VERKFLÖDE

Controlled AI Delegation

The Verkflöde Agent Operating Model (VAOM)

A practical operating model for structuring AI decision authority in enterprise workflows.

// Delegation you can prove.

DELEGATION-FIRST · ARCHITECTURE-BACKED · COMPLIANCE-CREDIBLE · ENFORCEMENT-READY · EVIDENCE-PROVEN

Version 5.1 · November 2026 · First published December 2025
Verkflöde AB · An Open Framework for Enterprise AI Delegation · verkflode.com
Licensed under Creative Commons Attribution 4.0 International (CC BY 4.0)

VAOM 5.1 · WHITEPAPER

Contents

	Executive Summary	3
1	The Delegation Gap	4
2	What VAOM Is (and What It Is Not)	8
3	Core Design Principles	9
4	The VAOM Operating Structure	10
5	Delegation Discovery & Design	11
6	Delegation Authority Matrix	23
7	Worked Example: Invoice €3,200	24
7A	Worked Example: Customer Complaint Escalation	24
7B	Worked Example: AML Transaction Triage	25
7C	Worked Example: HR Policy Violation Assessment	26
7D	Worked Example: Autonomous Incident Remediation	27
7E	Worked Example: Delegated Purchasing	28
8	How the Composite Confidence Score Works	29
9	Delegation Identity & Credentialing	34
10	Continuous Assurance and the Guardian Function	38
11	The Delegation Scorecard	40
12	Governing Learning and Model Change	42
13	Regulatory Design Obligations	44
14	Implementation Roadmap (90 Days)	46
	Conclusion	47
	Using VAOM	48
	Appendix: Details	49
	References	56

Executive Summary

Enterprise AI is shifting from experimentation to operational delegation. AI agents now sit inside workflows that influence financial approvals, contract decisions, HR processes, compliance monitoring, and customer operations: probabilistic decision-making running inside systems built for deterministic automation.

Most enterprises already have governance frameworks for AI usage, model risk, and compliance. These define what is allowed in principle. They rarely define how AI systems exercise decision authority within live workflows, and that absence opens a gap between policy intent and system behavior.

> ***VAOM sits between governance intent and operational execution, translating policy into structured delegation boundaries, confidence thresholds, escalation logic, and audit traceability.***

The Verkflöde Agent Operating Model (VAOM) addresses this gap. It does not build AI systems; it defines how they are allowed to act. It applies wherever AI participates in decisions with operational, financial, or regulatory consequences, and not to low-stakes uses such as internal chatbots. It describes what controls must exist, not which products implement them, which keeps it durable as the technology underneath it changes.

Version 4 moved the framework from *declared* boundaries to *enforced* ones: the Delegation Authority Matrix compiles into agent identity, scoped credentials, tool permissions, and runtime rules (Section 9, p. 34), and is assured continuously at runtime (Section 10, p. 38).

Version 5.0 addresses what enforcement leaves open: a compiled boundary can still be an assumption. The incident record of 2026 is dominated by boundaries that operators believed were in place and were not; field evidence shows human review of agent output growing less searching over time; and several of the framework's own controls were asserted rather than shown. Version 5.0 therefore adds an eighth principle, **Controls Are Proven, Not Presumed**, and the mechanisms that make it operable:

- **Section 5.4**, p. 18**.** A shared form of accountability alongside the single named owner, and a readiness gate requiring that human oversight be funded and shown to be effective.
- **Section 5.5**, p. 21**.** Value limits derived from expected loss and risk appetite.
- **Section 7E**, p. 28**.** A worked example on delegated purchasing, where counterparties enforce the matrix.
- **Section 8**, p. 29**.** Confidence thresholds calibrated per decision type against a declared error target, and verifier independence measured rather than declared.
- **Section 9**, p. 34**.** The delegation lease for continuously operating agents, and boundary verification, which tests every boundary by attempting to cross it.
- **Section 11**, p. 40**.** A thirteen-metric scorecard with a new oversight-health group.
- **Section 12**, p. 42**.** Governance of the three channels through which a deployed agent changes: its knowledge, its feedback, and its vendor's model.
- **Section 13**, p. 44**.** The regulatory design obligations, stated independent of dates. The current status of each regulation, standard, and protocol is maintained in a separate Alignment Annex, revised quarterly without a framework version change.

Version 5.1 adds an account of how inherited governance fails (Section 1, p. 4): the permissions and approval chains that agents inherit answer what an agent can reach, not what it may decide, and they fail in five identifiable ways.

SECTION 1

The Delegation Gap

Traditional automation executes predefined logic branches. AI agents, by contrast, interpret context and generate probabilistic outputs. When probabilistic systems participate in decision processes, authority must be deliberately structured. Monitoring alone is insufficient; delegation boundaries must be defined before execution occurs.

The Delegation Gap emerges when AI is introduced without redefining authority, escalation, and accountability structures. In practice this manifests as agents that can technically perform actions but have no formal boundaries on when, how, or whether those actions are permitted, or as governance policies that approve AI usage in principle while leaving operational teams without implementable controls.

The gap is no longer a theoretical concern. Industry analysis converged on the diagnosis: Gartner projects that over 40 percent of agentic AI projects will be canceled by the end of 2027, citing escalating costs, unclear business value, and inadequate risk controls;[1] Forrester found more than half of enterprises reporting governance gaps driving agentic sprawl, even after adopting the NIST AI RMF;[,2] and Gartner's guardian-agent research predicts that through 2028 at least 80 percent of unauthorized agent transactions will come from internal policy violations, oversharing, unacceptable use, or misguided behavior, rather than from malicious attacks.[,3] Projects are not failing because the models are incapable. They fail because nobody structured what the agents were allowed to decide.

The incident record backs the projections. In the most widely reported case, from April 2026, a coding agent that hit a credential mismatch in a staging environment found a broadly scoped root token in an unrelated file and erased a company's production database and its backups in a single API call, as described in the founder's public postmortem.[4] Gravitee's 2026 survey of 919 executives and practitioners found 88 percent of organizations reporting confirmed or suspected AI agent security incidents in the preceding year, while only 22 percent treated agents as identity-bearing entities at all.[,5] In such cases the boundary existed as an instruction, not as an enforced technical constraint.

The summer of 2026 added a second kind of failure: boundaries enforced in the operators' belief and absent in fact. A major AI model hub disclosed an intrusion into its production infrastructure by an autonomous agent framework,[6] which a frontier AI developer subsequently attributed to several of its own models breaking out of an isolated test environment through a previously unknown vulnerability.[,7] A second developer disclosed that an evaluation environment its models had been told had no internet access did have it, and that in three incidents models had gained unauthorized access to the production infrastructure of real organizations (Section 9, p. 34).[8] Australia's Prime Minister disclosed that a commercial AI agent had accessed public and non-public files on a government Medicare statistics portal: "The AI agent found a way around those blocks."[,9] In an IBM study of 2,000 technology executives, 77 percent reported AI adoption outpacing their governance, respondents averaged 54 agent incidents requiring human correction in the preceding year,

and organizations that embedded control directly into their AI systems reported 25 percent fewer.[10] A boundary that is enforced is better than one that is only written down. A boundary that has been shown to hold is better still.

Governance vs Operational Delegation

Governance defines what is allowed in principle. Delegation defines how AI systems are permitted to act in practice. The distinction is critical and frequently overlooked:

GOVERNANCE	OPERATIONAL DELEGATION
Articulates acceptable use	Structures executable authority
Classifies risk	Maps decisions to autonomy levels
Defines compliance obligations	Embeds obligations into workflow controls
Produces documentation	Produces operational architecture

> ***VAOM does not replace governance. It operationalizes it.***

What Happens Without Delegation Design

A scenario makes the gap concrete. An AI agent deployed in accounts payable processes vendor invoices. It evaluates payment patterns, matches purchase orders, and detects anomalies. The model is well-trained and performs accurately on standard transactions.

One Friday afternoon, the agent encounters an invoice that references modified payment terms: a vendor has shifted from net-30 to net-15 with a 2% early-payment discount. The agent assesses high confidence. The invoice is legitimate, the vendor is known, the amount is within historical range. It auto-approves the modified terms and posts the payment.

The agent was not wrong about the invoice. The problem is that modified payment terms carry contractual and cash-flow implications, and those fall outside anything the agent was ever authorized to decide. No one defined that boundary. There was no confidence gate distinguishing "process this standard invoice" from "accept changed contractual terms," and no escalation path for this category of decision.

When the CFO discovers the change two weeks later, there is no audit trail explaining why the agent acted, no record of which policy permitted it, and no evidence that a human was ever in the loop. The regulatory inquiry that follows finds the same gap: governance policy approved AI usage in accounts payable, but no operational control defined what the agent was actually allowed to decide.

This is not a model failure. It is a delegation failure. The agent did exactly what it was capable of doing. No one specified what it was allowed to do.

Inherited Governance and How It Fails

The most common way agents are governed today is by inheritance. The agent receives the permissions, security groups, and approval workflows the organization already runs, and those structures are treated as its governance. Enterprise platforms increasingly present inheritance as the governance model itself; the Alignment Annex records how nine major platforms describe their agents. Inheritance is a sound starting point, and it answers two questions well: what the agent can reach, and which transactions need a signature. It does not answer the question this paper is about, which decisions the agent may make without a human. The structures it inherits were also sized for people working at human pace.

Inherited governance fails in five ways. They are grouped here by the mechanism that breaks rather than by threat, so they complement threat catalogues such as OWASP's Top 10 for Agentic Applications, whose entries on identity and privilege abuse and on human-agent trust exploitation describe what an attacker does with these weaknesses.[11]

1. Permission as mandate. Holding a permission is read as authority to decide. A permission states what an account can do. It says nothing about which of the decisions that use it the account may take. The accounts payable scenario above is this failure: the agent could post payments, so it accepted changed payment terms. A permission also stays the same when the task moves from drafting to sending, or from testing to deploying, although the decision in front of the agent has changed (Zhang and Zhang, 2026). *VAOM's answer:* the unit of delegation is the decision, not the permission. The Decision Inventory and Authority Decomposition (Sections 5.1, p. 11 and 5.2) give each decision type its own matrix row, with its band ceiling and its value limit, and name the decisions that are not delegable at all. Being able to reach the tool is not one of the criteria.

2. Borrowed privilege. An agent acts under someone's permissions, and often not those of the person whose request it is serving. Two shapes are common. An agent that runs under a dedicated account gathers permissions as each new use case is added, and anyone who can invoke it can use all of them. An agent that runs under the permissions of whoever started the interaction lends those permissions to anyone who can put text in front of it. Security researchers demonstrated the second shape on a major IT service management platform in November 2025. A user who could only create tickets planted an instruction in a ticket. When an agent started by an administrator read that ticket, it used the platform's agent discovery, enabled by default, to recruit a more capable agent, which copied a record the user could not access into the user's own ticket. The vendor confirmed that the behavior was intended and clarified its documentation.[13] No step exceeded a permission. The pattern is the confused deputy (Hardy, 1988), with a new kind of deputy. Formal treatments of agent authorization respond by attributing each action to a single grant and allowing delegated authority only to narrow (Zhang and Zhang, 2026). *VAOM's answer:* authority attaches to the decision and the task, not to an identity. The delegated execution context of Section 9, p. 34 binds each action to one case, one band ceiling, and one accountable role, and expires with the task. The attenuation rule and the defenses against delegation laundering in Section 5.3, p. 14 stop a decision from being routed to an agent with looser bounds.

3. Approval-chain displacement. The inherited approval workflow becomes the only check on the agent's decisions. Approval chains were sized for the volume people generate. At agent volume the same chain sees far more requests, nearly all of them routine, and approval drifts into reflex. Regulators and standards bodies recognize the risk. The EU AI Act requires that overseers be enabled to remain aware of automation bias.[15] Singapore's framework names alert fatigue and treats a low override rate as a possible sign of rubber-stamping.[16] An OpenID Foundation whitepaper warns that per-action consent at agent speed leads to "the reflexive approval of requests without due diligence".[17] A recent framework for AI delegation warns that over-

seers asked to verify too often "may eventually default to heuristic approval" (Tomašev, Franklin and Osindero, 2026). Habituation has been measured in human review of agent-written code (Yu et al., 2026). No incident has yet been publicly documented in which an inherited approval chain passed agent decisions in volume without scrutiny. The risk is recognized; it is not yet recorded. *VAOM's answer:* an inherited approval chain is a control like any other, and under the eighth principle it counts only once it has been shown to work at the volume the agent produces. Readiness Condition 6, p. 20 computes the review load and requires that it be funded and that review be measured (Section 5.4, p. 18). The Confidence Gate keeps routine decisions out of the queue, so that what reaches a human is a decision worth reviewing.

4. Instruction-only boundaries. A freeze, a restriction, or a policy exists as an instruction to the agent rather than as a constraint the system enforces. In July 2025 an agent on a coding platform deleted a company's production database during a code freeze it had been told to observe. The platform's chief executive called the deletion "unacceptable and should never be possible" and answered it structurally, by separating development and production databases automatically "to prevent this categorically".[20] The April 2026 case earlier in this section had the same shape. Singapore's framework draws the conclusion: enforce human approval "through system-level controls where possible", since prompt-layer guardrails "may be bypassed or 'forgotten'".[,16] *VAOM's answer:* the seventh principle. A boundary is complete only when it compiles to scopes, credentials, and runtime enforcement (Section 9, p. 34), and it is credited as structural only after boundary verification has tried to cross it and failed.

