this paper:

Safeguard	Required effect
Mission lock	The purpose explicitly protects open, sovereign and verifiable digital infrastructure
Open-source lock	Essential security and governance functionality cannot be moved permanently behind a proprietary-only licence by an ordinary board decision
Asset lock	The articles designate mission-compatible recipients on dissolution and exclude a return of remaining assets to the founder or operating company
Independent supervision	Material IP, licence and related-party decisions require independent supervisory involvement
Conflict controls	Interested persons abstain; transactions are documented, valued and approved at arm's length
Continuity	The articles define appointment, temporary vacancy, incapacity and successor mechanisms so that the foundation can operate without founder capture or paralysis
Transparency	Material licensing, governance and related-party decisions are reported in a defined public form, subject to lawful confidentiality

The Estonian legal form does not create all of these safeguards automatically. They must be drafted, implemented and, where relevant, entered in the appropriate register.

The intended transition terms. GoSec Cloud OÜ would remain commercial operator, development partner and corporate contributor. In consideration for transferring the underlying rights, GoSec would receive a perpetual, fully paid-up commercial licence for operation and further development. Existing customer and licence relationships concluded by GoSec would continue until renewal or expiry; new licence relationships would be concluded by the foundation or the licensing structure then defined.

Governance intent. The arrangement is intended to prevent fundamental decisions about open-source availability, licence model, brand or project purpose from being changed unilaterally by a single commercial actor. That requires a protected statement of purpose in the articles (*põhikiri*); independent management and supervisory structures — under the Estonian Foundations Act (SAÕS) these are the *juhatus* and the mandatory *nõukogu*; documented conflict-of-interest rules; transparent decision and publication processes; elevated approval thresholds for changes to mission, the open-source principle or core licensing; and reliable rights for the operating provider without granting it sole control of the project.

Open questions we are not going to pretend are closed. Six items require legal work before the cutover, and a reader assessing the credibility of this structure should ask about all six:

1. **Trademark transfer is a registry act.** Transfer of a mark takes effect against third parties on entry at the Estonian Patent Office (*Patendiamet*) under the Trademarks Act (KaMS). Until that entry, the "brand cannot be captured unilaterally" claim is not yet effective, whatever the intent.
2. **Related-party structure.** An OÜ transferring valuable IP to a foundation it established, in exchange for a perpetual paid-up licence back, is a related-party transaction. It needs an arm's-length valuation, transfer pricing documentation under the Income Tax Act (TuMS), and a VAT analysis under the VAT Act (KMS). Foundation tax treatment in Estonia depends on listing, not on legal form alone, and commercial licensing activity may bear on it.
3. **Personnel separation.** If the same people run the OÜ and sit on the foundation's board or supervisory board, they are on both sides of the transaction. Independent supervisory appointments are what would make the governance claim credible; we intend to make them and have not yet.
4. **Revocation and reversion.** The perpetual GoSec licence should carry defined reversion or termination triggers — purpose failure, change of control at GoSec, insolvency, breach of the open-source principle. Without them, "reliable rights for the operating provider" is difficult to distinguish from a permanent extraction of foundation assets.
5. **Dissolution and asset lock.** The Foundations Act does not by itself guarantee the intended lock in every dissolution scenario. The articles determine distribution of remaining assets, and Estonian law can permit transfer to founders in circumstances where the articles do not provide otherwise. The articles must therefore expressly exclude reversion to the founder or operating company and name one or more mission-compatible fallback recipients, such as an established European open-source foundation. Any additional conditions attached to tax-benefit or public-interest listing require separate analysis.
6. **Contribution and relicensing authority.** The chosen CLA, licence grant or rights-transfer model must match the foundation transition and the commercial licensing path. Contributors need to know which entity receives which rights, whether commercial relicensing is possible, what transparency and remuneration rules apply, and what happens to their grant if the project purpose or steward changes. This cannot remain implicit when contributions open.

A broader open alliance or community initiative can complement this structure: dialogue with industry, research, public bodies and the open-source community; reference standards; support for other European sovereignty projects. Formal responsibility for IP, brand and licensing must remain clearly distinguished from open community participation.

Until the foundation is fully established, staffed and vested, GoSec Cloud OÜ is the responsible operating and licensing organisation. The model described here is the **intended** target architecture of institutional governance, not a claim of a completed transition, and the specific corporate, foundation, tax and licensing arrangements require separate legal review before implementation.

---

12. Conclusion

The next phase of artificial intelligence will not be determined by more capable models alone. It will be determined by how reliably organisations draw the line between **proposal and action**.

An organisation can use an excellent model and still lose control when identities are unclear, mandates are absent, entitlements are too broad, approvals are unbound, policies exist only on paper, business applications can bypass central controls, and logs do not form a shared, tamper-evident chain.

The central question is therefore not *"how autonomous can our AI become?"* but:

> **"What autonomy can we take responsibility for — controllably, revocably and demonstrably?"**

Governed Intelligence describes an architecture in which AI does not act merely because it is technically capable of acting. It acts inside a reviewable authority structure, along the control chain set out in §4.2.

Whether that capability is delivered by a single platform or by several tightly integrated components is an architecture decision, and Chapter 5 sets out why assembling it from existing standards is often the right answer. What is not optional is the end-to-end connection of identity, mandate, context, policy, human authorisation, controlled execution and defensible evidence — with parameter provenance, because without it the evidence answers the wrong question.

Vianordis is our attempt at an open European reference implementation of this architecture class. Annex C states what exists today, which is less than this paper describes. Technical openness is to be complemented by institutional governance that protects the project from unilateral capture; Chapter 11 states how far that has actually progressed, which is not far.

The thesis of this paper is deliberately testable:

> **Enterprises do not fail to scale AI only because of model quality or regulation. They fail because organisational authority and technical execution are not connected.**

§10.5 states four measurements that would confirm, weaken or falsify material parts of it. We do not have them yet. Developing a defensible answer should not happen behind closed doors.

Vianordis therefore invites enterprises, IT leaders, security officers, data protection professionals, lawyers, developers, researchers and the open-source community to examine, criticise and develop both the proposed architecture and the GAIP-0.1 candidate profile — through the contact route below. Substantive critique will be reflected in the next edition with attribution, unless the contributor prefers otherwise.

Annex A — Control capabilities and regulatory objectives

This matrix is orientation, not a conformity assessment. Article references are indicative and, for AI Act high-risk provisions, apply from 2 December 2027 or 2 August 2028 as set out in §6.2.1.