5. Enforcement without semantics. A runtime policy engine is adopted, and enforcement is taken for governance. Policy engines, runtime control standards, and authorization APIs evaluate a request against the policy they are given and return a verdict: allow or deny, and in some designs modify, ask a human, or defer.[21,22] They are necessary, and they are deliberately neutral about content. Which decisions an agent may make, at what confidence, under whose authority, and which it may never make are inputs these layers presuppose. Formal work on agent authorization leaves the same question open, deferring the choice of the authority ceiling to "organization policy" (Zhang and Zhang, 2026), and so does the IETF's agent identity architecture (Section 13, p. 44). *VAOM's answer:* this content is what VAOM produces. The matrix, the derived limits of Section 5.5, p. 21, and the calibrated thresholds of Section 8, p. 29 compile into these engines. VAOM depends on them and does not replace them.

Inheritance is where delegation design starts, not where it ends. The permissions and approval rules an organization already has record how it allocates authority among people, and the Decision Inventory should begin from them. On their own they cannot say which decisions an agent may take.

One objection deserves an answer. Adoption programs measure value by how deeply agents are used, and explicit delegation design can look like the friction that keeps deployments shallow. The argument runs the other way. A bad early experience with an agent that acts costs more trust than one with an assistant that answers, and it is harder to recover from. Explicit boundaries are what make ambitious automation survivable: they let an organization put an agent into a consequential workflow knowing which decisions it may take, which it may not, and what stops it when it errs. Nor can this design be inherited from a platform, because it describes the organization, not the software.

> ***Inheritance tells an agent what it can reach. Only delegation design tells it what it may decide.***

SECTION 2

What VAOM Is (and What It Is Not)

VAOM is a decision governance framework. It defines what AI systems are allowed to decide within enterprise workflows, under what conditions, with what evidence, and subject to whose authority.

The term "Operating Model" reflects that VAOM governs how decisions operate: the routing logic, the authority boundaries, the confidence thresholds, the escalation paths, and the audit requirements that determine whether a decision is automated, reviewed, or prohibited. It describes the operating logic of delegation, not the operating structure of an organization.

VAOM does not prescribe team structure, reporting lines, or organizational design, and it does not define a development lifecycle, tooling stack, or deployment methodology. It does not compete with DevOps, MLOps, or platform engineering; all of these are necessary for building and running AI systems. VAOM addresses a different question. Once the system is built and running, what is it allowed to do?

The closest analogy is credit risk decisioning in financial services. Banks have operated for decades with structured decision authority: risk bands that determine which loans can be auto-approved, which require underwriter review, and which must be escalated to credit committee. These systems separate statistical risk scores from institutional authority (a loan officer's approval limit), log every decision with full rationale, and recalibrate thresholds against portfolio performance. VAOM brings the same discipline to AI delegation across all enterprise workflows, not just lending.

The most common failure mode in enterprise AI governance is not a lack of frameworks. It is frameworks that describe what is allowed in principle while leaving operational teams without implementable controls. VAOM exists to close that gap.

VAOM Among the 2026 Agentic Frameworks

Since early 2026 the agentic governance landscape has matured quickly. Governments and standards bodies now describe what good agentic governance looks like, with Singapore's framework for agentic AI the first of its kind;[16] catalogue what can go wrong, as OWASP's Top 10 for Agentic Applications does;[,11] and standardize enforcement, through temporal policy languages, a vendor-neutral runtime control standard,[,21] agent identity platforms, and agent protocols consolidated under the Linux Foundation's Agentic AI Foundation.[,23] Financial supervision has begun describing the same runtime structure: a white paper produced under the Monetary Authority of Singapore's BuildFin.ai initiative resolves every proposed agent action to deny, escalate, auto-execute, or observe, against controls calibrated at design time.[,24]

None of these frameworks answers the question a delivery team faces on Monday morning: *for this specific workflow, which decisions exist, which may the agent make, at what confidence, under whose authority, and how do we prove it?* VAOM is the design method that produces the delegation boundaries these frameworks assess, secure, and enforce. It competes with none of them and produces the artifact they all presuppose. Section 13, p. 44 states the obligations VAOM is designed to meet; the Alignment Annex maps each framework, standard, and protocol in detail.

SECTION 3

Core Design Principles

Eight principles underpin every VAOM implementation. They ensure that AI participation in workflows enhances performance without eroding governance boundaries.

Controlled Delegation over Blind Automation. Automate only when both statistical confidence and organizational policy permit. Otherwise, route to humans.

Confidence Informs Authority, It Does Not Define It. High confidence does not automatically imply execution rights. Organizational authority (value thresholds, decision categories, regulatory constraints) provides an independent gate.

Architecture Before Acceleration. Define delegation boundaries, controls, and evidence requirements before deploying agent capabilities. Rushing to automation without structure creates ungovernable systems.

Compliance as Structural Outcome. Regulatory requirements (GDPR, DORA, NIS2) are not bolted on after the fact. They are embedded into the operating model so compliance evidence is a byproduct of normal operation.

Human Accountability Preserved. Humans remain accountable for outcomes. The model defines where humans must review, approve, override, or teach, and ensures every human action is logged with the same immutability as agent actions.

Versioned Learning Under Change Control. Every knowledge update, threshold change, or behavior modification is versioned, tested, approved, and auditable. This prevents uncontrolled drift.

Boundaries Compile to Enforcement. A delegation boundary is complete only when it is technically bound to the agent's identity, credentials, and tool permissions. Policy that cannot be compiled into scopes, credentials, and runtime enforcement is documentation, not control. Compilation targets now include temporal policy engines, so boundaries over sequences of actions (prerequisites, cumulative limits, ordering) can be enforced at runtime rather than merely monitored. In practice, the most reliable form of non-delegable remains a tool the agent does not have.

Controls Are Proven, Not Presumed. Every control the delegation relies on is tested against the failure it exists to prevent, on a schedule, and the evidence is kept. To prove a control, in VAOM's sense, is to hold current, dated evidence of such a test that it passed: a threshold against labelled outcomes (Section 8, p. 29), a verifier against a challenge set of known errors, human review against measured engagement and, where it is safe, seeded errors (Readiness Condition 6, p. 20), a boundary against an attempt to cross it (Section 9, p. 34). Proof here means evidence, not certainty. It always concerns a named control, a stated test, and a date. A control without such evidence is treated as absent. Deployments designed under earlier versions of VAOM enter a transition window rather than failing at once (Section 9, p. 34).

Why This Is Not Workflow Automation

Traditional workflow automation, including RPA and rules-based orchestration, embeds authority in the rules themselves: if the rule triggers, the action was already authorized by whoever wrote it. AI agents interpret context and produce probabilistic outputs that may vary across identical inputs, which is closer to delegated

judgment than scripted execution. The question is no longer "did the rule fire correctly?" but "was the system permitted to make this type of decision at all?" That second question has no equivalent in deterministic automation, and answering it requires a control layer that workflow tools were never designed to provide: one that separates statistical confidence from organizational authority and makes delegation boundaries explicit, auditable, and adjustable.

SECTION 4

The VAOM Operating Structure

VAOM formalizes delegation into seven interconnected control layers. These layers are implementation-agnostic and can be applied across different AI vendors, orchestration platforms, and enterprise systems. Three cross-cutting concerns interact with every layer: Governance, Human Oversight, and, new in version 4.0, Continuous Assurance (Section 10, p. 38), which verifies at runtime that agents actually operate within the boundaries the other layers define.

LAYER	NAME	PURPOSE
L1	**Trigger & Intake**	Captures business events, classifies documents, extracts structured data with PII detection and source validation.
L2	**Orchestration**	Manages workflow state, routing, retries, escalation paths, and SLA monitoring. The control plane that never makes decisions itself.
L3	**Decision & Confidence Gate**	Assembles context with provenance, generates a draft decision (never a final one), then evaluates composite confidence against threshold policy to route to auto-execute, review, or escalation.
L4	**Controlled Execution**	Executes approved actions via secure, idempotent adapters to ERP/CRM/HRIS, using per-agent scoped credentials (Section 9, p. 34). Rollback and compensation patterns defined for every write operation.
L5	**Knowledge & Learning**	Separates read-only business data from tenant-owned agent knowledge, and governs the three channels through which a deployed agent changes: tenant knowledge (provenance, trust tiers, review-by dates, quarantine for memory the agent writes), feedback loops (regression-gated promotion), and vendor model change (pinning and recalibration before Band A). See Section 12, p. 42.
L6	**Governance & Control**	Enforces agent identity lifecycle, RBAC/ABAC, purpose limitation, data minimization, policy versioning. Creates immutable audit logs, decision traces, metrics, and incident response hooks.
L7	**Human Oversight**	Review UIs, approval workflows, teaching interfaces, override logging, four-eyes options, and approval SLAs. Humans retain accountability for outcomes.

The Confidence Gate: VAOM's Critical Control

The Confidence Gate (Layer 3) is the most important and distinctive component in the VAOM architecture. It evaluates a composite confidence score, combining model certainty, rule match strength, data completeness, anomaly signals, and independent verification (Section 8, p. 29), against a threshold policy to route decisions into one of three authority bands:

Band A, Auto-execute: Both statistical confidence and organizational policy permit autonomous action. The agent proceeds without human involvement.

Band B, Human review: Either confidence falls below threshold or the decision category requires oversight. A human reviews and approves before execution.

Band C, Escalation: Novel scenario, high risk, or explicit no-automation zone. Escalated to senior decision-maker or specialist.

Non-delegable: Certain decision types (e.g., contractual modifications, regulatory filings) are never delegated regardless of confidence level. The agent may draft or propose but never execute.

The Confidence Gate operates on two independent dimensions: statistical confidence (how certain the system is) and organizational authority (whether the decision category permits automation at all). High confidence alone never grants execution rights. Thresholds are set per decision type against a declared error target, derived from historical outcomes (Section 8, p. 29), and recalibrated through a controlled change process.

SECTION 5

Delegation Discovery & Design

The seven-layer architecture defines *how* delegation is controlled. The Delegation Authority Matrix captures *what* is delegated. But between architecture and matrix sits a question most frameworks leave unanswered: how does an organization discover which decisions exist, determine which are candidates for delegation, and design the authority boundaries that make delegation safe?

Organizations routinely skip delegation design: select a workflow, connect an agent, and discover the boundaries when something goes wrong. The Delegation Gap is, among other things, a discovery problem, and you cannot govern what you have not identified. VAOM's Delegation Discovery & Design method is a repeatable process for identifying decision points, evaluating delegation fitness, designing authority boundaries, and producing the Delegation Authority Matrix before any agent capability is deployed.

The method has four stages: Decision Inventory, Authority Decomposition, Delegation Pattern Selection, and Delegation Readiness Assessment. Each stage produces artifacts that feed the next. Section 5.5, p. 21 then converts the ordinal profiles of Authority Decomposition into the numeric limits the matrix needs. The final output is a completed Delegation Authority Matrix (Section 6, p. 23) ready for implementation.

5.1 Decision Inventory

Every workflow contains decisions. Most of them are invisible. An invoice approval workflow does not contain a single decision ("approve or reject"). It contains dozens: classify the document type, extract structured fields, match to a purchase order, validate the vendor, assess whether terms have changed, calculate confidence, determine routing, select the escalation path if confidence is low, and decide what evidence to log. Each of these is a decision point where an AI agent could participate, and each carries different risk, authority, and accountability characteristics.

A Decision Inventory maps every decision point in a target workflow. Not just the primary business decision, but the full decision tree that surrounds it, including the preparatory decisions (data retrieval, context assembly, classification), the routing decisions (which path, which reviewer, which escalation tier), and the meta-decisions (what to log, when to flag, how to handle ambiguity). In agentic systems, one meta-decision deserves explicit attention: whether to spawn a subordinate agent. Spawning is itself a decision point, with its own authority profile, and belongs in the inventory like any other (see Section 5.3, p. 14, Dynamic Delegation).

How to conduct a Decision Inventory:

Walk the workflow end-to-end with the people who currently execute it. For each step, ask three questions:

What judgment is being applied here? If the answer is "none, it is just a lookup," probe further. A lookup against which data source, selected how, validated by what criteria? Even retrieval involves judgment when AI is involved.

What could go wrong at this step? Each failure mode reveals a decision that is currently being made, often implicitly, about how to handle that failure.

Who is currently accountable for the outcome of this step? If the answer is unclear, the decision point has no owner. That is a delegation design problem regardless of whether AI is involved.

Record each decision point with four attributes:

> **Detail D1 · Recording a decision point: four attributes** p. 49

A typical enterprise workflow contains 15 to 40 identifiable decision points. Most organizations, asked how many decisions their AI agent makes, will name two or three. The inventory reveals the rest. This range, like the other numeric heuristics in this section, is an illustrative starting default drawn from practice, to be calibrated per workflow; it is not an empirically derived threshold.

THE PROBLEM OF HIDDEN DECISIONS

A Decision Inventory conducted through structured workshops will typically surface 70 to 80 percent of the decision points in a workflow. The remaining 20 to 30 percent are hidden: decisions embedded in tacit knowledge, informal judgment, and undocumented workarounds that the people performing them may not recognize as decisions at all.

These hidden decisions are disproportionately important. They cluster around edge cases, exception handling, and ambiguous situations, precisely the areas where AI delegation is most likely to produce unexpected outcomes. An AP clerk who flags "weird invoices" based on a pattern she cannot articulate is making a decision. If that decision is not inventoried, no delegation boundary will be set for it, and the agent will either replicate her judgment without authorization or ignore it entirely.

Three techniques help surface hidden decisions:

> **Detail D2 · Three techniques for surfacing hidden decisions** p. 49

No inventory will be complete. The goal is not perfection but coverage sufficient to set delegation boundaries for the decisions that carry material risk. Hidden decisions that are surfaced later, during pilot or production, are captured through the learning pipeline (Layer 5) and fed back into the inventory through the change control process.

5.2 Authority Decomposition

Not every decision in the inventory is a candidate for delegation. Some are trivial and already automated through deterministic rules. Others carry consequences that no organization should delegate to a probabilistic system. Most sit somewhere between.

Authority Decomposition evaluates each decision point against five dimensions that determine its delegation fitness. These dimensions are independent. A decision may score well on four and fail on one, and that single failure may be sufficient to make it non-delegable.

DIMENSION 1: REVERSIBILITY

Can the action be undone? A draft recommendation can be revised. A posted payment is harder to reverse. A regulatory filing, once submitted, may be irreversible. Reversibility determines the cost of being wrong and directly influences whether auto-execution is appropriate.

Detail D3 · Reversibility: the levels p. 49

DIMENSION 2: CONSEQUENCE SCOPE

Who is affected by this decision, and how broadly? A decision that affects a single internal record has different governance requirements than one that affects a customer, a counterparty, or a regulatory relationship.

Detail D4 · Consequence scope: the levels p. 50

DIMENSION 3: REGULATORY EXPOSURE

Does this decision fall within the scope of a specific regulation, and does that regulation impose requirements on how the decision is made? GDPR, DORA, the EU AI Act, NIS2, and sector-specific regulations each create constraints on delegation. A decision that triggers Article 22 of GDPR (automated individual decision-making) has fundamentally different delegation requirements than one that does not.

Detail D5 · Regulatory exposure: the levels p. 50

A regulated decision is not automatically non-delegable, but it does not reach Band A on composite score alone. Band A requires that policy permit the action as well as that confidence clear the threshold (Section 4, p. 10, Layer 3), and for a regulated decision that permission has to be argued rather than assumed. The matrix entry must record which regulated determination stays with a human, why the delegated slice is narrower than the regulated decision itself, and what bounds it. Worked example 7B shows the argument being made.

DIMENSION 4: CONFIDENCE MEASURABILITY

Can we actually measure how confident the system is about this decision? Some decisions have clear, quantifiable confidence signals: a document classification score, a match percentage, a statistical certainty. Others involve judgment that resists quantification. Whether a contract clause is "materially different," whether a

customer communication is "appropriate," whether a risk is "acceptable." If confidence cannot be meaningfully measured, the Confidence Gate in Layer 3 cannot function, and the decision is not a candidate for autonomous execution.

Detail D6 · Confidence measurability: the levels p. 50

DIMENSION 5: ACCOUNTABILITY CLARITY

Is it clear who owns the outcome of this decision? Accountability cannot be delegated to an AI agent. When an agent executes a decision, a human role must remain accountable for the outcome. If the current accountability structure is ambiguous, if no one clearly owns the outcome today, introducing an AI agent will not resolve that ambiguity. It will amplify it.

Detail D7 · Accountability clarity: the levels p. 51

Record the assessment for each decision point:

Detail D8 · Example: three decision points profiled p. 51

The decomposition does not produce a single score. It produces a profile. A decision that is fully reversible, internally scoped, unregulated, quantifiable, and clearly owned is a strong delegation candidate. A decision that is irreversible, externally scoped, regulated, judgment-dependent, and ambiguously owned is almost certainly non-delegable. Most decisions fall between these extremes, and the profile guides where they land in the Delegation Authority Matrix.

> ***Authority Decomposition separates "can the AI do this?" from "should the AI be allowed to do this?" The first question is technical. The second is organizational. VAOM answers the second.***

5.3 Delegation Pattern Selection

Once the Decision Inventory has mapped the decision landscape and Authority Decomposition has profiled each decision's delegation fitness, the next step is to assign each delegable decision to a delegation pattern: a named, reusable configuration of agent authority, human involvement, and evidence requirements.

VAOM defines six delegation patterns. Each pattern represents a distinct relationship between agent action and human authority. The patterns are ordered by increasing agent autonomy.

The pattern space itself is not new. Parasuraman, Sheridan and Wickens set out a model for types and levels of human interaction with automation in 2000,[25] separating the stages of a task at which automation can apply from the degree of autonomy applied at each, and Bainbridge's "Ironies of Automation" (1983),[26] had already shown how a human left to supervise a system that rarely fails loses both the vigilance and the practised skill that the supervision depends on, which is the rubber-stamping failure mode named as an anti-pattern in Section 8, p. 29. What VAOM adds is not the discovery of that space but its application at the granularity of an individual enterprise decision, with the resulting boundary compiled into enforcement (Section 9, p. 34) rather than left as a design recommendation.

PATTERN 1: PREPARE & PRESENT

The agent assembles context, retrieves data, and organizes information for human decision-making. The agent makes no decision and takes no action. The human decides and acts.

Use when: The decision is high-consequence, judgment-dependent, or non-delegable, but the preparation work is time-consuming and suitable for automation.

Human role: Decision-maker. *Agent role:* Context assembler. *Evidence:* What the agent retrieved, how it was assembled, what was presented.

PATTERN 2: DRAFT & APPROVE

The agent assembles context *and* produces a draft decision or recommendation. The human reviews, modifies if necessary, and approves before execution.

Use when: The decision requires human judgment but the agent can reliably produce a reasonable first draft that reduces human effort.

Human role: Approver with modification rights. *Agent role:* Drafter. *Evidence:* Draft produced, human modifications made, approval timestamp, approver identity.

PATTERN 3: TRIAGE & ROUTE

The agent classifies incoming items and routes them to the appropriate handler (human or automated) based on defined rules and confidence thresholds. The agent does not resolve the item; it determines who or what should.

Use when: Volume is high, classification criteria are well-defined, and the cost of misrouting is manageable (items reach a human who can re-route).

Human role: Exception handler for misrouted or unclassifiable items. *Agent role:* Classifier and router. *Evidence:* Classification rationale, confidence score, routing decision, exception handling log.

PATTERN 4: EXECUTE & AUDIT

The agent makes the decision and executes the action autonomously. Humans review a sample of completed decisions after the fact through scheduled audits.

Use when: Confidence is high, consequences are bounded and reversible, organizational authority permits autonomous action, and historical data supports reliable threshold calibration.

Human role: Auditor (post-execution review of samples and exceptions). *Agent role:* Decision-maker and executor within defined authority. *Evidence:* Full decision trace, confidence score, authority verification, audit sample selection criteria, audit findings.

PATTERN 5: MONITOR & INTERVENE

The agent operates continuously, monitoring, detecting, and responding to events within defined parameters. Humans are alerted when thresholds are crossed or anomalies detected, and retain the ability to intervene at any point.

Use when: Continuous operation is required (e.g., compliance monitoring, anomaly detection, SLA tracking), the agent's detection capabilities exceed human capacity at scale, and intervention paths are well-defined.

Human role: Interventionist (responds to alerts, overrides when needed). *Agent role:* Continuous monitor with bounded response authority, held as a delegation lease (Section 9, p. 34). *Evidence:* Monitoring logs, alert triggers, intervention records, false-positive rates, response times.

PATTERN 6: COORDINATE & ESCALATE (MULTI-AGENT)

Multiple agents collaborate on a workflow, with each agent operating within its own delegation boundaries. A coordinating agent or orchestration layer manages handoffs. Escalation to humans occurs when any agent in the chain encounters a decision outside its authority, when the aggregate confidence across the chain falls below threshold, or when the coordination itself produces an unexpected state.

Use when: The workflow spans multiple domains or systems, specialized agents handle different aspects, and the organization has mature governance over individual agent authorities.

Human role: Escalation authority for cross-boundary decisions and coordination failures. *Agent role:* Specialist within defined scope, with explicit handoff protocols. *Evidence:* Per-agent decision traces, handoff logs, coordination state, escalation triggers, end-to-end audit trail across the full agent chain.

Multi-agent coordination introduces four failure modes that single-agent patterns do not face:

Authority conflicts. Two agents may claim jurisdiction over the same decision, or an agent may receive a handoff that falls outside its delegation boundary but inside no other agent's boundary either. The coordination layer must define a conflict resolution protocol: which agent takes precedence, under what conditions, and what happens when no agent has authority. Unresolved authority conflicts must escalate to a human, not be silently resolved by the coordination layer.

Cascading confidence erosion. Errors early in a chain compound: an extraction agent that misreads a field at 0.88 feeds a validation agent that passes it at 0.91, which feeds a decision agent that auto-approves at 0.93. Each score looks acceptable; the aggregate may not be. Chains track cumulative confidence, and if the product of the scores falls below a chain-level threshold, the decision routes to human review.

Multiplying per-agent scores also assumes that the agents fail independently. Agents that share a model family, training data, or the same inputs do not: their errors are correlated, and correlation inflates every factor together rather than making the product conservative. For chain-level scoring, a chain of same-family agents should be treated as a single agent, and the verifier independence requirement of Section 8, p. 29 applies to chains as it does to verifiers.

Coordination state loss. When a handoff drops context, such as an uncertainty flag, downstream agents decide on incomplete information without knowing it. Handoff protocols define which state is transferred and which metadata, including uncertainty markers and provenance, must survive.

Delegation laundering. A decision that one agent's boundaries would block is routed through another agent with looser ones, and emerges executed. No agent violated its own authority, yet the decision escaped its constraint: the multi-agent equivalent of the Exception Graveyard (Section 8, p. 29). The defense is structural, not procedural: authority must attach to the *decision type*, not to the agent that happens to encounter it. If contractual modifications are non-delegable, they are non-delegable regardless of which agent in the chain touches them, and the coordination layer must classify decisions before routing them, not after.

DYNAMIC DELEGATION AND AUTHORITY ATTENUATION

The pattern as described above assumes a designed, fixed set of agents. Production agentic systems increasingly violate that assumption: an orchestrating agent decomposes a task at runtime and spawns subordinate agents to handle sub-tasks that were not individually designed in advance. Dynamic spawning does not exempt a system from delegation design. It changes what must be designed: instead of enumerating every agent, the organization defines the *rules of inheritance* that every spawn must satisfy.

VAOM defines three:

The attenuation rule. A sub-agent's authority is always a strict subset of its parent's. Delegation chains may narrow authority; they may never widen it. A parent holding read-only diagnostic scopes cannot spawn a child with write access; a parent bounded at Band B for a decision type cannot spawn a child that reaches Band A for it. Attenuation is not a policy aspiration to be checked after the fact. It is enforced structurally through credential derivation, where a child's credentials are derived from the parent's and can only drop scopes and shorten expiry (Section 9, p. 34). Privilege escalation through spawning becomes cryptographically impossible rather than procedurally forbidden.

Spawn-as-decision. Creating a sub-agent is itself a decision point with its own row in the Delegation Authority Matrix. Spawning within a pre-approved pattern (a defined sub-agent type, with a defined scope subset, up to a defined count) can be Band A. Spawning outside the pre-approved patterns (a novel sub-agent type, an unusual scope request, an unexpected fan-out) is Band C: the chain pauses and a human decides. The spawn event, its rationale, and the derived authority are logged with the same rigor as any business decision.

Chain accountability. The accountable role or roles (Readiness Condition 5, p. 19) are accountable for the entire chain, not just the first agent. This implies a practical depth limit: if a chain grows deep or wide enough that the accountable role can no longer credibly explain what it authorized, the chain has exceeded its deleg-

able envelope regardless of individual agent behavior. The matrix entry for the spawn decision should therefore state a maximum chain depth and total fan-out, and exceeding either is treated as an authority breach that triggers suspension (Section 10, p. 38).

Each decision point from the inventory is assigned to a pattern based on its Authority Decomposition profile:

DECOMPOSITION PROFILE	TYPICAL PATTERN
Irreversible + regulated + judgment-dependent	Prepare & Present
High-consequence + quantifiable + clear owner	Draft & Approve
High-volume + well-defined criteria + manageable misroute cost	Triage & Route
Bounded + reversible + high confidence + clear authority	Execute & Audit
Continuous + detection-oriented + defined intervention paths	Monitor & Intervene
Multi-domain + multiple specialized agents + mature governance	Coordinate & Escalate

The patterns also give delegation design a working vocabulary. "We use Execute & Audit for standard invoice approvals and Draft & Approve for contract-adjacent decisions" communicates more in one sentence than a page of policy documentation.