Control capability	EU AI Act	GDPR	NIS2 / BSIG 2025	DORA	Management-system standards
Distinct identities	Provider and deployer responsibilities, Art. 16, 26	Accountability, Art. 5(2); processing under instruction, Art. 28	Access control, § 30(2) no. 9	Clear roles, Art. 5(2)	ISO/IEC 27001 A.5.15–A.5.18; ISO/IEC 42001 cl. 5
Mandates and least privilege	Controlled use and human oversight, Art. 14, 26	Purpose limitation and minimisation, Art. 5(1)(b),(c); by default, Art. 25(2)	Appropriate technical and organisational measures, § 30(1)	ICT risk management and access protection, Art. 9	ISO/IEC 27001 A.5.15; NIST AI RMF GOVERN 1
Automatic logging	Art. 12; retention Art. 19, 26(6)	Accountability and documentation, Art. 5(2), 30	Documentation and evidence duties, § 30(1)	Documented ICT risk framework, Art. 6(1)	ISO/IEC 27001 A.8.15; NIST AI RMF MEASURE 2
Human approval	Art. 14, 26(2)	Safeguards per processing context; Art. 22 where applicable	Management responsibility, § 38(1)	Management body responsibility, Art. 5(2)	ISO/IEC 42001 cl. 8; NIST AI RMF MANAGE 4
Parameter provenance	Supports Art. 12 record quality and Art. 14 interpretability	Supports accuracy, Art. 5(1)(d)	Supports incident analysis, § 32	Supports incident reporting, Art. 19	NIST AI RMF MAP 4
Precondition and freshness binding	Supports reliable controlled use and oversight	Supports accuracy and purpose limitation at processing time	Supports effectiveness and incident containment	Supports operational resilience and controlled change	ISO/IEC 27001 change and access controls
Provider abstraction	Roles along the AI value chain, Art. 25	Processors and third-country transfers, Art. 28, 44 ff.	Supply chain security, § 30(2) no. 4	ICT third-party risk, Art. 28	ISO/IEC 27001 A.5.19–A.5.22
Evidence record	Traceability and cooperation with authorities, Art. 21, 26	Demonstrability, Art. 5(2); DPIA, Art. 35	Documentation, audit, incident handling, §§ 30, 32	Governance, reporting and audit, Art. 5, 6	ISO/IEC 27001 A.5.28; ISO/IEC 42001 cl. 9
Cryptographic tamper-evidence	Supports integrity of technical evidence	Supports accountability; constrained by minimisation and erasure	Cryptography, § 30(2) no. 8	Supports controllable audit trails	ISO/IEC 27001 A.8.24
Trusted time and key lifecycle	Supports reliable timing and retained verification of logs	Supports demonstrability; retention must remain proportionate	Supports incident timelines and cryptographic governance	Supports auditability and resilience	RFC 3161 / RFC 4998; ISO/IEC 27001 cryptographic controls
Lifecycle and revocation	Monitoring, correction and oversight, Art. 26(5), 72	Storage limitation, Art. 5(1)(e); change of risk, Art. 35(11)	Effectiveness review, § 30(2) no. 6	Continuous monitoring and improvement, Art. 6(5)	ISO/IEC 42001 cl. 10
Model and provider substitution	Avoids unclear role and integration dependencies, Art. 25	Controllable data paths	Supply chain and dependency management, § 30(2) no. 4	Exit strategies and concentration risk, Art. 28(8)	— (Data Act Art. 23–31 also applies)
Core isolation	Supports controlled technical use	Supports privacy by design, Art. 25(1)	Segmentation, access control, secure administration, § 30(2) nos. 5, 9	Supports ICT risk containment and resilience, Art. 9	ISO/IEC 27001 A.8.20–A.8.23

Control capability	EU AI Act	GDPR	NIS2 / BSIG 2025	DORA	Management-system standards
Vulnerability handling	—	Art. 32, 33 where personal data affected	§ 32 reporting	Art. 17–19 incident management	CRA Art. 14; ISO/IEC 27001 A.8.8

Annex B — Trust boundaries and assumed compromises

A defensible architecture must state explicitly what it trusts and which compromises it aims to bound.

B.1 Central trust boundaries

#	Boundary	Scope
1	**Human identity boundary**	Authentication and authorisation of natural persons and organisational roles
2	**Agent identity boundary**	Registration, attestation and lifecycle of individual agents and instances
3	**Mandate authority boundary**	Issuance, validation and revocation of mandates. Business applications must not assume this authority
4	**Policy boundary**	Versioned decisions on permitted actions. Models must not alter or bypass policy
5	**Execution boundary**	Action broker and tool gateways as the controlled path to target systems
6	**Key boundary**	Key management and signing services outside business applications and ordinary service accounts
7	**Evidence boundary**	Separate, append-only or comparably protected evidence path with independent checkpoints
8	**Tenant boundary**	Separation of identities, policies, key material, data, evidence and network paths between tenants
9	**Input trust boundary**	**All content the agent reads — documents, mail, web pages, tool descriptions, retrieved records — is untrusted input. Anything derived from it is a claim, not a fact**
10	**Target-state boundary**	Material target-system state is revalidated at commit; approval-time state is not assumed to remain true
11	**Time and cryptographic boundary**	Clock sources, timestamp services, signing keys, validation material and renewal processes are protected separately from business applications

Boundaries 9 to 11 are the additions that move v0.4 from parameter binding to transaction assurance: untrusted input, mutable target state, and the lifecycle of time and cryptographic proof.

B.2 Compromise scenarios to be bounded

Compromised component	Expected bound
Single business application	Cannot issue its own mandates; no direct privileged target-system access; bounded scope
Single agent	No self-escalation; revocable; only mandated actions
AI model or provider	No technical authority; bounded data context; substitutable
Service account	Short lifetime and bounded scopes; no access to central signing keys
Evidence producer	Independent chaining, signature or witness makes manipulation detectable
Tenant administrator	No cross-tenant rights; critical actions optionally under multi-party approval
Platform administrator	Key separation, four-eyes controls, independent evidence, constrained break-glass paths
Build or release pipeline	Signed releases, SBOM, separated approval, reproducible or verifiable builds
Input data and tool descriptions (indirect prompt injection, tool poisoning)	**Not prevented. The model may derive falsified parameters and a correctly bound approval will bind to them. Bounds: parameter provenance recorded in the evidence record; out-of-band verification of impact-determining fields; provenance surfaced in the approval dialogue; change thresholds on master data; tier ceilings for wholly untrusted-input-derived actions (§3.2)**
Target state changes after approval	Approval becomes stale; broker revalidates versions, policy and revocation; conditional commit where supported; lower assurance declared otherwise
Replay or duplicate delivery	Transaction ID and idempotency key prevent a second commit; repeated requests return the existing outcome or fail closed

Compromised component	Expected bound
Clock manipulation	Monitored clock source, skew bounds, signed checkpoint timestamps and independent timestamping for high assurance
Evidence signing-key compromise	Key separation, rotation, revocation, compromise timestamp, historical validation material and re-sealing of unaffected evidence
Provider model alias changes	Record provider request ID, deployment, API version, alias and reproducibility status; do not claim an unavailable exact version
Break-glass credential misuse	Multi-party activation, short expiry, real-time alerting, separate evidence and mandatory review

B.3 Limits of isolation

No microservice or zero-trust architecture removes all risk. Legitimate but overly broad data access can still be abused. An organisation with full control of all administrative keys can destroy technical evidence unless independent checkpoints or external witnesses exist. And, as B.2 states, an agent acting with valid authority on false premises is outside the model entirely.

The architecture must therefore document its security assumptions openly and must not state protective effects in absolute terms.

B.4 Claims this architecture must not make

The following statements exceed what the controls prove:

Prohibited shorthand	Defensible statement
“Immutable audit log”	Subsequent alteration, gaps or reordering are detectable under the stated trust assumptions
“The approval proves the payment was correct”	The approval proves that the approver accepted the displayed action and provenance under the recorded preconditions
“Signed means trusted”	The object verifies under a particular key and validation profile; key trust and compromise status still matter
“The AI was authorised”	A specific agent instance was authorised for a bounded action on behalf of a principal
“EU hosted means sovereign”	The deployment satisfies the separately stated controls over identities, keys, data paths, providers and exit
“Open source means compliant”	Source can be inspected; compliance depends on system, deployment, organisation and use
“No bypass”	No bypass was found for the declared action set and tested trust boundaries; the claim has a scope and date

These distinctions are not semantic caution. They determine what an auditor, customer or incident investigator can rely on.

Annex C — Vianordis as a reference implementation

> **Implementation status, 14 August 2026.** This annex describes both the target architecture and what actually exists. Where they differ, the difference is stated. Platform status is drawn from the public release status page, Edition 02 / Revision 142 (6 August 2026).

C.1 What Vianordis is

Vianordis is a European SaaS product designed to govern how people, AI agents and digital systems act, decide and produce verifiable evidence. It is a **product, not a legal entity**; GoSec Cloud OÜ develops, commercialises and operates it and holds customer contracts.

The applications are intended to share a common Core rather than each building isolated identity, policy and audit worlds. The Core connects identities, authority and mandates, policies, agents and models, human approvals, controlled actions, logging and evidence, lifecycle and monitoring.

The public product description presents the Core as five connected controls — **identity, authority, policy, action, evidence**. This paper's eight-element chain (§4.2) is the same model at finer granularity: Intent and Context are inputs to the authority and policy decision, and Mandate and Approval are the artefacts that make authority and human oversight explicit. The two vocabularies should be reconciled in the product documentation; at present they are not, and that is a documentation defect rather than an architectural one.

C.2 What Vianordis is not

Not a language model. Not a mandatory proprietary model. Not a thin AI wrapper. Not GRC document management. Not a replacement for IAM, SIEM or business applications. Not an automatic compliance guarantee.