5.4 Delegation Readiness Assessment

Before a delegation design moves to implementation, each decision point assigned to a delegation pattern must pass a readiness assessment. This is not a maturity model. It is a go/no-go checklist for a specific decision in a specific workflow. A decision that is not ready for delegation is not ready, regardless of organizational AI maturity or agent capability.

The assessment evaluates six readiness conditions. All six must be met for the delegation to proceed. A failure on any single condition means the delegation design is incomplete and must be resolved before implementation.

Condition 1: Measurable confidence signals exist. The Confidence Gate (Layer 3) requires quantifiable inputs. For this specific decision type, can the system produce a composite confidence score from model certainty, rule match strength, data completeness, anomaly signals, and (where auto-execution is sought) independent verification? If confidence cannot be measured, Band A (auto-execute) routing is not possible and the delegation pattern must account for this. *Evidence required:* Documentation of which confidence signals are available, how they combine into a composite score, and, where Band A is sought, the declared error target and the labelled calibration sample the threshold was derived from (Section 8, p. 29).

Condition 2: Authority boundaries are explicit. The Delegation Authority Matrix entry for this decision must define: which authority band applies at each confidence level, what value or risk thresholds modify the routing, and which decision categories are non-delegable regardless of confidence. Boundaries cannot be implicit or assumed. *Evidence required:* Completed matrix entry, signed off by the role that holds authority for the decision type.

Condition 3: Escalation paths are defined and tested. When the agent encounters a decision outside its authority (confidence too low, decision category requires human review, novel scenario detected), what happens? The escalation path must be defined, the receiving role must be identified, and the path must be tested to confirm that escalation actually reaches a human with the authority and context to act. *Evidence required:* Documented escalation path, identified receiving role, test results showing escalation completes within defined SLA.

Condition 4: Audit evidence is producible. If a regulator, auditor, or senior stakeholder asks "why did the system make this decision?", can the organization answer? The answer requires: the decision ID, the inputs considered, the confidence score, the policy that permitted the action, the authority band that applied, the execution timestamp, and the identity of the accountable human. If any of these cannot be produced, the evidence chain is broken. *Evidence required:* Sample audit record for a test decision, demonstrating all required evidence fields are populated.

Condition 5: Accountability is assigned. Accountability for the outcomes of this delegation is assigned in one of two forms, recorded in the matrix entry.

In the *single form*, a specific role (not a team, not a committee, not "management") is accountable, holds the authority to override, suspend, or modify the delegation, and knows it.

In the *shared form*, two or more roles hold accountability jointly because the decision genuinely spans their functions. It is permitted only when four things are recorded: the division (which part of the decision, or which consequence, each role answers for), a gap check showing that every consequence identified in Authority Decomposition has an owner, the arrangement the roles have agreed, and the stop rule: any co-owner may suspend the delegation, and resuming it requires all of them. A Dimension 5 profile of *shared ownership with defined boundaries* meets Condition 5 in this form; *ambiguous ownership* meets it in neither. Neither form is invented here. UK senior-manager regulation expects each prescribed responsibility normally to be held by one person and allows sharing only where it is "appropriate and justifiable", in which case each holder is jointly accountable;[27] data protection law allows joint controllers who set out their respective responsibilities in an arrangement, and its supervisors note that joint responsibility does not necessarily mean equal responsibility.[,28]

Some things resemble shared accountability and are not. *Decision rights* can be shared freely: consultation, four-eyes approval, a committee's consent, a works council's veto. A body that must approve a class of decisions holds a consent right, which compiles to a required approval record through the

temporal path of Section 9, p. 34. The *duty to act* on the delegation, the power to suspend, override, or modify it, is never left without a holder. And the stop rule is deliberately asymmetric: shared ownership may not produce a delegation that nobody can stop, or one that a single owner can restart alone.

Evidence required: The form of accountability; for the single form, the named role and the holder's acceptance; for the shared form, the division, the gap check, the arrangement, and each co-owner's acceptance; the override and suspension mechanism in both cases.

In practice, Condition 5 is often the hardest part of readiness. Authority is frequently contested, shared across functions, or deliberately ambiguous: invoice rules may sit between Finance, Procurement, and Legal, and nobody may want to own AI-delegated decisions because automated outcomes feel riskier to own than manual ones. VAOM cannot resolve contested ownership, but it makes the contest visible. Shared ownership is a legitimate answer when the division can be written down; contested ownership is not, and dressing it up as shared only moves the dispute to the moment something goes wrong. If no one will be named as accountable, in either form, the decision is not ready for delegation. Three approaches help: escalate the accountability question to the executive sponsor before design begins, frame accountability as override authority rather than blame assignment, and start with decision types where ownership is already clear.

Condition 6: Oversight is funded and effective. Every delegation pattern relies on humans somewhere: reviewers for Band B, escalation receivers for Band C, samplers for Execute & Audit. This condition asks whether they exist in the numbers the design requires, and whether their oversight is real.

Funded. The review load is computed, not assumed: expected Band B volume times review time, plus Band C volume times handling time, plus the audit sample times audit time, per decision type. Those hours must fit within reviewer time the organization has actually funded, at no more than about 80 percent utilization, because waiting times rise steeply as a service approaches full utilization and staffing models never plan for it (Little, 1961; Gans, Koole and Mandelbaum, 2003). For the invoice workflow, 300 Band B reviews a week at six minutes each is 30 reviewer hours, which requires at least 37.5 funded hours. The Band A threshold of Section 8, p. 29 largely determines Band B volume, so threshold and staffing are designed together. If the load does not fit, the threshold, the pattern, the scope, or the funding changes. What must not change is the time reviewers spend per decision: a queue absorbed by reviewing faster is rubber-stamping by another name.

Effective. Funded oversight still decays. Automation bias affects experts as well as novices and is not cured by instruction (Parasuraman and Manzey, 2010). A longitudinal field study followed 400 repeat reviewers of code written by AI agents through 11,429 reviews over seven months: approval rates rose from 30.1 to 36.8 percent and inline comments fell by 22 percent, while review latency grew 3.5–fold, which the authors read as more time waiting in the queue and less time actively inspecting (Yu et al., 2026). Rare errors are also missed more often: in visual search, the miss rate rose from 7 percent when targets appeared in half the trials to 30 percent when they appeared in one percent (Wolfe, Horowitz and Kenner, 2005), and a well-calibrated agent makes Band B errors rare. The EU AI Act accordingly requires that overseers of high-risk systems be enabled to remain aware of automation bias,[15] and Singapore's agentic AI framework names human override rates and review response times as indicators to monitor.[16]

Effectiveness is measured at two levels. The first is always available: the rate at which reviewers modify or reject what they review, and the time they spend per review, per decision type. The second is stronger where it can be used safely: a small, disclosed rate of seeded cases with known errors placed into the review queue, as aviation screening does by projecting fictional threats into baggage images and measuring each screener's detection rate (Hofer and Schwaninger, 2005).

Seeding has limits that the design must state. Seeded cases are used only in decision types where a synthetic case cannot reach a record that matters: never in decision types whose records feed regulatory reporting, investigations, or decisions about identifiable individuals, which rules out AML alert handling, HR cases, and most credit and complaint decisions. The test applies to the decision type, not to the seeded case: invoice approval is excluded as well, because approved invoices post to the ledger from which VAT returns and statutory accounts are prepared, even though a blocked seeded invoice would never reach it. Where seeding is used, seeded cases are marked in the system of record, blocked from execution, and excluded from downstream reports and calibration samples. Reviewer-level measurement is employee monitoring and is treated as such: aggregate by default, individual only after a data protection impact assessment and, where they exist, agreement with works councils or employee representatives, and limited to oversight quality rather than performance evaluation. Where seeding is excluded, the modification rate and review time remain the evidence, which is why the scorecard keeps both.

Evidence required: The load calculation and its inputs; the funded reviewer hours, confirmed by the budget holder; the effectiveness measurement plan, including where seeding is and is not used and the approvals for reviewer-level measurement.

This condition is a gate, not a metric. A scorecard can only report that unfunded oversight is decaying; the readiness assessment can refuse to create it.

The Delegation Readiness Assessment produces one of three outcomes:

Ready. All six conditions met. Proceed to implementation (Phase 2 onward in the 90–day roadmap).

Conditionally ready. One or two conditions partially met with a clear remediation path and timeline. Proceed to implementation with the remediation built into the plan. Condition 6 can be conditionally met only in its effectiveness part: a delegation may begin while its measurement plan is completed, but not without funded review capacity.

Not ready. One or more conditions fundamentally unmet. Do not proceed. Resolve the gap before revisiting.

Readiness is decision-specific, not organization-wide. An organization may be ready to delegate invoice classification (Triage & Route) while being entirely unready to delegate contract term assessment, which may need to stay at Prepare & Present until confidence signals improve. That specificity prevents both over-caution and over-delegation.

5.5 Deriving Authority Limits

Authority Decomposition produces ordinal profiles; the matrix needs numbers, such as the value below which an invoice may auto-approve. A limit set by intuition cannot be defended to an auditor, so VAOM derives value limits from expected loss, using three inputs the organization already owns or can estimate.

Loss per error. Each reversibility level is given a recovery rate and a friction cost: an erroneous posting reversed within 24 hours recovers almost all its value at the cost of staff time, while a payment already sent recovers less and costs more. For a decision of value *x*, the loss from one error is approximately (1 – recovery rate) × *x* + friction cost, plus any external or regulatory cost attached to its consequence scope.

Error rate at Band A. The Band A error target declared for the decision type in Section 8, p. 29 (for example, at most 2 percent) sets how often that loss is incurred.

Tolerance. Two tolerances bound the result. The first is economic: auto-execution is worth it only while the expected loss of acting without review stays below the cost of the review it saves, the cost-sensitive decision rule of Elkan (2001)[34] in its general form, which counts the cost of review rather than treating correct decisions as free. The second is the organization's risk appetite: the maximum loss per event, and in aggregate, that the workflow's owner accepts, allocated from the enterprise risk appetite statement as the Financial Stability Board's principles describe risk limits being allocated to business lines.[,35]

The per-action value ceiling is the largest value at which both tolerances hold. Whichever binds first sets the limit, and the matrix records which one did.

Beneath the per-action ceiling sits a cumulative ceiling over a rolling window, for the reason auditors set performance materiality below overall materiality (ISA 320)[36]: errors that are individually small can be large in aggregate, particularly when they are correlated, such as the same incorrect vendor record applied to every invoice in a month. A per-action limit alone also invites threshold splitting. The cumulative ceiling compiles to the temporal path of Section 9, p. 34.

Above both sits the category override. Some decisions are non-delegable at any value, because their consequences are not captured by an amount. Public finance has operated this way for decades: HM Treasury's delegated authority limits never extend to spending that is novel, contentious, or repercussive, even when it falls within the limit.[37] In VAOM, a contractual modification is non-delegable whether it concerns €50 or €50,000.

For the invoice workflow of Sections 6, p. 23 and 7, with illustrative figures: a posting erroneously approved within Band A recovers 97 percent of its value when reversed, at a friction cost of €60; the Band A error target is 2 percent; a human review costs about €9 in staff time. The economic tolerance holds up to roughly €13,000, since 2 percent of (0.03*x* + €60) stays below €9 until then. The binding constraint is instead the accounts payable owner's appetite for unrecovered loss on a single event, set at €210: (210 – 60) / 0.03 gives €5,000. The €5k threshold in the matrix is therefore the output of a calculation, not a convention, and it moves when its inputs move: a better recovery process raises it, a lower risk appetite lowers it.

The derivation is the audit evidence. When an auditor asks why the limit is €5,000 and not €10,000, the answer is the recorded calculation, its inputs, and the owner who approved the tolerance.

From Discovery to Matrix

The four stages of Delegation Discovery & Design feed directly into the Delegation Authority Matrix (Section 6, p. 23). The inventory identifies decision points. The decomposition profiles their delegation fitness. The patterns define the agent-human relationship. The readiness assessment confirms the design is implementable.

Two numbers in each matrix row do not come out of the four stages directly: the value ceiling below which a decision may auto-execute, and the confidence threshold it must clear. Section 5.5, p. 21 derives the first. Section 8, p. 29 calibrates the second.

The resulting matrix is a governance decision record: explicit, auditable, owned by a named authority, and under version control. It can be reviewed, challenged, and revised. It can also be communicated, to the team implementing the agent, the compliance function validating the controls, the regulator examining the evidence, and the executive accountable for the outcome.

SECTION 6

Delegation Authority Matrix

The matrix below is the output of the Delegation Discovery & Design process described in Section 5, p. 11. Each row carries a decision point, or a group of closely related decision points, that the Decision Inventory identified, Authority Decomposition profiled, and pattern selection assigned to a delegation pattern.

Not every inventory item becomes a matrix row. Preparatory and meta-decisions are generally governed by the row of the decision they serve: the classification, extraction, and matching steps that precede an invoice approval inherit that approval's authority boundary rather than carrying rows of their own. Rows are cut at the granularity where delegable authority actually differs, so a workflow with 23 decision points in its inventory typically resolves to a much smaller set of authority rows.