And, per Chapter 5, not a novel invention. It composes established components — policy engines, workload identity, transparency-log constructions, supply-chain attestation — and adds the mandate model, provenance-aware approval binding and unified evidence schema described in §5.2.

C.3 Bring your own AI

Organisations should be able to connect public models, European AI services, private cloud models, self-hosted models, specialised domain models and local on-device models.

The control decision must not be delegated wholly to the model provider. The control plane decides which model is admissible for a transaction, which data may be provided, which region or operating form is required, which tools the agent may use, which approval is necessary, and which evidence must be produced.

C.4 Applications as instruments inside a shared security boundary

The 25 Vianordis applications are specialised instruments on a common platform. They are **not intended to** access connected enterprise systems directly with standing privileged credentials.

Identity, mandate validation, policy decision, approval, controlled execution and evidence generation **are to be** provided by separate Core services inside their own security and authority boundary. Business applications **are to** authenticate to those services with distinct workload identities over mutually authenticated connections. Entitlements **are to be** granted on least-privilege principles and bounded in time, scope and transaction as far as practicable.

The Core is therefore intended as an independent trust boundary with separate technical identities, isolated key custody and defined enforcement interfaces — not a shared software library. Critical functions — mandate issuance, policy decision, signing, key management and evidence checkpointing — **are to be** separated so that compromise of one component does not

confer all control functions.

Verification status: these are design commitments, not verified properties. The public release status lists "external installation, upgrade, recovery, compatibility and security review" as the next gate for the Core. No independent security assessment has been completed or published.

This isolation is not an absolute guarantee in any case. A compromised application can still generate incorrect requests or misuse data it was lawfully permitted to access. The objective is to bound blast radius, make bypass attempts visible and force additional authority checks.

C.5 The 25 applications and their status

The four tables below list **25 applications**, all at **Controlled Beta**: access by invitation, validation support only, **no production SLA**, source not yet public. Two sector solutions are at **Alpha** and are listed separately after them; they are not included in the count of 25.

Work suite

Application	Function	Status
Workplace	Unified mail, calendar, contacts, files, tasks, archive and AI assistant	Controlled Beta
Aevis	Protected email evidence for search, retention and eDiscovery	Controlled Beta
Bicameral	Governed assistant chat and approval workflows	Controlled Beta
Korum	Sovereign video meetings with SSO, recordings and SIP/VTC interoperability	Controlled Beta
Viaport	Zero-knowledge encrypted file transfer	Controlled Beta
Stratis	Secure file storage and management	Controlled Beta
Voxia	Email marketing and automation	Controlled Beta
Kognis	Knowledge base and documentation	Controlled Beta

Business

Application	Function	Status
Nodus	Customer relationship management	Controlled Beta
Kinetis	Project and task management	Controlled Beta
Taktis	Time tracking and billing	Controlled Beta
Zelibri	HR and people platform	Controlled Beta
Enklave	Device and endpoint management	Controlled Beta
Orbitas	Workflow automation with human control	Controlled Beta
Synkron	Event management and ticketing	Controlled Beta

Industry

Application	Function	Status
Paideia	Corporate learning, academies and skills development	Controlled Beta
Lumina	Video streaming and creator platform	Controlled Beta
Strukta	Sovereign hybrid headless CMS for regulated content teams	Controlled Beta
Spectra	Document intelligence and fast search	Controlled Beta
Sonoris	Sovereign audio platform for music, podcasts, rooms and live streams	Controlled Beta

Application	Function	Status
Visoris	Video surveillance and camera management	Controlled Beta

Platform

Application	Function	Status
Skills	Governed skills marketplace for AI agents and customer-built extensions	Controlled Beta
O365 Migrator	Controlled Microsoft 365 migration to sovereign European infrastructure	Controlled Beta
Inference	Dedicated vLLM inference on European GPUs	Controlled Beta
Velum	Tenant encryption plane for BYOK, CYOK and HYOK custody	Controlled Beta

Sector solutions — Alpha, subject to organisational qualification (additional to the 25 above):

Solution	Domain	Status
Omniscola	Education	Alpha
Aether	Healthcare	Alpha

Note: the public site currently spells these inconsistently across pages ("Omniscolaa" / "Omniscola", "Aethera" / "Aether"). The release-status page spelling is used here; the product documentation should be reconciled.

> **A note on the portfolio.** Twenty-five applications plus a control plane is a large surface for an organisation at Controlled Beta. Two questions follow, and readers are entitled to ask both: whether the Core can be a credible neutral control layer while the same organisation supplies the business applications above it, and whether the portfolio can be secured to the standard the Core's premise demands. On the first: the Core's enforcement interfaces are intended to be the same for first-party and third-party applications, and the value of that claim depends entirely on the source being published so it can be checked. On the second: the answer will be visible in whether independent security review happens before general availability, not in this paragraph.

C.6 European operation and trust posture — stated without embellishment

Vianordis is designed for controllable operation in the European Union: EU-based operating options, controllable data paths, tenant separation, own key and identity control, model and provider portability, open interfaces, documented dependencies, exit and migration capability. "EU operation" is understood as a combination of legal, technical and operational control capability, not merely a hosting location.

What is **not** currently claimed, per the public trust centre:

Item	Status
ISO/IEC 27001	Controls mapping only. **No public certificate. No certification or completed audit claimed.**
SOC 2	Not claimed
Sub-processors	Contract-specific; processor schedule supplied with the contract. **No universal EU-only chain claimed.**
Data residency	Managed-service validation locations disclosed per invitation. **No universal production region, replication, RPO, RTO or failover claim.** The public website currently runs on OVHcloud VPS in France.
Key custody	BYOK, CYOK or HYOK by tenant and data domain; architecture scope supplied during review. HYOK metadata, proxy, backup and failure boundaries remain to be documented.
Incident history	Publication pending

Item	Status
CRA applicability and readiness (§6.6.1)	Scope assessment in progress. Pure SaaS operation is generally outside the CRA; installable components and remote data processing solutions integral to them are not. Coordinated disclosure process and reporting readiness to be published before any such component is distributed.
Source code	Not yet public for any component
Foundation	Planned; no cutover completed

C.7 Positioning

Vianordis does not claim that only one platform can meet the requirements described here. Organisations can assemble the necessary capabilities from several tightly integrated components, and Chapter 5 explains when that is the better choice.

Vianordis positions itself as an open European reference implementation of this architecture class: a shared authority and policy core, model-independent AI integration, controlled tool and system actions, bound human approvals with parameter provenance, cryptographically verifiable evidence, and institutionally protected open-source stewardship. **None of these is independently verified today**; C.6 states what is and is not claimed.

The claim is not that regulation requires a particular product. The claim is that a real architectural gap exists, and that a verifiable implementation of it should be available to inspect.

C.8 Evidence required before a general-availability claim

The target architecture in this paper should not become a product claim merely because code exists. Before Vianordis describes the Core as generally available for consequential actions, the following evidence should be public or available under review terms:

Capability	Evidence required	Status at v0.4
Core as an independent security boundary	Architecture and trust-boundary review; deployment diagrams; key and network separation evidence	Not published
Mandate issuance and revocation	Schema, conformance tests, in-flight revocation test results	Not published
No-bypass enforcement	Credential and network-path review for declared action types; negative tests	Not published
Action and precondition binding	GAIP test vectors; stale-state and replay tests; adapter assurance modes	Candidate profile only
Evidence integrity	Canonicalisation/signature profile; key lifecycle; checkpoint verification tests	Not published
Tenant isolation	Independent penetration test and cross-tenant negative tests	Not published
Control-plane resilience	Tested degradation matrix, backup, recovery, RPO/RTO and break-glass exercise	Not published
Software supply chain	Signed releases, SBOM, build provenance and coordinated disclosure process	Not public because source is not yet released
Privacy and employee safeguards	DPIA support material, evidence-role model, retention classes and works-agreement guidance	Not published
Operational performance	p50/p95/p99 machine path, approval latency, false-block rate and cost per governed action	No field data published

This table is a release gate, not a promise that every item will be public in unrestricted detail. Sensitive penetration-test evidence may need controlled disclosure. The important point is that the assurance claim must be backed by reviewable evidence and a scope date.

Annex D — Glossary

Action Broker — Controlled technical intermediary that executes approved actions in a target system with bounded entitlements.

Agent — Software unit that pursues goals, prepares decisions, selects tools or executes actions. An agent may use one or more AI models.

Agent Identity — Distinct technical identity of an agent or of a specific agent instance.

AI action map — An inventory of the AI-assisted *actions* an organisation performs — purpose, owner, models, data, connected systems, possible actions, impact tier, approvals, logs and third parties — as distinct from an inventory of AI models (§9.1).

Anbieter → see Provider.

Approval — Human or organisational release of a defined action.

Assurance level — Graded evidential strength applied to a transaction (§7.8.3).

Betreiber → see Deployer.

Blast radius — The extent of systems, data and actions reachable by a single compromised component.

Break-glass — A pre-authorised emergency path that bypasses normal controls, subject to heightened logging and after-the-fact review.

BYOK / CYOK / HYOK — Bring, control or hold your own key: three custody models for tenant encryption keys, differing in whether the key is supplied to, controlled within, or held entirely outside the service provider's boundary.

Control chain — The eight linked elements of §4.2.

Control Plane — Shared control layer for identities, mandates, policies, approvals, execution and evidence.

Cryptographic tamper-evidence — Mechanisms by which subsequent alteration, gaps or reordering in an evidence chain become detectable. It does not mean manipulation is technically impossible in every scenario.

Deployer (German: Betreiber) — Under the AI Act, in principle a natural or legal person, public authority or other body using an AI system under its authority, excluding personal non-professional use. The actual legal role depends on the case.

Digest — A cryptographic hash used as a compact, verifiable reference to a specific data state.

Forge — The Vianordis source, issue and contribution platform. In testing and not publicly available as at this edition (§11.4).

Evidence — Correlated proof of instruction, authority, decision, approval, execution and result.

Governed Intelligence — AI-assisted processing and execution within an enforceable, reviewable and revocable control chain.

Impact tier — Classification of an action by its effect, tiers 0 to 5 (§2.4). Not the AI Act's legal risk classification.

Indirect prompt injection — Manipulation of a model's behaviour or extracted parameters through content in data the model reads, rather than through the user's instruction (§3.2).

Intent — Structured description of the intended business action.

Mandate — Bounded authorisation for an agent to act on behalf of a principal within a defined period and context.

Parameter provenance — The recorded origin of each value in an action: model-derived from a specific untrusted document, read from a system of record, entered by a user, or independently verified.

Policy Decision Point (PDP) — Component that decides, from rules, whether and under what conditions an action is permitted. Term from XACML and NIST SP 800-207.

Policy Enforcement Point (PEP) — Component that technically enforces a policy decision.

Policy drift — Divergence between the policy under which a mandate or agent was approved and the policy currently in force.

Principal — Person, role or organisation on whose behalf an agent acts.

Provider (German: Anbieter) — Under the AI Act, in principle the body that develops or has developed an AI system or general-purpose AI model and places it on the market or puts it into service under its own name or trademark. Substantial modification or a change of intended purpose can reallocate the role (Art. 25).

Runtime governance — Enforcement of governance rules during actual processing or execution, not only through upstream documentation or downstream review.

Shadow AI — Use of unapproved or insufficiently recorded AI tools.

Shadow Execution — An AI-assisted action whose principal, purpose, governing policy version and approval binding cannot be reconstructed from existing records within a defined time budget (§2.2).

Sihtasutus — Estonian foundation; a purpose-bound legal person without members, governed by the Foundations Act (SAÕS), with a management board (*juhatus*) and a mandatory supervisory board (*nõukogu*).

Trust boundary — Technical and organisational boundary at which trust is explicitly verified, reduced or re-established.

Witness — An independent service that observes and countersigns checkpoints of an evidence log, so that a single operator cannot rewrite history undetected.

WORM — Write-once, read-many storage, used to make deletion or alteration of stored evidence infeasible for the operator.

Small mid-cap — Category introduced by the Digital Omnibus for enterprises below defined employee, turnover and balance-sheet thresholds, receiving simplified documentation duties and capped penalties under the AI Act.

Workload identity — Cryptographic identity issued to a running software workload rather than to a human, e.g. via SPIFFE/SPIRE. In SPIFFE, the identity document is an SVID.

Action commitment — Digest or equivalent binding of the operation and parameter set presented for approval.

Canonicalisation — Conversion of a structured object into a deterministic representation so that independent parties hash or sign the same bytes.

Conditional commit — Target operation that succeeds only if declared state or version preconditions still hold.

Cryptographic agility — Ability to change keys, algorithms and profiles without losing the capacity to verify retained evidence.

Execution receipt — Signed or otherwise verifiable record emitted by the broker and, where possible, the target system, identifying what committed, under which transaction and state version.

Governed Action Envelope — Short-lived signed artefact carrying the authorised action, authority references, parameter provenance, approval, preconditions and execution constraints from decision to enforcement.

GAIP-0.1 — Candidate Governed Action Interchange Profile defined in Annex G.

Idempotency key — Unique key used to ensure that retries of the same governed action do not create duplicate side effects.

Precondition commitment — Digest or version set representing the material target state under which an action was approved.

Provenance assurance class — P0 to P4 classification of how an impact-determining parameter was sourced and verified (§2.5).

Reproducibility status — Evidence-field classification stating whether a model execution can be tied to a reproducible artefact, only to a provider deployment or request, only to a mutable alias, or to no reliable version identifier.

Stale approval — An approval whose expiry, policy, mandate, revocation status, action parameters or material execution preconditions no longer match at commit time.

TOCTOU — Time-of-check/time-of-use: the risk that facts or state validated at decision time change before execution.

Trusted time — Time assertion whose source, accuracy, integrity and trust model are stated and suitable for the assurance claim; may include RFC 3161 timestamping.

Annex E — Primary sources

Official texts govern. Secondary sources are marked as such.

EU AI Act and Digital Omnibus

1. Regulation (EU) 2024/1689 (Artificial Intelligence Act), consolidated text: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02024R1689-20260727
2. Regulation (EU) 2026/1744 of 8 July 2026 amending Regulations (EU) 2024/1689, (EU) 2018/1139 and (EU) 2023/1230 as regards the simplification of the implementation of harmonised rules on artificial intelligence (Digital Omnibus on AI). OJ 24 July 2026; in force 27 July 2026: https://eur-lex.europa.eu/eli/reg/2026/1744/oj
3. European Commission, AI Act Service Desk — implementation timeline: https://ai-act-service-desk.ec.europa.eu/en/ai-act/timeline/timeline-implementation-eu-ai-act
4. AI Act Service Desk, Articles 2, 12, 14, 19, 25, 26, 50, 99: https://ai-act-service-desk.ec.europa.eu/en/ai-act/article-99

DORA

5. Regulation (EU) 2022/2554 (Digital Operational Resilience Act): https://eur-lex.europa.eu/eli/reg/2022/2554/oj
6. BaFin, DORA overview: https://www.bafin.de/DE/unternehmen-maerkte/aufsicht/alle-unternehmen/dora/ueberblick/ueberblick_node.html

NIS2 and German BSI Act

7. NIS2UmsuCG of 2 December 2025, BGBl. 2025 I no. 301: https://www.recht.bund.de/bgbl/1/2025/301/VO
8. BSIG 2025 § 30 (risk management measures): https://www.gesetze-im-internet.de/bsig_2025/__30.html
9. BSIG 2025 § 38 (implementation, supervision, training and liability of management bodies): https://www.gesetze-im-internet.de/bsig_2025/__38.html
10. BSI, NIS-2 obligations and registration duty: https://www.bsi.bund.de/DE/Themen/Regulierte-Wirtschaft/NIS-2-regulierte-Unternehmen/NIS-2-Pflichten/nis-2-pflichten_node.html
11. Draft CRA implementation act, BT-Drs. 21/6134 of 26 May 2026: https://dserver.bundestag.de/btd/21/061/2106134.pdf

GDPR and further instruments

12. Regulation (EU) 2016/679 (GDPR): https://eur-lex.europa.eu/eli/reg/2016/679/oj
13. Regulation (EU) 2024/2847 (Cyber Resilience Act): https://eur-lex.europa.eu/eli/reg/2024/2847/oj
14. Regulation (EU) 2023/2854 (Data Act): https://eur-lex.europa.eu/eli/reg/2023/2854/oj
15. Regulation (EU) 2024/1183 (eIDAS 2.0): https://eur-lex.europa.eu/eli/reg/2024/1183/oj