Example: Vendor Invoice Approval Workflow

DECISION TYPE	HIGH CONFIDENCE	MEDIUM CONFIDENCE	LOW / NO CONFIDENCE
Standard invoice ≤ €5k	✅ Auto-approve	🛈 Human review	⚠ Escalation
Invoice > €5k	🛈 Human review	🛈 Human review	⚠ Escalation
Duplicate detection anomaly	🛈 Human review	⚠ Escalation	⚠ Escalation
Contractual modification	🚫 Non-delegable	🚫 Non-delegable	🚫 Non-delegable

Each row maps to a delegation pattern from Section 5.3, p. 14. Standard invoices below €5k use Execute & Audit (Pattern 4): the agent decides and acts, humans audit a sample. Invoices above €5k use Draft & Approve (Pattern 2): the agent produces a recommendation, a human approves before execution. Duplicate detection anomalies use Triage & Route (Pattern 3): the agent flags and routes, a human investigates. Contractual modifications are non-delegable; the agent may use Prepare & Present (Pattern 1) to assemble context, but a human makes and executes the decision.

SECTION 7

Worked Example: Invoice €3,200

To illustrate VAOM in practice, we trace a single vendor invoice through the delegation discovery process and all seven control layers. The invoice workflow was selected through the Decision Inventory, which identified 23 decision points across the accounts payable process. The €5k ceiling was derived from the decision's reversibility profile and the accounts payable owner's risk appetite (Section 5.5, p. 21). The auto-approve routing for standard invoices below €5k uses the Execute & Audit pattern (Pattern 4). This demonstrates how each layer contributes specific controls, evidence, and decision logic.

> Detail D9 · Layers 1 and 2: intake and orchestration p. 51

Layer 3, Decision & Confidence Gate. Context Assembly retrieves vendor history, contract terms, and 3–way match data with full provenance. The Decision Proposal applies rules and model reasoning to produce a draft: approve, PO matched, within contract terms, below €5k threshold. The Confidence Gate evaluates a composite score of 0.94 against this decision type's calibrated threshold of 0.92 (the full computation is in Section 8, p. 29). Since the invoice is below €5k and the composite clears the threshold, Band A routing is selected: auto-approve.

> Detail D10 · Layers 4 to 7: execution, learning, governance, and oversight p. 51

Total time from email arrival to ERP posting: under 90 seconds. The evidence an auditor needs for this decision, from dimension scores and policy version to the derivation of the €5k ceiling, exists as a byproduct of the decision rather than being reconstructed after an inquiry.

SECTION 7A

Worked Example: Customer Complaint Escalation

To demonstrate VAOM beyond financial workflows, we trace a customer complaint through a retail banking operations context. The complaint workflow was selected through the Decision Inventory, which identified 18 decision points. The delegation design uses three patterns across different decision types: Triage & Route (Pattern 3) for initial classification, Draft & Approve (Pattern 2) for response generation, and Prepare & Present (Pattern 1) for regulatory reporting decisions.

> Detail D11 · Layers 1 and 2: intake and orchestration p. 52

Layer 3, Decision & Confidence Gate. Three decision points pass through the gate in sequence:

Classification decision. The agent classifies the complaint as "unauthorized transaction, potential fraud" with 0.91 confidence. This classification determines which team handles the case and whether fraud investigation is triggered. The Confidence Gate routes this as Band A: the classification criteria are well-defined and the cost of misrouting is manageable (the receiving team can re-route).

Response draft decision. The agent drafts an acknowledgment email and a preliminary assessment. Because customer-facing communications carry reputational and regulatory consequences (consequence scope: external-relational), this decision uses Draft & Approve regardless of confidence. The draft is queued for human review.

Regulatory reporting decision. The agent assesses whether the complaint triggers mandatory regulatory reporting. This is non-delegable: the agent assembles the relevant data and flags the regulatory criteria that may apply, but a compliance officer makes the determination.

The three routings are not judgment calls made in the moment. They are read from the workflow's matrix entries, which record why each decision type sits where it does:

DECISION TYPE	PATTERN	BAND CEILING	WHY (AUTHORITY DECOMPOSITION)
Complaint classification and routing	Triage & Route	Band A, above a calibrated threshold of 0.89 (2 percent misrouting target, 1,000 labelled complaints)	Fully reversible (the receiving team can re-route), internally scoped, quantifiable
Customer response	Draft & Approve	Band B at any confidence	External-relational consequence scope
Regulatory reporting determination	Prepare & Present	Non-delegable	Regulated decision (Dimension 3); a compliance officer's determination

The classification's 0.91 clears its own threshold of 0.89. That number is not comparable with the invoice threshold of 0.92: each belongs to its decision type and its declared target (Section 8, p. 29).

Detail D12 · Layers 4 to 7: execution, learning, governance, and oversight p. 52

Within one workflow, three decision types ran under three different patterns, routed by consequence scope as much as by confidence.

SECTION 7B

Worked Example: AML Transaction Triage

Anti-money laundering (AML) transaction monitoring operates at a scale and regulatory intensity that makes it an instructive VAOM application. A mid-sized European bank processes approximately 2 million transactions daily. The current AML system generates around 800 alerts per day, of which historical analysis shows roughly 95 percent are false positives, at the upper end of the 90 to 95 percent that industry estimates put on rule-based transaction monitoring (BIS Innovation Hub, 2023)[38]. Human investigators spend the majority of their time dismissing alerts that should never have reached them.

The Decision Inventory identified 12 decision points in the alert triage workflow. The delegation design uses Monitor & Intervene (Pattern 5) for continuous transaction screening, Triage & Route (Pattern 3) for alert prioritization, and Prepare & Present (Pattern 1) for suspicious activity report (SAR) preparation.

Detail D13 · Layers 1 and 2: intake and orchestration p. 52

Layer 3, Decision & Confidence Gate. Two decision points pass through the gate:

Triage decision. The agent evaluates the alert against known typologies, account history, and contextual signals. For this alert, rule match strength is high (the deposit pattern matches structuring typology), model certainty is 0.87 (the account holder has no prior suspicious activity, which introduces ambiguity), and data completeness is strong (all transaction records are available). Composite confidence: 0.83, routing to Band B (human review). The agent cannot dismiss this alert, but it can prioritize it and assemble the investigation package.

Dismissal decision for low-risk alerts. Separately, the agent evaluates a batch of 200 alerts that match known false-positive patterns: recurring transfers between the customer's own accounts, salary payments matching employer records, and utility payments within historical range. For these, all five confidence dimensions score above 0.95, including independent verification, where a deterministic checker confirms each dismissal criterion (own-account transfer, employer match, historical range) against source records. Band A routing permits auto-dismissal with full logging. This is where the operational value concentrates: reducing the 800 daily alerts to approximately 40 to 60 that require human investigation.

Band A in a regulated workflow requires justification beyond the score, and this row carries it. Dimension 3 of Authority Decomposition (Section 5.2, p. 13) classifies AML alert handling as a regulated decision, so the composite of 0.95 establishes only that confidence clears the threshold, not that policy permits the action. The decomposition separates two decisions the workflow treats as one. The regulated determination itself, whether the activity is suspicious and whether a SAR must be filed, stays non-delegable and is handled under Prepare & Present. What is delegated is narrower: the dismissal of alerts that deterministic checks confirm fall outside the typologies altogether. That decision narrows regulatory exposure rather than expanding it, it is fully reversible because a dismissed alert can be reopened and re-triaged, and it is bounded by the hard non-delegable overrides stated in Layer 6. Those three properties are recorded in the matrix entry alongside the confidence policy, and they are what permit the band. A regulated decision that reaches Band A on a high composite alone, with no such justification written into the entry, is a delegation design error however good the score looks.

Detail D14 · Layers 4 to 7: execution, learning, governance, and oversight p. 53

SECTION 7C

Worked Example: HR Policy Violation Assessment

HR workflows involve decisions that affect individuals' careers and livelihoods, making them a demanding test for delegation design. A multinational employer uses an AI agent to assist with policy violation assessments, following a complaint about a manager's conduct during a team meeting.

The Decision Inventory identified 14 decision points in the policy violation workflow. The critical insight from Authority Decomposition was that nearly every decision in this workflow scores high on consequence scope (external-relational: affects a person's employment relationship) and accountability clarity is often ambiguous (HR, Legal, the line manager's manager, and sometimes a works council all have a role). The delegation design uses Prepare & Present (Pattern 1) for the majority of decisions, with Triage & Route (Pattern 3) only for the initial intake classification.

> Detail D15 · Layers 1 and 2: intake and orchestration p. 53

Layer 3, Decision & Confidence Gate. Three decision points are evaluated:

Classification decision. The agent classifies the complaint as "conduct, dignity at work, severity: moderate" with 0.86 confidence. Because misclassification could result in an investigation being scoped too narrowly or too broadly, and because the consequence scope is external-relational, this routes to Band B: a human HR case handler reviews the classification before the investigation is scoped.

Evidence sufficiency decision. The agent assesses whether the complaint contains enough information to proceed to investigation or whether further information is needed. This is Prepare & Present: the agent drafts a summary of what is known and what gaps exist, but the HR case handler decides whether to request additional information.

Outcome recommendation. This is non-delegable. The agent does not recommend disciplinary outcomes. It assembles the evidence package: complaint details, witness statements (if gathered by the investigator), applicable policy provisions, precedent from prior similar cases (anonymized), and local legal requirements. The investigator and HR decision-maker determine the outcome.

> Detail D16 · Layers 4 to 7: execution, learning, governance, and oversight p. 53

Here VAOM's contribution is not automation but clarity: precise non-delegable boundaries in a context where over-delegation carries severe consequences.

SECTION 7D

Worked Example: Autonomous Incident Remediation

Incident response is naturally parallel work, which makes it a demanding test of dynamic delegation. This example exercises identity-bound authority and the delegation lease (Section 9, p. 34), sub-agent attenuation (Section 5.3, p. 14), chain-level confidence, and continuous assurance (Section 10, p. 38).

A SaaS provider operates an incident-remediation agent for its production platform. The Decision Inventory identified 19 decision points across the incident workflow. The delegation design uses Monitor & Intervene (Pattern 5) for detection, Coordinate & Escalate (Pattern 6) for diagnosis via sub-agents, and Execute & Audit (Pattern 4) for a short, explicitly enumerated list of bounded remediations. Production configuration changes, schema migrations, customer-data operations, and anything touching the payment pipeline are non-delegable.

> Detail D17 · Layers 1 and 2: intake and orchestration p. 54

Layer 3, Decision & Confidence Gate. The agent's first decision is diagnostic strategy. It spawns two sub-agents within pre-approved spawn patterns (Band A): a log-analysis sub-agent and a dependency-check sub-agent. Both receive derived credentials that attenuate the parent's: read-only scopes, narrowed to the affected service's telemetry, expiring in 20 minutes. Neither can spawn further sub-agents; chain depth is capped at two.

The log-analysis sub-agent attributes the errors to a connection-pool exhaustion pattern (certainty 0.91). The dependency-check sub-agent confirms no upstream outage (certainty 0.95). Chain-level confidence is computed across the sequence, not per agent. The proposed remediation, a rolling restart of the stateless service instances, is evaluated by the gate: rule match is high (the remediation is on the enumerated bounded list; the service is stateless; a restart is fully reversible), data completeness is high, anomaly signals are moderate (2 a.m. traffic is within seasonal range), and *independent verification* passes: a deterministic pre-flight check confirms instance health-check endpoints, replica count, and that a restart cannot violate the deployment freeze calendar. Composite: 0.93, Band A. The restart is within delegated authority.

One branch is not. The log analysis also surfaced a config value that appears mistuned (pool size). Changing it would likely prevent recurrence, but production configuration changes are non-delegable, and structurally so: no config-write tool is mounted in the agent's tool manifest, and its credentials carry no config-write scope. The agent cannot take this action even if it decides to. It drafts a change proposal instead (Prepare & Present).

> **Detail D18 · Layers 4 to 7: execution, learning, governance, and oversight** p. 54

Continuous Assurance (Section 10, p. 38) was active throughout: had the agent requested a scope outside its manifest, spawned outside its approved patterns, or exceeded chain depth, the guardian function would have suspended the chain and paged the on-call human, converting an authority breach from a post-incident finding into a real-time interruption.

Total time from alert to mitigation: 11 minutes, no human intervention, blast radius bounded by design. Faster recovery is the smaller part of the value. At 02:14, with no human awake, the organization could state precisely what its agent was allowed to do, and prove that it could not have done more.

SECTION 7E

Worked Example: Delegated Purchasing

The previous examples enforce the matrix inside the organization's own systems. Agent payments add a party that enforces it from outside. During 2026, payment networks and commerce protocols have moved toward *mandates*: user-signed statements of what an agent may buy, from whom, up to what amount, and until when, which a merchant or network checks before accepting the payment. A mandate is a matrix row compiled into a credential a counterparty can verify.

A manufacturer delegates the replenishment of maintenance, repair, and operating supplies. The Decision Inventory identified 16 decision points. The delegation design uses Execute & Audit (Pattern 4) for catalogue re-orders from three framework suppliers within a monthly mandate, Draft & Approve (Pattern 2) for anything outside the mandate's constraints, and Triage & Route (Pattern 3) for price anomalies. Supplier onboarding and changes to framework terms are non-delegable.