Technical standards and prior art referenced in Chapter 5

16. OASIS, eXtensible Access Control Markup Language (XACML) v3.0
17. NIST SP 800-207, Zero Trust Architecture
18. RFC 8693, OAuth 2.0 Token Exchange
19. RFC 6962, Certificate Transparency
20. SPIFFE / SPIRE specifications: https://spiffe.io
21. Open Policy Agent: https://www.openpolicyagent.org · AWS Cedar: https://www.cedarpolicy.com
22. SLSA: https://slsa.dev · in-toto: https://in-toto.io · Sigstore: https://www.sigstore.dev
23. ISO/IEC 42001:2023, AI management systems · ISO/IEC 27001:2022, information security management
24. NIST AI Risk Management Framework 1.0
25. Model Context Protocol specification: https://modelcontextprotocol.io

26. Hardy, N. (1988). *The Confused Deputy*. ACM SIGOPS Operating Systems Review 22(4)
27. Birgisson, A. et al. (2014). *Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud*. NDSS
28. RFC 8785, JSON Canonicalization Scheme (JCS): https://www.rfc-editor.org/info/rfc8785
29. RFC 7515, JSON Web Signature (JWS): https://www.rfc-editor.org/info/rfc7515
30. RFC 9052 and RFC 9053, CBOR Object Signing and Encryption (COSE): https://www.rfc-editor.org/info/rfc9052 · https://www.rfc-editor.org/info/rfc9053
31. RFC 3161 and RFC 5816, Time-Stamp Protocol and update: https://www.rfc-editor.org/info/rfc3161 · https://www.rfc-editor.org/info/rfc5816
32. RFC 4998 and RFC 9169, Evidence Record Syntax and updated ASN.1 modules: https://www.rfc-editor.org/info/rfc4998 · https://www.rfc-editor.org/info/rfc9169
33. RFC 9110, HTTP Semantics — conditional requests and entity tags: https://www.rfc-editor.org/info/rfc9110

Vianordis product and status sources

34. Vianordis platform overview: https://www.vianordis.eu
35. Applications and status: https://www.vianordis.eu/en/services
36. Release status, Edition 02 / Revision 142: https://www.vianordis.eu/en/release-status
37. Source and licensing: https://www.vianordis.eu/en/git
38. Trust centre: https://www.vianordis.eu/en/trust
39. Company: https://www.vianordis.eu/en/company

German provisions referenced in the text

40. § 87(1) no. 6 Betriebsverfassungsgesetz (Works Constitution Act) — co-determination on technical monitoring devices: https://www.gesetze-im-internet.de/betrvg/__87.html
41. § 43 GmbHG and § 93 AktG — directors' duty of care and liability standards referenced by § 38(2) BSIG: https://www.gesetze-im-internet.de/gmbhg/__43.html · https://www.gesetze-im-internet.de/aktg/__93.html

Pending legislation (not current law)

42. European Commission, Digital Omnibus proposal on data, privacy and ePrivacy, 19 November 2025 — in the legislative process as at August 2026: https://www.europarl.europa.eu/legislative-train/theme-a-new-plan-for-europe-s-sustainable-prosperity-and-competitiveness/file-digital-package

Estonian law referenced in Chapter 11

43. Foundations Act (*Sihtasutuste seadus*, SAÕS), official English consolidated text: https://www.riigiteataja.ee/en/eli/510042014001/consolide · Copyright Act (*Autoriõiguse seadus*, AuÕS), Trademarks Act (*Kaubamärgiseadus*, KaMS), Law of Obligations Act (*Võlaõigusseadus*, VÕS), Income Tax Act (*Tulumaksuseadus*, TuMS), Value Added Tax Act (*Käibemaksuseadus*, KMS) — current consolidated texts: https://www.riigiteataja.ee

Annex F — Version history and what changed

Version	Date	Language	Principal changes	Author
0.1	2026 (exact date to be restored)	DE	Initial draft	Y. F. N'Soussoula
0.2	2026-08-14	DE	Institutional stewardship; Core as security boundary; cryptographic tamper-evidence; sharpened positioning; extended trust-boundary analysis; updated regulatory sources including Regulation (EU) 2026/1744	Y. F. N'Soussoula
0.3	2026-08-14	EN	Prior art, parameter provenance, input manipulation, implementation transparency, maturity model and corrected regulatory timelines	Y. F. N'Soussoula
0.4	**2026-08-14**	**EN**	**Architecture Profile Edition: Governed Action Envelope; precondition and TOCTOU binding; trusted time and key lifecycle; brownfield modes; mandatory maturity gates; corrected foundation asset-lock analysis; GAIP-0.1 and measurement protocol**	**Y. F. N'Soussoula**
0.5	planned	EN + DE	First measured pilot results; independent technical and Estonian legal review; aligned German executive edition; published test vectors and reference implementation status	—

What changed in v0.3, and why

This edition was revised following a structured review of v0.2. The substantive changes:

Argument

1. **Shadow Execution redefined independently of any solution** (§2.2). The v0.2 definition described the absence of the vendor's own control chain, which made the thesis circular. The new definition is stated in terms an auditor can evaluate.
2. **The novelty claim withdrawn and replaced** (§2.2). Shadow Execution is identified as the confused-deputy and delegation problem at agent scale, not a newly discovered risk class. Shadow AI and Shadow Execution are stated as overlapping rather than sequential.
3. **New Chapter 5, "Relationship to prior art".** XACML, NIST SP 800-207, RFC 8693, macaroons, SPIFFE/SPIRE, OPA/Cedar, Certificate Transparency, SLSA/in-toto/Sigstore, ISO/IEC 42001, NIST AI RMF and MCP are named, credited, and mapped against the capabilities described here. §5.2 states the four things actually claimed as contributions; §5.3 tells a reader when to assemble rather than buy.
4. **New §3.2 on input manipulation.** Indirect prompt injection, tool poisoning and related failures were absent from v0.2 entirely. §3.2 states that a correctly bound approval binds to falsified parameters, works the failure through the paper's own example (§8.5), and derives the parameter-provenance requirement that now runs through §4.3.6, §7.8, Annex B.2 and the maturity model.
5. **Existing controls described at their real strength** (§1.3). The v0.2 comparison table understated IAM, API gateways and ERP workflow approval, which made the argument easy to dismiss. The table now states what each mechanism genuinely covers.
6. **Evidence status stated openly.** The paper carries no empirical field data, and now says so before Chapter 1 rather than implying otherwise.
7. **Falsification criteria added** (§10.4). Four measurements are specified that would confirm or weaken the core thesis — three testing whether the control gap exists, and one testing whether it is actually what stops AI adoption, which is the part of the thesis most likely to be wrong.

Regulation

8. **AI Act timeline as a table** (§6.2.1), replacing a dense paragraph repeated in two places.
9. **Application dates flagged on the high-risk articles** (§6.2.2). Articles 12, 14, 19, 25 and 26 are not yet in force; v0.2 presented them as current duties.

10. **Article 14 quoted in the official wording** (§7.6). The v0.2 paraphrase diverged from the enacted text.
11. **Penalty bands completed** (§6.2.4): Art. 99(5), the SME cap in Art. 99(6), Art. 100, Art. 101, and "whichever is higher".
12. **Digital Omnibus substantive changes added** (§6.2.3): Art. 4 weakened, "safety component" narrowed, new Art. 5 prohibitions from 2 December 2026, new Art. 4a, Art. 75 enforcement shift, registration duty retained, small mid-cap category.
13. **§ 38(2) BSIG added** (§6.3) — personal liability of management bodies, the sharpest currently-applicable obligation for this paper's primary audience, absent from v0.2.
14. **BSIG staged deadlines added** (§6.3), including the 24 h / 72 h / one month reporting clock.
15. **GDPR Art. 25(2), data protection by default, added** (§6.5) — the more directly applicable provision for agent architectures.
16. **Cyber Resilience Act and Data Act added** (§6.6), with its scope stated precisely — pure SaaS is generally outside the CRA and covered by NIS2 instead — and with the consequences for our own distributable components acknowledged. eIDAS 2.0 flagged as one to watch.
17. **DORA precision** (§6.4): "ultimate responsibility" under Art. 5(2)(a), the Art. 6(4) independent control function, and the Art. 16 simplified regime for smaller entities.
18. **Clarification that the data and privacy Digital Omnibus is not law** (§6.5), to counter a common market confusion.

Credibility and status

19. **All 25 applications named with status** (Annex C.5). v0.2 asserted the number three times without ever listing them.
20. **Implementation status stated throughout.** Controlled Beta, by invitation, no production SLA, no public source, no completed certification, no universal EU-only sub-processor chain, foundation not yet established. Chapter 11 and Annex C.6 state this in the product's own published terms.
21. **"Vianordis is the control layer" removed.** Present-indicative claims about unbuilt properties are rewritten as design commitments.
22. **Chapter 11 open items enumerated** — trademark registry act, related-party valuation and tax treatment, personnel separation, licence reversion triggers, dissolution and asset lock — rather than covered by a general "subject to legal review" clause.
23. **CLA/DCO addressed** (§11.4), including the Estonian moral-rights constraint under AuÕS that makes a US-template CLA inadequate.
24. **Publisher disclosure, named author, imprint, document licence, citation form, feedback route and legal-status date added.** An "Open Review Edition" with no review channel was self-contradicting.
25. **The vendor chapter moved to an annex** and the product block removed from the executive summary.

Structure and language

26. **Board brief on one page** added before the executive summary.
27. **Impact tiers moved forward** to §2.4, since Chapters 6 to 10 depend on them; **limitations moved forward** to Chapter 3, before the architecture rather than after it.
28. **Separate principles chapter dissolved** into Chapter 7, removing the largest redundancy in v0.2.
29. **Control chain reduced from three verbatim block quotes plus seven prose paraphrases to two statements** — the executive summary and the canonical definition in §4.2 — with cross-references thereafter.
30. **Employee representation and co-determination addressed** (§3.5) — v0.2 named works councils in its audience and then said nothing to them; the dual-use character of the evidence layer is now stated explicitly.
31. **Availability, cost, latency and brownfield migration acknowledged** (§3.3, §3.4) as open weaknesses rather than omitted.
32. **Maturity model replaces the 20-question checklist** (Chapter 10), scored and banded.

33. **Reference diagram corrected** (§7.1): the evidence path now originates from the control plane and the broker as separate decision and execution records, rather than appearing to originate from the ERP; the revocation and re-evaluation return path is drawn; and the note on payload staying in the system of record is carried into the figure.
34. **Terminology fixed**: Mandate/Mandat, Evidence/Evidenz, Policy/Richtlinie and Protokoll/Log inconsistencies resolved; glossary extended with the terms used but previously undefined; heading hierarchy, table of contents and footnote rendering corrected.

What changed in v0.4, and why

Version 0.4 incorporates the structured review of v0.3 and advances the paper from architectural argument to candidate implementation profile.

Corrections

1. **Foundation asset-lock statement corrected** (§11.5). Estonian foundation law does not automatically create the intended dissolution lock in every scenario; the articles must explicitly exclude reversion and name mission-compatible recipients.
2. **“Three measurements” corrected to four** and the metrics expanded to include precondition freshness (§10.5).
3. **BSIG liability wording narrowed** in the one-page brief to potential, fault-based internal liability under applicable company law.
4. **CRA scope and reporting timelines corrected** in the one-page brief: the duty applies to manufacturers of in-scope products with digital elements; vulnerabilities and severe incidents have different final-report deadlines.
5. **Unsupported market-wide wording softened** and framed as a hypothesis to be measured.
6. **Prior-art table revised** so that reuse of OPA, Certificate Transparency and SLSA does not imply that privacy, lifecycle and operational integration disappear.

Transaction assurance

7. **TOCTOU and mutable target state added** (§1.4, §3.6, §7.7, §8.6). Approval now binds to material preconditions and expires when they change.
8. **Action commitment and precondition commitment separated.** This distinguishes “the approved values were executed” from “the state that justified them still held”.
9. **Replay, idempotency and partial-commit states added** to the action broker.
10. **Execution receipts added** as a first-class evidence artefact.
11. **Ten control invariants added** (§4.4), including authority separation, no high-impact bypass, state freshness, evidence independence and safe failure.

Evidence engineering

12. **Trusted time and cryptographic lifecycle added** (§3.7, §7.8.4), drawing on RFC 3161 and RFC 4998.
13. **Model/provider reproducibility status added** (§7.8.5) so mutable provider aliases are not misrepresented as exact versions.
14. **Privacy-preserving evidence access added** (§7.8.6), including role separation, query logging and controlled re-identification.
15. **Canonicalisation and standard signature containers added** using RFC 8785, JWS or COSE.
16. **Evidence claim limits added** (Annex B.4): tamper-evident is not immutable; signed is not correct; approved is not truthful.

Implementation profile and adoption

17. **Governed Action Envelope introduced** (§7.12) with an illustrative signed data model and lifecycle state machine.

18. **GAIP-0.1 candidate profile added** (Annex G), defining artefacts, required fields, processing rules, conformance levels and negative tests.
19. **Impact tier and provenance assurance separated** (§2.5) using P0–P4 classes and a conservative control matrix.
20. **Brownfield enforcement modes M0–M4 added** (§7.13), separating inventory, observation, shadow decision, soft enforcement and hard enforcement.
21. **Brownfield integration patterns added** (§9.7): credential choke point, network mediation, target-native workflow and observe/reconcile.
22. **Safe-degradation matrix and break-glass requirements added** (§7.14–§7.15).
23. **Pilot acceptance criteria expanded** to include revocation, stale state, replay, bypass, evidence compromise and partial commit.

Measurement and release credibility

24. **Mandatory maturity gates added** so a high score cannot conceal self-issued authority, bypass or unprotected evidence keys.
25. **Operational measures added**, including governed-action coverage, policy latency, false blocks, stale approvals, break-glass use and cost per action.
26. **Measurement protocol added** (Annex H) with sampling, metric definitions and a publication template.
27. **Vianordis general-availability evidence gate added** (Annex C.8), listing what has not yet been published or independently verified.
28. **Mermaid source figures added** for architecture, impact/provenance, state machine, execution sequence and brownfield migration, making the diagrams renderable as SVG.

Annex G — Candidate Governed Action Interchange Profile

G.1 Status and purpose

GAIP-0.1 is a candidate, vendor-neutral profile for exchanging and verifying a governed action between an authority decision point, a human approval service, an enforcement point and an evidence service.

It is published for open review. It is not an IETF, ISO, CEN/CENELEC, ETSI or regulatory standard. Conformance claims should identify the exact profile version, action scope, assurance level, implementation version and assessment date.

The profile has three goals:

1. make the object approved by a human the same object evaluated and constrained by the enforcement point;
2. carry parameter provenance and material target-state preconditions with that object;
3. produce a verifiable relationship between the authority decision and the committed result.

GAIP does not define an AI model interface, policy language, identity provider, workflow engine or target-system API. It defines the minimum contract between them.

G.2 Conformance language

The key words **MUST**, **MUST NOT**, **REQUIRED**, **SHALL**, **SHALL NOT**, **SHOULD**, **SHOULD NOT**, **RECOMMENDED**, **MAY** and **OPTIONAL** are used as requirement terms for this candidate profile.

They do not state legal obligations. A legal or regulatory requirement must be traced separately to its source.

G.3 Roles

Role	Responsibility
Principal Resolver	Resolves the human or organisational principal and the legal or organisational context
Workload Identity Authority	Attests the acting agent, application or service instance
Mandate Authority	Issues, validates and revokes bounded assignments
Context and Provenance Resolver	Supplies data classification, target context, parameter origins and assurance classes
Policy Decision Point	Evaluates the action against versioned policy and returns a signed decision
Approval Service	Presents the complete material action to an authorised human and binds the approval
Policy Enforcement Point / Action Broker	Revalidates and executes through controlled target credentials
Target Adapter	Maps the governed operation to target-native conditional, idempotent or compensating semantics
Evidence Service	Correlates, signs, checkpoints and serves decision and execution evidence
Witness / Timestamp Service	Provides independent checkpoint or time evidence for higher-assurance profiles

One component may perform more than one role in lower-impact deployments. For tier 4 and 5 actions, the Mandate Authority, Action Broker and Evidence-signing/key roles SHOULD be separated so that compromise of one does not confer all authority and history.