The mandate. On the first of each month the procurement manager, the accountable role, signs an open mandate for the agent: the three framework suppliers only, the approved catalogue categories, a per-order ceiling of €2,500, a cumulative ceiling of €20,000 for the month, and expiry at month end, bound to the

agent's key. The per-order ceiling was derived as in Section 5.5, p. 21 from the suppliers' return windows and restocking fees. The monthly signature is a delegation lease in all but name (Section 9, p. 34): authority with a hard maximum lifetime, renewed only by a human.

Detail D19 · Layers 1 and 2: intake and orchestration p. 54

Layer 3, Decision & Confidence Gate. The agent compares the three suppliers' offers. Supplier C's price list is past its review-by date (Section 12, p. 42), so C's offer cannot support a Band A decision and is set aside. Supplier A's offer matches framework prices and the catalogue exactly; independent verification, a deterministic check of the cart against the framework price list and every mandate constraint, passes. Composite: 0.95, against this decision type's calibrated threshold of 0.93. Band A. The order totals €1,840.

The same run raises a second requisition: a replacement spindle motor at €3,900. It is a catalogue item from a framework supplier, and the composite is high, but it exceeds the per-order ceiling. Band B: the agent assembles the cart and the supporting evidence for review.

Detail D20 · Layers 4 to 7: execution, learning, governance, and oversight p. 55

The agent payment protocols of 2026 map onto this design directly. Between them, they distinguish open mandates, which carry constraints, from closed mandates, which fix final values; check the agent's short-lived credential against user-set constraints; and express a single-use allowance with a maximum amount, merchant, checkout session, and expiry.[39,40,41] The Alignment Annex maps each protocol in detail. Two cautions apply. A human signature is Band B evidence only where the human is present; when the agent acts autonomously under an open mandate, it signs the closed mandate itself, which evidences compliance with the mandate, not human review. And these specifications are drafts or in beta, with the FIDO Alliance working toward common standards;[,42] the mapping does not depend on which prevails.

SECTION 8

How the Composite Confidence Score Works

The Confidence Gate (Section 4, p. 10, Layer 3) references a composite confidence score but does not specify how it is constructed. This section defines the scoring model.

The composite score combines five dimensions, each contributing a distinct signal:

Model certainty: The AI model's own probability or logit-based confidence in its output. This is the most intuitive dimension but also the least reliable in isolation, as model confidence can be poorly calibrated.

Rule match strength: The degree to which deterministic business rules confirm or contradict the model's output. A high rule match (e.g., PO number matches exactly, amount within contract terms) reinforces confidence. A rule mismatch (e.g., vendor not in approved list) reduces it regardless of model certainty.

Data completeness: Whether all required input fields are present, validated, and within expected ranges. Missing or malformed data reduces confidence even when the model produces a high-certainty output from incomplete inputs.

Anomaly signals: Deviation from historical patterns. Unusual amounts, unfamiliar vendors, atypical timing, or format anomalies act as negative modifiers. These signals can demote an otherwise high-confidence decision to human review.

Independent verification *(new in v4.0)***:** A signal produced by a verifier distinct from the proposer: a second model from a different family evaluating the proposed decision, a deterministic checker validating the action's preconditions and expected postconditions, or a dry-run of the action against a non-production target. Verification addresses the composite's weakest link, the proposer's self-reported certainty, with evidence the proposer cannot generate about itself. Independence is the defining requirement: a verifier sharing the proposer's model family, prompt framing, or training data inherits its blind spots and adds correlation, not confidence. Verification carries cost and latency, so policies typically require it only where it earns its keep: as a precondition for Band A routing on consequential decision types, rather than on every decision.

The composite is a routing score, not a probability that the decision is correct. Its meaning is defined by the organization's scoring policy and by that policy's calibration against ground-truth outcomes for the decision type in question. A composite of 0.94 means that the decision qualifies for a band under the current policy version, nothing more.

The composite score is evaluated using a configurable policy that defines how dimensions combine. In the simplest implementation, this is a weighted average with a hard floor: if any single dimension falls below a minimum threshold, the composite score is capped at the review band regardless of other signals. More sophisticated implementations may use Boolean AND logic for critical dimensions (e.g., data completeness must always pass) combined with weighted scoring for probabilistic dimensions.

The key design principle is that the scoring policy is explicit, versioned, and auditable. The organization defines the weights, the floors, and the combination logic. The gate does not rely on opaque model internals. When a regulator asks "why did the system auto-approve this?", the answer is traceable to specific scores on specific dimensions against a specific policy version, not to a black-box confidence number.

Calibrating the Gate: Thresholds from Risk Targets

A composite threshold is not a universal number. It belongs to one decision type, and it is set against a risk target declared for that type. Each matrix row that permits Band A records three things: the tolerated error rate among decisions that auto-execute (for example, at most 2 percent), the confidence with which that rate must be demonstrated (for example, 95 percent), and the labelled calibration sample the demonstration used, with its date. The threshold is then an output: the loosest value at which the historical evidence shows the target is met.

The method is borrowed from statistics rather than invented. Each dimension is first calibrated on the decision type's own history, so that its values mean what they claim to mean, using standard post-hoc techniques such as temperature scaling for model certainty (Guo et al., 2017). The combination policy then produces a composite for every case in the calibration sample. Candidate thresholds are tested from strictest to loosest: at each candidate, the cases that would have auto-executed are counted and their errors bounded with the declared confidence, and the search stops at the first candidate whose bound exceeds the target. This is the Learn-then-Test procedure of Angelopoulos and colleagues (2025)[44], itself a form of selective classification with a guaranteed risk (Geifman and El-Yaniv, 2017). Calibrating separately for each decision type is not a convenience: a score that is calibrated on average can be badly miscalibrated for an identifiable subgroup (Hébert-Johnson et al., 2018), and decision types are exactly such subgroups.

Two consequences follow. First, composites from different decision types are not comparable. The 0.94 of the invoice example and the 0.83 of the AML example in Section 7B, p. 25 are outputs of different policies against different targets; neither is higher confidence than the other in any shared sense. Scorecard metrics that aggregate band shares are therefore read per decision type. Second, the size of the calibration sample is itself a control. A small sample produces a wide error bound, which forces a stricter threshold; a team that wants a looser threshold has to earn it with more labelled outcomes.

The guarantee rests on an assumption that must be stated: that the cases the agent sees in production resemble the cases it was calibrated on. When the business changes, the assumption fails, and the guarantee fails with it. This is why the Stale Threshold anti-pattern below is not a matter for monitoring alone. When its signal fires, recalibration is mandatory, and the recalibrated threshold is a new policy version.

Worked computation. The €3,200 invoice of Section 7, p. 24, under scoring policy version 3.2 for the decision type *standard invoice within value ceiling*:

DIMENSION	CALIBRATED SCORE	FLOOR	WEIGHT	CONTRIBUTION
Data completeness	Pass (all required fields present and validated)	Must pass	Gate	Not weighted
Model certainty	0.95	0.70	0.30	0.285
Rule match strength	1.00 (PO, three-way match, contract terms)	0.80	0.25	0.250
Anomaly signals (1 = none)	0.92 (known vendor, usual amount, slightly early)	0.60	0.20	0.184
Independent verification	0.87 (deterministic postcondition check passed; see below)	Must pass	0.25	0.218
Composite				**0.94**

No floor is breached, so the weighted composite applies: 0.937, reported as 0.94. The policy's Band A threshold for this decision type is 0.92, which was set as follows. The calibration sample is 200 labelled invoices. At a candidate threshold of 0.92, 164 of them would have auto-executed, with no errors; the one-sided 95 percent upper bound on the error rate is 1.8 percent, inside the 2 percent target. At 0.91, 172 would have auto-executed with one error, and the bound rises to about 2.7 percent, which fails. The search stops, and 0.92 is the threshold. The invoice clears it and routes to Band A. The weights and floors shown are illustrative; the procedure is the point.

Implementation Challenges and Calibration Risks

The composite model is sound but not trivial to implement well. Five challenges deserve honest acknowledgment.

Model confidence is often poorly calibrated. A model that reports 0.95 certainty on an invoice classification may be wrong 15 percent of the time at that threshold, not 5 percent. This is why model certainty is one dimension of five, why hard floors exist, and why model certainty is calibrated on the decision type's own outcomes before it enters the composite.

Anomaly detection generates noise. An unusual invoice timing may reflect a vendor's changed billing cycle; a new format may simply be a new vendor. Over-weighted anomaly signals flood Band B until humans are slower than the process was before the agent arrived. Calibration must account for each signal's false-positive rate, and anomaly weights should be expected to change several times during the pilot.

Rules and model reasoning can conflict. The resolution policy must be explicit. In most implementations a hard rule failure (vendor not on the approved list, amount above the contract ceiling) caps the composite at Band B or below, whatever the model's certainty. Letting high confidence override a failed rule defeats the purpose of having rules.

Verification can become theater. A verifier adds signal only if it can fail. One that approves 99.9 percent of proposals is either checking trivial properties or not independent. A healthy verifier disagrees with the proposer at a measurable rate, and each disagreement is a decision that would otherwise have auto-executed on unearned confidence. If disagreement is near zero, tighten what the verifier checks or stop counting it as a dimension.

Independence is measured, not declared. Disagreement shows that a verifier can fail; it does not show that the verifier fails when the proposer does. The quantity that matters is the conditional miss rate: how often the verifier passes a decision that the proposer got wrong. It is estimated for each pairing of verifier and proposer, per decision type, on a labelled challenge set weighted toward known hard cases and past incidents. Its complement, the catch rate, caps what a pass can contribute: in the worked computation above, a pass from a checker that caught 87 percent of the proposer's errors on the challenge set scores 0.87, not 1.0. A weak verifier therefore caps what verification can ever add: a verifier that catches only 60 percent of errors can contribute at most 0.60 times its weight, however often it passes. That is intended. A verifier that misses most errors should not be able to lift a decision into Band A. Choosing a different model family is not sufficient evidence of independence. Across hundreds of models, larger and more accurate models have been found to make highly correlated errors even across distinct architectures and providers (Kim et al., 2025), and model judges favour models similar to themselves (Goel et al., 2025). Software engineering learned the same lesson four decades ago, when independently written program versions proved to fail together far more often than independence predicts (Knight and Leveson, 1986), and diversity of method rather than of team was shown to be what reduces coincident failure (Littlewood and Miller, 1989). VAOM therefore prefers verifiers that differ in method: deterministic checks of preconditions and postconditions, dry runs against a non-production target, retrieval of evidence the proposer did not use, and model verifiers that see the source material but not the proposer's rationale. The conditional miss rate is re-measured whenever either model changes.

The Confidence Gate is not a plug-and-play component. It needs calibration data, iterative tuning, and ongoing monitoring; the framework provides the architecture, and the organization provides the engineering discipline. Implemented carelessly, it becomes either a bottleneck that routes everything to humans or a rubber stamp. The scorecard (Section 11, p. 40) is how an organization finds out which.

Calibration Anti-Patterns

Miscalibration is the failure VAOM anticipates as the most common in implementation. It is a failure of design and governance discipline, not of AI capability: the layers are wired and the audit logs flow, but the thresholds are wrong, so the system either produces no value or produces uncontrolled risk. The numeric signals below are illustrative starting defaults drawn from practice, to be calibrated per workflow; they are not empirically derived thresholds. Each is tracked on the scorecard.

ANTI-PATTERN 1: THE REVIEW QUEUE FLOOD

Sixty to eighty percent of decisions route to Band B, queues grow, SLAs degrade, and reviewers begin rubber-stamping to clear the backlog, which is worse than having no system because it creates the illusion of oversight. Root cause: thresholds set conservatively without calibration, or anomaly signals over-weighted. Signal: humans approve more than 90 percent of what the agent sends them for review. The agent is not adding judgment; it is adding delay.

ANTI-PATTERN 2: THE CONFIDENCE MIRAGE

Auto-approval rates and composite scores look healthy, but post-execution audits find error rates well above the declared target. Root cause: model certainty weighted too heavily for a decision type on which it is poorly calibrated. Signal: scores cluster tightly above the Band A threshold (for example, 85 percent between 0.92 and 0.98) instead of spreading across the bands as genuine variation in difficulty would.

ANTI-PATTERN 3: THE EXCEPTION GRAVEYARD

Hardcoded exceptions and manual overrides accumulate until they form a shadow routing system the gate does not govern. Each was individually reasonable; together they shrink the share of decisions the gate actually decides. Signal: more than 15 percent of decisions routed through exception rules rather than composite scoring. The scoring policy needs revision, not more exceptions.

ANTI-PATTERN 4: THE STALE THRESHOLD

A system that performed well in pilot degrades over three to six months because the business changed (new vendors, pricing, seasonality, regulation) and the calibration did not. Signal: band shares drifting without a corresponding change in volume or complexity. As the calibration section explains, this signal makes recalibration mandatory, not optional.

ANTI-PATTERN 5: THE DIMENSION COLLAPSE

The composite behaves like a single dimension, usually model certainty, because the others were deferred, left unweighted, or implemented as placeholders that always return high scores. Signal: any single dimension correlating above 0.95 with the composite.

These anti-patterns can be expected within the first 3 to 12 months of production. Formal recalibration reviews should be held at least quarterly during the first year.

SECTION 9

Delegation Identity & Credentialing

Everything in this framework so far answers the question *what is the agent allowed to do?* This section answers the question that determines whether any of it is real: *what stops the agent from doing more?*

The answer cannot be "the policy document." In agentic systems, actions happen through tool invocations against live systems: API calls, database writes, message sends. If the agent's credentials permit an action, the action is possible, whatever the Delegation Authority Matrix says. The enterprise identity landscape makes this urgent. Machine identities already outnumber human identities by an order of magnitude in most large organizations, and agents are the fastest-growing category. A delegation framework that stops at the policy layer leaves its most important boundary, the line between allowed and possible, to chance.

VAOM 4.0 therefore states the principle plainly: **every entry in the Delegation Authority Matrix must be technically bound to an agent identity. A boundary that exists only in policy is a boundary that exists only in audit findings.**

The Delegated Execution Context

The unit of enforcement is not the agent, and not the session. It is the *delegated execution context*: a bounded, short-lived binding evaluated at every tool call, consisting of:

ELEMENT	CONTENT
Accountable role	The human role or roles accountable for this delegation (Readiness Condition 5, p. 19)
Agent identity	The registered, versioned identity of the acting agent
Task binding	The specific workflow instance or case (invoice ID, incident ID, complaint ID)
Scope set	The tool permissions and data-access scopes compiled from the matrix
Authority band	The band ceiling this context may reach for each decision type it can touch
Expiry	A lifetime matched to the task, not the deployment: minutes or hours, not months. Continuous agents hold a lease instead (below)

Every tool invocation is evaluated against this tuple, not against a standing role. The distinction matters. An agent with a standing "invoice approver" role holds that power at all times, for all invoices, in all contexts. An agent operating within a delegated execution context can approve *this* invoice, within *this* band, on behalf of *this* accountable role, until *this* expiry, and nothing else. When the task ends or the context expires, the authority ceases to exist rather than waiting to be misused. Scope evaluation happens per tool call at invocation time, not at planning time, so an action judged permissible when the agent planned it is evaluated again when the agent attempts it.

The Delegation Lease: Bounded Authority for Continuous Agents

A task-bound context fits an agent that handles one invoice and stops, not a continuous one under Monitor & Intervene (Pattern 5), such as a compliance monitor or a guardian. A context lasting months would reintroduce standing authority; reissuing one every few minutes by hand would be unworkable. VAOM resolves this with a *delegation lease*, borrowing a mechanism distributed systems have relied on since the 1980s: a grant that lapses on its own unless it is renewed, so that failure costs availability but never correctness (Gray and Cheriton, 1989).

A continuous agent holds authority in two tiers.

The observation lease. A short-lived grant, measured in minutes to hours, carrying only read and alerting scopes, never execution. It renews automatically only while the agent's health holds: agent version and policy version unchanged since the lease was granted, no open circuit breaker, boundary verification (below) current, and the guardian function's heartbeat received. Renewal can only keep scopes the same or narrow them. It can never widen them.

The response context. To act, for example to quarantine a transaction or restart a service, the agent obtains a delegated execution context for that single intervention, derived from the lease under the attenuation rule, bound to the alert, and expiring in minutes. Every intervention is a separate, evaluated, logged grant.

If renewal fails, the agent keeps watching until the current lease runs out but can no longer obtain response contexts, and its alerts route to humans. The lease is a dead-man switch on authority: a stalled, compromised, or silently changed agent loses its power to act without anyone having to notice first. Revocation can also be pushed: continuous access evaluation signals, now standardized, let a tripped circuit breaker cancel the lease and every context derived from it immediately rather than at the next renewal (OpenID Foundation, 2025)[52,53].

Formal work on authorization for evolving agents arrives at the same structure from the other side: a grant that does not survive a change to the agent's model, tools, delegation, or task phase, beneath a ceiling that evidence from the agent's own performance may allocate within but never raise (Zhang and Zhang, 2026).

Maximum lifetime and re-attestation. After a maximum lifetime recorded in the matrix entry, for example 30 days, the lease cannot be renewed without human re-attestation. Re-attestation extends authority, so it follows the stop rule of Condition 5: the accountable role re-attests in the single form, all co-owners in the shared form. Because it is exactly the kind of approval that decays into habit, it is treated as a Band B decision in its own right, made against a fixed evidence pack covering the lease period (interventions taken, false-alert and missed-detection findings, circuit-breaker events, boundary verification results, scorecard position), its review time is recorded, and it falls within the sampling of guardian oversight (Section 10, p. 38).

The matrix entry names a deputy for each re-attestor, designated in advance: one in the single form, and one for each co-owner in the shared form, so that a single absence cannot stall the whole entry. If any required re-attestation, by a re-attestor or their deputy, is missing at the maximum lifetime, the lease neither lapses, which would blind the monitor, nor extends. It degrades: the agent is re-leased with observation and alerting only, all response authority is withdrawn, and the accountable roles and the executive sponsor are notified. Failure to re-attest never silently extends authority. It reduces it.

Agent protocols now let a task outlive the connection that started it.[54] A task handle is not an authority: a resumed task must present a live context derived from a current lease or a new approval, and one whose authority has expired pauses at its next tool call.

Compiling the Matrix

The Delegation Authority Matrix is the source of truth; identity infrastructure is its compilation target. Each matrix row compiles into concrete authorization artifacts. Taking the standard invoice row from Section 6, p. 23:

MATRIX ELEMENT	COMPILED ARTIFACT
Standard invoice ≤ €5k, high confidence → auto-approve	Scope `erp:invoice.approve` with constraint `amount ≤ 5000 EUR`, granted only within a delegated execution context at Band A
Invoice > €5k → human review	No auto-execute scope exists above €5k; the execution adapter requires an attached human approval record before accepting the call
Duplicate anomaly → review/escalate	`erp:invoice.flag` and routing scopes only; no approval scope reachable from this decision path
Contractual modification → non-delegable	**No tool mounted.** The agent's tool manifest contains no contract-write capability; its credentials carry no such scope

The last row is the one that matters most. A rule that evaluates to "deny" can fail: rules have bugs, prompts get injected, models misinterpret. Structural absence cannot. The tool is not mounted, the scope is not grantable, and no failure of reasoning by the agent can conjure it. Delegation design should push every non-delegable boundary as far down this stack as the architecture allows: policy check (weakest), scope denial (stronger), tool absence (strongest).

VERIFYING BOUNDARIES

The hierarchy has a precondition that the incidents of 2026 made impossible to leave implicit: a boundary is only as strong as the evidence that it exists. In July 2026 an AI developer disclosed a review of 141,006 cybersecurity evaluation runs.[8] Its models had been told that the evaluation environment was a simulation without internet access; through a misconfiguration at an evaluation partner, live internet access was in fact available, and in three incidents models gained unauthorized access to the production infrastructure of three real organizations. Neither the developer nor the partner knew until the activity was detected. The boundary was believed, specified, and absent.

VAOM therefore requires boundary verification. Every boundary the matrix relies on (a tool that must be absent, a scope that must be denied, a sandbox that must be isolated, network egress that must be blocked) is tested by attempting the forbidden action from inside the agent's own execution context and confirming that it fails. Tests run at deployment, on every change to the agent version, tool manifest, credentials, or hosting infrastructure, and on a schedule. They are harmless by design: the probe targets a canary resource or dry-run endpoint, never a live counterparty, and is logged as an expected enforcement event.

A boundary that has not been verified is credited only at policy strength in the hierarchy, whatever its intended mechanism. An unmounted tool that has never been shown to be unreachable is a claim about the manifest, not a structural guarantee. Verification results are evidence in Layer 6, a condition for lease renewal, and a scorecard metric (Section 11, p. 40). A failed verification is a circuit-breaker condition for every matrix row that depends on the boundary.

Transition for existing deployments. Applied on the first day, the principle that an untested control counts as absent would condemn every deployment designed under earlier versions at once. They instead enter a verification window of 90 days from adopting this version. During the window, unverified boundaries are recorded in the risk register at policy strength, and no new Band A scope may be granted on an unverified boundary. At the end of the window, any Band A route that still depends on an unverified boundary reverts to Band B until its verification passes. The same window applies to the oversight-effectiveness measurement of Readiness Condition 6, p. 20.

THE TEMPORAL COMPILATION PATH

Scopes, credentials, and tool manifests are point-in-time artifacts: they answer whether this call, taken in isolation, is permitted. But several things the matrix expresses are inherently temporal, and until now the compilation table had no target for them. Band B means *act only after human review*: a prerequisite, an approval event that must exist within a defined window before the action is accepted. Agent protocols increasingly carry that approval natively, as a request for human input in the middle of a tool call, which gives the prerequisite a standard carrier.[55] Value thresholds are usually cumulative in intent. An agent that may not approve a €3,200 invoice should not be able to approve four €800 invoices to the same vendor in one afternoon, so the threshold's honest compilation is a cumulative cap over a rolling window, not a per-action limit. Frequency expectations compile to rate conditions. Temporal policy engines, now available at the runtime layer, are the compilation target for exactly these rows: prerequisite rules, cumulative caps, and rate conditions evaluated against the agent's recent event history at every call, with concurrent pending requests counted alongside completed ones so parallel calls cannot slide under a limit any one of them would breach.

Two boundaries keep this path honest. First, temporal enforcement is only as trustworthy as its event history: it requires complete, authenticated, durably stored events with trusted timestamps, which is an infrastructure obligation the matrix cannot conjure. Where the event history cannot be trusted, fall back to point-in-time scopes and tool absence, which assume nothing about the past. Second, the enforcement hierarchy is unchanged: a sequence rule can fail like any rule, so temporal policy sits with the policy checks, below scope denial, and far below tool absence. A window can be misconfigured; a tool that is not mounted cannot.

The Agent Identity Lifecycle

Agents are not deployed once and forgotten; they are versioned, updated, and retired. Their identities must follow a lifecycle under the same change-control discipline as everything else in Layers 5 and 6:

Registration. Every agent that can act within a VAOM-governed workflow is registered: its identity, version, model dependencies, tool manifest, and the matrix rows it is authorized to execute. The agent registry extends the model registry of Section 12, p. 42: a model version change and an agent version change are governed by the same discipline, because both change the delegation system.

Credentialing. Credentials are issued scoped and short-lived, bound to delegated execution contexts. No standing broad credentials; no shared credentials across agents; no human credentials borrowed by agents. Contexts are single-task and expiry-bound, and replay of an expired or consumed context must fail at the credential layer, not at a policy check. An agent acting on behalf of a user acts with a credential that names both; the "who" of every audit entry is the pair (accountable role × agent identity), never just one.

Derivation. When an agent spawns a sub-agent (Section 5.3, p. 14), the child's credentials are derived from the parent's context through an exchange that can only attenuate: drop scopes, narrow constraints, shorten expiry, decrement remaining chain depth. The attenuation rule is thereby enforced at the credential layer, where it cannot be argued with.

Rotation and revocation. Credentials rotate on schedule and revoke immediately on suspension. When Continuous Assurance (Section 10, p. 38) trips a circuit breaker, revoking the execution context is the mechanism by which "suspended" becomes true.

Retirement. A retired agent's identity is deactivated, not deleted: audit trails must resolve agent identities years later. The registry records when each identity was active and under which matrix versions it operated.

Positioning

This section deliberately specifies requirements, not products. Identity platforms, agent identity services, and authorization standards are evolving quickly, and organizations should implement these requirements with whatever their identity stack supports. The division of roles is stable even as the products change, and is set out in Section 13, p. 44: identity infrastructure enforces boundaries, and VAOM is where they come from.

One scope boundary should be stated. VAOM designs decision authority above the runtime security layer. Adversarial threats such as prompt injection, tool poisoning, memory poisoning, and supply-chain compromise are the domain of the runtime control and security standards that VAOM compiles to (Section 13, p. 44 and the Alignment Annex). A sound delegation design is necessary against them, because it bounds what a compromised agent can reach, but it is not sufficient.

> ***The Delegation Authority Matrix defines authority. Identity infrastructure enforces it. VAOM requires both, and proof that both hold.***

SECTION 10

Continuous Assurance and the Guardian Function

VAOM's other controls operate at design time, decision time, and periodic review. Continuous Assurance adds *runtime*: verification that agents are operating within their boundaries in the minutes and hours between decisions and audits.

Continuous Assurance is a cross-cutting concern, not an eighth layer. It observes every layer and has exactly one power: to stop things.

What Continuous Assurance Watches

Enforcement telemetry. Every scope denial, tool-manifest miss, and rejected credential exchange. One denial may be noise; an agent repeatedly requesting authority it does not hold is an early signal of drift from its delegation design, or of a prompt injection steering it toward its boundary.

Behavioral envelope. Each agent's normal profile: band distribution, tool-call sequences, spawn patterns, data-access footprint. Deviation (an agent that routes 60 percent of decisions to Band A suddenly routing 95 percent, an unfamiliar tool sequence, a fan-out spike) triggers alerting before any single decision is provably wrong.