G.4 Required artefacts

A GAIP transaction consists of:

1. **Intent Proposal** — structured proposed business action and origin marking;
2. **Mandate Reference** — bounded authority issued by a separate authority role;
3. **Policy Decision Record** — decision, policy digest, inputs and obligations;
4. **Approval Record** — where required, identity and signature bound to action, provenance, preconditions and expiry;
5. **Governed Action Envelope** — the short-lived object delivered to the enforcement point;
6. **Execution Receipt** — outcome and committed target-state evidence;
7. **Evidence Record** — correlated decision and execution proof with checkpoint status.

An implementation MAY store these as separate signed objects referenced by digest or as a signed composite envelope. It MUST be possible to verify that the references resolve to the exact objects used.

G.5 Envelope data model

The envelope MUST contain or cryptographically reference the following fields:

Field	Requirement
`profile`	Exact GAIP profile version
`transaction_id`	Globally unique transaction identifier
`tenant_id` / authority domain	Domain within which identities, policies, keys and evidence are resolved
`principal`	Human or organisational principal; MUST NOT be only a shared technical account
`actor`	Distinct workload and instance identity
`intent`	Type, schema, structured values and origin marking
`mandate`	Identifier, version, purpose, scope, risk limit, delegation rule, validity and revocation reference
`context`	Legal entity, business scope, data class, action tier, provenance classes and relevant security posture
`parameters`	Name or field identifier, value or protected reference, source, provenance class, verification status and digest
`policy_decision`	Decision ID, effect, policy digest, decision time, obligations and expiry
`approval`	Where required: approver, authority, action digest, precondition digest, presentation digest, time and expiry
`preconditions`	Material resource versions, states, balances, policy or sanction results and freshness rules
`execution_constraints`	Target, operation, allowed adapter, idempotency key, maximum attempts and assurance mode
`evidence_profile`	Canonicalisation, signature, timestamp, checkpoint and retention profile

The profile SHOULD minimise disclosure. Sensitive values MAY be encrypted or represented by digests where the enforcement point can resolve them securely and the approver is shown an intelligible representation.

G.6 Mandatory invariants

A conforming GAIP-0.1 enforced implementation:

1. **MUST** resolve a principal and distinct actor identity before evaluation.
2. **MUST NOT** accept a mandate minted by the same untrusted business application or model that requests the action, unless an independently controlled authority role validates and re-issues it.
3. **MUST** validate mandate purpose, scope, risk limit, validity and revocation.
4. **MUST** record the origin and provenance assurance class of every impact-determining parameter.
5. **MUST NOT** treat model-generated intent or parameters as authority.

6. **MUST** bind required approval to the canonical action, provenance presentation, material preconditions and expiry.
7. **MUST** revalidate expiry, mandate revocation, policy obligations and material preconditions immediately before commit.
8. **MUST** reject or re-evaluate a stale envelope.
9. **MUST** provide replay or duplicate protection for side-effecting actions.
10. **MUST** emit an execution receipt for success, denial, stale state, partial commit, compensation and indeterminate outcome.
11. **MUST** maintain a verifiable link between policy decision, approval, envelope and execution receipt.
12. **MUST NOT** permit a declared tier 4 or 5 action to bypass the enforcement point except through a separately defined break-glass profile.
13. **MUST** declare the safe-failure mode for every supported action tier.
14. **MUST** declare the canonicalisation, signature, key, time and checkpoint profile used.
15. **MUST** declare the action scope and date of any conformance or no-bypass claim.

G.7 Processing procedure

A verifier or enforcement point processes an envelope in this order:

1. Parse the envelope and reject unknown critical profile features.
2. Canonicalise according to the declared profile.
3. Verify signatures, key identifiers, authority domain and required certificate or trust-chain status.
4. Verify transaction uniqueness and idempotency state.
5. Resolve principal and actor status.
6. Validate the mandate chain, purpose, risk limit, validity and revocation.
7. Validate the policy decision signature, digest, obligations and expiry.
8. Validate the approval, including approver authority, presentation digest, action digest, provenance and precondition digest.
9. Resolve current material state and compare it with the preconditions.
10. Re-evaluate policy where a declared input changed or where policy requires commit-time evaluation.
11. Execute once through the declared target adapter and assurance mode.
12. Obtain or create the target and broker execution receipt.
13. Emit and checkpoint the evidence record.

A failure at steps 1 to 10 MUST NOT fall through to execution. The outcome is recorded as denied, invalid, expired or stale.

G.8 Canonicalisation and cryptographic profiles

GAIP-0.1 permits at least two representation families:

JSON profile

- I-JSON compatible data model;
- RFC 8785 JSON Canonicalization Scheme;
- JWS or another explicitly declared signature container;
- digest and algorithm identifiers carried in protected headers;
- optional encrypted disclosures for sensitive values.

CBOR profile

- deterministic CBOR representation;
- COSE Sign1 or COSE Sign structure under RFC 9052/9053;
- countersignature or multiple signatures where multi-party approval is required;
- optional RFC 3161 timestamp integration through an explicitly defined wrapper or COSE timestamp profile.

The profile MUST prohibit ambiguous duplicate field names and MUST define number, time, Unicode and binary-value representations. Implementations MUST publish test vectors before claiming interoperability.

G.9 Time, expiry and key rules

- Times MUST be UTC and include sufficient precision for the process.
- The envelope MUST contain an expiry; tier 4 and 5 approvals SHOULD have short, action-appropriate validity.
- The verifier MUST enforce a maximum clock skew.
- The evidence profile MUST identify whether time is local, monitored, independently timestamped or qualified under a separate trust-service framework.
- Key identifiers and validation material required for later verification MUST be retained for the evidence lifetime.
- A key-compromise event MUST trigger a documented review of affected evidence ranges.
- Long-retention evidence SHOULD support re-sealing or archive-timestamp renewal before algorithm or key expiry.

G.10 Target-adapter assurance modes

Each adapter declares one mode:

Mode	Required behaviour	Assurance statement
`conditional-commit`	Target atomically enforces version or state preconditions and idempotency	Strongest transaction equivalence
`locked-transaction`	Adapter holds an appropriate target-native lock from final validation through commit	Strong, subject to lock scope and failure semantics
`immediate-reread`	Adapter re-reads state immediately before an unconditional commit	Detects many stale states but retains a race
`compensating`	Multi-step action uses defined compensations and records partial outcomes	Eventual repair, not atomicity
`observe-only`	No enforcement; records outcome for comparison	No prevention or no-bypass claim

Tier 4 and 5 actions SHOULD use `conditional-commit` or `locked-transaction`. Use of a weaker mode MUST be visible to the approver where it materially changes risk and MUST be present in the evidence record.

G.11 Conformance levels

Level	Minimum
GAIP-C — Correlated	Required identities and records are correlated after execution; no enforcement claim
GAIP-S — Shadow	Policy and envelope are produced in parallel with the existing path; no blocking claim
GAIP-E — Enforced	All mandatory invariants hold for the declared action set; no high-impact bypass; stale-state and replay tests pass
GAIP-H — High assurance	GAIP-E plus separated key/evidence boundary, conditional or locked commit for tier 4/5, high assurance evidence and independent assessment

Level	Minimum
GAIP-X — Externally verifiable	GAIP-H plus independent witness or transparency checkpoint and independently reproducible verification package

A conformance statement MUST name the action types and tiers covered. “The platform is GAIP compliant” without scope is not a valid statement.

G.12 Required negative tests

A GAIP-E or higher implementation MUST publish or make available results for at least these tests:

1. model attempts to add an action outside the intent schema;
2. application presents a self-issued or widened mandate;
3. mandate revoked after approval but before commit;
4. policy changed after approval;
5. target state changed after approval;
6. approval replayed with modified parameters;
7. exact envelope replayed after a successful commit;
8. duplicate delivery during target timeout;
9. evidence producer attempts to omit or reorder an entry;
10. business application attempts direct target access;
11. signing key rotated; historical verification still succeeds;
12. evidence signing key declared compromised;
13. control-plane component unavailable; declared safe state is observed;
14. partial multi-system commit; compensation succeeds and fails in separate tests;
15. cross-tenant identity, key or evidence reference supplied;
16. provider model alias cannot be resolved to an exact artefact; evidence records the weaker status.

The test report MUST state which tests were automated, which required manual review, the environment used and known exceptions.

G.13 Privacy and employee safeguards

GAIP conformance does not require every evidence field to be visible to every verifier. An implementation SHOULD support selective disclosure or purpose-specific evidence packages.

It MUST define:

- personal-data fields;
- retention per record class;
- roles allowed to query named records;
- conditions for re-identification;
- export and legal-hold rules;
- deletion or rectification behaviour for payload references;
- handling of employee-representation agreements where applicable.

A cryptographic digest of personal data can still be personal data when it is linkable or when the controller can resolve it. Hashing is not automatic anonymisation.

G.14 Non-goals

GAIP does not prove:

- that source data were truthful;
- that a model output was correct or unbiased;
- that the legal role or risk classification was correctly determined;
- that a human approval was wise or freely given;
- that a fully compromised authority domain cannot collude;
- that the target system faithfully implements its own receipt;
- that an organisation is compliant.

It aims to make the authority and execution claim bounded, testable and comparable.

Annex H — Measurement protocol and pilot scorecard

H.1 Purpose

This annex defines a minimum protocol for testing the central thesis and the operational value of a runtime authority pattern. It is designed so that an organisation can publish results without using Vianordis.

H.2 Unit of analysis

The unit is a completed, denied, stale, partially completed or indeterminate **AI-initiated or AI-assisted action transaction**. Purely informational model calls may be sampled separately but should not dominate a study intended to measure execution governance.

Each study must define:

- action types included;
- impact tiers and provenance classes;
- systems and business units;
- observation period;
- whether the path is M0, M1, M2, M3 or M4;
- whether the sample is pre-pilot, post-pilot or both.

H.3 Sampling

For each action type, draw a random or systematically sampled set large enough to include normal, denied, exceptional and failed transactions. Convenience samples selected by the implementation team are not sufficient for an independent reconstruction measure.

The reviewer measuring reconstruction time should not have built the integration and should receive only the access and documentation that an internal auditor or incident responder would normally receive.

Record exclusions and missing transactions. An unexplained exclusion is a failed reconstruction, not an absent denominator.

H.4 Thesis metrics

H.4.1 Reconstruction coverage

Numerator: sampled transactions for which the reviewer reconstructs principal, actor, purpose, mandate, policy version, decisive parameter provenance, approval status and execution outcome.

Denominator: all sampled transactions, including denied, failed and indeterminate outcomes.

Publish both full reconstruction and partial reconstruction by missing field.

H.4.2 Time to reconstruct

Start when the reviewer receives the transaction identifier or business reference. Stop when the reviewer produces a complete evidence package or declares the reconstruction incomplete.

Publish median, p95, maximum, staff minutes and number of systems or people consulted.

H.4.3 Approval and state-binding rate

For tier 4 and 5 sampled actions report:

- proportion with approval bound to executed parameters;
- proportion also carrying parameter provenance;
- proportion also carrying material preconditions;
- proportion with commit-time state verification;
- number and rate of stale approvals detected and rejected;
- number of approvals executed after expiry or mismatch.

H.4.4 Governance-constrained initiatives

For the relevant planning period, classify AI initiatives stopped, delayed, reduced in autonomy or materially rescoped. Record the primary and secondary reasons using at least:

- model quality;
- business case;
- cost;
- data availability;
- data protection;
- cybersecurity;
- authority/accountability;
- auditability/evidence;
- integration complexity;
- workforce or co-determination concerns;
- supplier or sovereignty risk.

The category should be assigned from decision records, not retrospective intuition.

H.5 Operational metrics

Metric	Definition
Machine-path latency	Identity through evidence emission, excluding human and queue time; p50/p95/p99
Approval latency	Presentation to decision; p50/p95 and oldest queue age
False block	Action later determined permissible but blocked by control logic
False permit	Action permitted despite a policy or mandate condition that should have blocked it
Stale-approval rejection	Envelopes rejected due to changed preconditions or expiry
Bypass exception	Action completed through a path outside declared enforcement
Replay prevented	Duplicate attempt rejected or mapped to the original outcome
Partial commit	Multi-step action not fully completed atomically
Compensation failure	Partial action for which declared compensation did not restore the intended state
Break-glass use	Count, purpose, approvers, duration and review closure
Evidence verification success	Percentage of sampled records whose signatures, timestamps and checkpoints validate
Cost per governed action	Allocated service, storage, approval and operating cost divided by governed actions

H.6 Pilot scorecard

A pilot report should include:

Area	Before	After	Target	Result
Reconstruction coverage	—	—	Defined by organisation	Pass / fail
Median reconstruction time	—	—	—	—
p95 machine-path latency	—	—	—	—
Median approval latency	—	—	—	—
Approval parameter-binding rate	—	—	100% for declared tier 4/5 scope	—
Precondition-binding rate	—	—	100% for material preconditions	—
Stale approvals rejected	—	—	100% of injected test cases	—
Replay tests passed	—	—	100%	—
Known bypasses	—	—	0 for declared M4 action scope	—
Evidence verification success	—	—	100% of sample	—
False-block rate	—	—	Organisation-defined	—
Partial/indeterminate outcomes	—	—	Organisation-defined and reviewed	—
Break-glass exercises reviewed	—	—	100%	—
Cost per governed action	—	—	Organisation-defined	—

Blank values are deliberate. Version 0.4 does not contain field measurements and must not manufacture them.

H.7 Publication template

A published result should state:

1. organisation type and approximate scale, without disclosing sensitive identity where necessary;
2. action scope, tiers and provenance classes;
3. enforcement modes and target-adapter assurance modes;
4. sample method and size;
5. metric definitions and any deviations from this annex;
6. results with denominators;
7. known bypasses, exclusions and failed tests;
8. who performed the review and whether they were independent of the implementation;
9. date and software/profile versions;
10. conflicts of interest and vendor involvement.

H.8 Interpretation

A result showing excellent reconstruction without a dedicated control plane is evidence against a product-centric reading of this paper and in favour of the architecture being assembled from existing controls. That is a valid outcome.

A result showing that governance rarely constrains AI adoption weakens the commercial urgency claim even if the control gap exists technically. That is also a valid outcome.

The objective is not to produce a benchmark that Vianordis wins. It is to determine whether the problem is material, which controls solve it, and at what operational cost.

Imprint and contact

Publisher and licensor

GoSec Cloud OÜ 9 Järvevana tee, 11314 Tallinn, Estonia Estonian Commercial Register no. 17359627 · VAT EE102990704 Registry court: Tartu District Court Authorised representative: Yves Frédéric N'Soussoula info@gosec.cloud · +372 60 26553

Vianordis is a product of GoSec Cloud OÜ and is not itself a legal entity.

Author

Yves Frédéric N'Soussoula, Founder & CEO, GoSec Cloud OÜ.

Feedback on this paper — Open Review

Substantive criticism, corrections and counter-arguments are welcome through https://www.vianordis.eu/en/contact. Please indicate the section number. Corrections of fact will be applied in the next edition and listed in Annex F; substantive critique will be credited unless you prefer otherwise.

Related routes: platform and applications https://www.vianordis.eu/en/services · release status https://www.vianordis.eu/en/release-status · source and licensing https://www.vianordis.eu/en/git · trust centre https://www.vianordis.eu/en/trust · community and contribution https://www.vianordis.eu/en/build-the-constellation

Document licence

Cite as

N'Soussoula, Y. F. (2026). *The AI Control Gap: Why model catalogues, policies and API gateways stop short once AI acts inside the enterprise*. Vianordis Position Paper 01, version 0.4. Tallinn: GoSec Cloud OÜ. https://www.vianordis.eu

Legal status: law as at 14 August 2026. Not legal advice.

End of draft — version 0.4