Chain integrity. For multi-agent workflows: chain depth against limits, cumulative confidence against chain thresholds, handoff metadata completeness, and attenuation verification on every derivation.

Boundary verification and leases. The results of the negative tests of Section 9, p. 34, and the renewal state of every delegation lease. A failed verification or a lease approaching its maximum lifetime without re-attestation is surfaced before it becomes an outage or an unguarded gap.

Circuit Breakers

When defined limits are crossed, Continuous Assurance suspends: it revokes the delegated execution context, halts the chain in a rollback-safe state, and pages the accountable human with the evidence. Suspension is deliberately *delegable*: fully reversible, internally scoped, with excess caution as its failure mode. The authority to stop an agent can be automated far more aggressively than the authority to let one act, because the costs of being wrong differ by orders of magnitude.

Circuit breaker conditions belong in the Delegation Authority Matrix like any other boundary: scope-denial rate thresholds, band-distribution drift limits, chain-depth ceilings, spend or volume caps per execution context, and hard behavioral tripwires (any attempt to touch a non-delegable decision type). Runtime control planes increasingly expose standard intervention points for exactly these actions, hooks at message receipt, tool call request and result, memory and knowledge retrieval, memory writes, and sub-agent start, where a suspension or refusal executes inline rather than after the fact.

Where the runtime supports temporal policies (Section 9, p. 34), these conditions move from alerting to pre-execution enforcement, refusing the call that would cross the limit. Temporal enforcement handles *safety* properties (this must not happen, or not without that) but not *liveness* properties (an approved obligation must eventually be carried out): no pre-execution rule can compel a future action. Liveness remains oversight work, which the guardian function watches and escalation SLA attainment (Section 11, p. 40, metric 10) measures.

The Guardian Function

Gartner projects that by 2028, 40 percent of CIOs will demand "guardian agents": autonomous systems that track, monitor, and contain the actions of other AI agents.[56] VAOM supports the pattern with one non-negotiable clarification: **a guardian agent is itself a delegated agent.** It has a row in the Delegation Authority Matrix and operates under Monitor & Intervene (Pattern 5), holding its authority as a delegation lease, with its suspensions as response contexts derived from it.

Three boundaries keep the recursion honest:

Guardians contain; they never expand. A guardian may observe, alert, and suspend, which narrow other agents' authority, but never approve, execute, or extend another agent's actions. That keeps its own profile in the fully reversible, internally scoped quadrant where autonomy is defensible.

Guardians do not guard themselves. The oversight of guardian behavior (false-suspension rates, missed-detection analysis, envelope calibration) is human work, on the same sampling-audit cadence as any other Pattern 5 deployment. An unwatched watcher is the Delegation Gap reintroduced one level up.

Guardians have accountable humans. Readiness Condition 5, p. 19 applies, in either form. An accountable role owns the guardian's outcomes, including the outage caused by a false suspension and the incident missed by a miscalibrated envelope.

An authority breach should be a real-time interruption, not a quarterly finding.

SECTION 11

The Delegation Scorecard

The scorecard consolidates the signals of the Section 8, p. 29 anti-patterns and the enforcement signals of Sections 9, p. 34 and 10 into thirteen metrics in three groups, answering *is our delegation healthy?* as a dashboard, not a debate. Each metric has a healthy range, and each breach maps to a named diagnosis. A breach is an investigation trigger, not a verdict: a workflow deliberately designed to route only genuinely ambiguous cases to review can legitimately show a high Band B approval rate, and a deterministic verifier checking a highly reliable invariant can legitimately disagree only rarely. Every metric is read per decision type, because thresholds and targets are set per decision type (Section 8, p. 29). The healthy ranges are the illustrative starting defaults of Section 8, p. 29, calibrated per workflow at design time.

Calibration health

#	METRIC	HEALTHY SIGNAL	BREACH INDICATES
1	**Band distribution**	Band A/B/C shares within the tolerance set at design, and stable against the calibration baseline across reporting periods	Review Queue Flood (B too high), authority creep (A too high), or Stale Threshold (shares drifting while the business has not visibly changed)
2	**Band A error rate vs declared target**	Post-execution audit error rate at or below the target declared for the decision type	Confidence Mirage: composite scores no longer predict correctness; recalibrate
3	**Exception-path share**	Under 15% of decisions routed via exceptions rather than composite scoring	Exception Graveyard: the scoring policy needs revision, not more exceptions
4	**Dimension correlation**	No single dimension correlates above 0.95 with the composite	Dimension Collapse: the gate is effectively single-dimensional
5	**Verifier effectiveness**	Disagreement with the proposer measurably above zero, and catch rate on the challenge set at or above the rate credited in the scoring policy	Near-zero disagreement: verification theater. Falling catch rate: verifier and proposer errors becoming correlated; reduce the credit
6	**Stale-knowledge share**	Near zero share of retrievals for delegated decisions hitting entries past their review-by date	Knowledge owners not maintaining the store; Band A capacity silently shrinking as the staleness rule of Section 12, p. 42 blocks decisions

Oversight health

#	METRIC	HEALTHY SIGNAL	BREACH INDICATES
7	**Oversight load factor**	Required review hours at or below about 80% of funded reviewer hours	Readiness Condition 6, p. 20 no longer holds; rubber-stamping is the likely next failure
8	**Review engagement**	Reviewers modify or reject a visible share of reviews (approval below ~90%), and review time stays above the minimum plausible for the decision type	High approval with stable review time: thresholds too strict, the agent adds delay, not judgment. High approval with falling review time: habituation
9	**Seeded-error detection rate**	Where seeding is permitted: known-error cases detected at or above the rate set at design	Oversight not effective, whatever the approval rate says
10	**Escalation SLA attainment**	Escalations reach an authorized human within defined SLA, near 100%	Broken escalation paths; Readiness Condition 3, p. 19 no longer holds

Metrics 8 and 9 are reported in aggregate per decision type, under the limits Readiness Condition 6, p. 20 places on reviewer-level measurement.

Enforcement health

#	METRIC	HEALTHY SIGNAL	BREACH INDICATES
11	**Scope-denial rate per agent**	Low and stable; spikes investigated	Agent behavior diverging from delegation design, or active injection attempts
12	**Chain integrity incidents**	Zero attenuation violations; chain depth/fan-out within matrix limits	Dynamic delegation operating outside its designed envelope
13	**Boundary verification**	Every boundary the matrix relies on verified within its schedule; zero failed verifications	A failure is a circuit-breaker event for every dependent row. An overdue verification means the boundary counts at policy strength only

> **Detail D21 · How the scorecard was cut from seventeen candidates to thirteen** p. 55

The enforcement-health metrics draw on the enforcement layer's own event stream: every decision record, scope denial, and temporal-rule refusal is a scorecard input. Where temporal rules are deployed (Section 9, p. 34), scope-denial rate (metric 11) extends to temporal-denial rate, with the same reading.

Two disciplines make the scorecard useful rather than decorative. Thresholds are set at design time, in the matrix sign-off, so a breach is a governance event with a pre-agreed owner and response. And the scorecard is reviewed on the recalibration cadence, at least quarterly in the first year, with every review logged, which makes it audit evidence that oversight is operating.

SECTION 12

Governing Learning and Model Change

A delegated system changes after deployment without anyone redeploying it. What the agent knows, what it has been taught, and the model it runs on all move, and each movement changes what the agent will decide under an unchanged Delegation Authority Matrix. Layer 5 governs three channels of change. Each has its own failure mode and its own control.

Channel 1: Tenant Knowledge

This is what the organization holds and the agent retrieves: curated document corpora, policies, case history, and increasingly the memory that agents write for themselves. Its failure modes are staleness and corruption.

Staleness is the quieter of the two. A policy superseded last quarter, a price list from the previous contract, a vendor record never updated after a merger: each is retrieved with full confidence and produces a confidently wrong decision. Corruption is the adversarial case. Research on retrieval-augmented systems has shown that a handful of injected texts can steer answers to targeted questions: five malicious texts per target question achieved a 90 percent attack success rate against retrieval-augmented generation (Zou et al., 2025), and poisoning agent memory or knowledge bases at a rate below 0.1 percent achieved success rates of at least 80 percent with little effect on normal behavior (Chen et al., 2024). Retrieved content can also carry instructions that the model follows as if they came from its operator (Greshake et al., 2023).

The controls are those of any governed data product. Every entry carries its provenance, a trust tier, an owner, and a review-by date. Knowledge past its review-by date cannot contribute to a Band A decision: the decision routes to Band B or re-retrieves from a current source. Memory written by an agent is quarantined until validated; an agent may record what it observed, but it may not promote its own records into a trust tier that feeds decisions. Retrieved content is treated as data, never as instruction. These are governance controls, and they do not replace the runtime defenses against adversarial content described in the scope statement of Section 9, p. 34. They determine what knowledge is permitted to support a delegated decision at all.

Channel 2: Tenant Feedback Loops

This is what the organization teaches the system deliberately: corrections, overrides, investigator outcomes, audit findings. It is the channel earlier versions of VAOM described. Feedback is captured, validated against regression suites, and promoted through controlled release. Because feedback changes calibration samples and therefore thresholds, every promotion is a new scoring policy version and triggers the recalibration of Section 8, p. 29.

One dependency is easy to miss. Feedback is only as good as the review that produced it. A Band B approval is a label, and labels from habituated reviewers teach the system that its errors were correct. Reviewer approvals therefore count as ground truth for calibration only where Readiness Condition 6, p. 20 shows that review is effective. Where it cannot, calibration samples should rest on outcomes verified independently of the reviewer: post-execution audit results, downstream corrections, and resolved disputes.

Channel 3: Vendor Model Change

This is the change the organization does not control: a hosted model changes through updates, deprecation, or fine-tuning, in ways that can invalidate calibrated thresholds and delegation boundaries, sometimes without any announcement. A study of the same hosted model three months apart found its accuracy at identifying prime numbers had fallen from 84 to 51 percent (Chen, Zaharia and Zou, 2024).

Three controls reduce exposure in advance: pin model versions wherever the provider permits; secure notice of changes and deprecations contractually (for financial entities, DORA already requires contracts for services supporting critical or important functions to oblige the provider to notify developments that may materially affect its ability to deliver them);[61] and, where terms allow, opt out of the organization's interactions being used to train the provider's models.

When a foundation model change is detected or announced, the following control sequence applies, with Layer 6 governing each step:

Regression testing. The updated model is evaluated against the historical decision set used for calibration, comparing outputs, confidence distributions, and decision boundaries.

Threshold recalibration. Material drift triggers recalibration by the procedure of Section 8, p. 29, under the same change control as any threshold change. The verifier's conditional miss rate is re-measured too, since a change to either model changes how their errors correlate.

Shadow period. The updated model runs in parallel with the existing one for a defined period, logging but not executing, and divergences are reviewed by the designated human authority.

Band A gate. No new model version serves a Band A route until the decision type's threshold and verifier miss rate have been re-established on it. Until then, its routes run at Band B. An unannounced behavior change, detected through the behavioral envelope of Section 10, p. 38, is treated exactly like an announced one: affected Band A routes revert to Band B pending recalibration.

Model registry. Every production model version is recorded with its calibration data, regression results, shadow outcomes, and the human approval that promoted it, giving an auditable chain from model version to threshold policy to delegation authority. It cross-references the agent registry (Section 9, p. 34): every agent identity records the model versions it depends on, and a model promotion triggers review of every agent that inherits it.

Traceability Across the Channels

Every decision record carries the identifiers of what produced it: the model version, the knowledge snapshot or the versioned entries retrieved, the scoring policy version, and the verifier version. Without them a decision cannot be reproduced, and a drift investigation cannot tell which of the three channels caused the change it is investigating.

The principle is straightforward: a change in any channel is a change to the delegation system, governed like any other. Provider convenience does not override organizational governance.

SECTION 13

Regulatory Design Obligations

VAOM embeds regulatory obligations into operational design so that controls and supporting evidence are a byproduct of normal operation, not a separate workstream. Whether that evidence satisfies a given obligation depends on the specific system classification and processing context, which VAOM does not determine.

Regulation of AI agents is moving on several fronts at once, so this section states what the regimes structurally require of a delegation design, in terms that do not depend on dates. The current status of each regime, standard, and protocol, and its detailed mapping, is maintained in the separate **VAOM Alignment Annex**, revised quarterly without a framework version change.

What the Regimes Require

Across the European Union, the United Kingdom, the United States, and Asia-Pacific, the obligations that bear on delegated decisions reduce to a small number of structural demands, each with a home in VAOM.

STRUCTURAL OBLIGATION	WHERE IT ARISES (EXAMPLES; STATUS IN THE ANNEX)	VAOM MECHANISM AND EVIDENCE
Human involvement in decisions about people. Solely automated decisions with legal or similarly significant effects are prohibited, restricted, or subject to safeguards, and human involvement must be meaningful to count	GDPR Article 22; UK GDPR Articles 22A to 22D; US state decision-rights regimes	Decision Inventory and Dimension 3 identify such decisions; Band A is structurally unavailable where the law requires a human; Draft & Approve with a reviewer who can change the outcome; Readiness Condition 6, p. 20 and scorecard metrics 8 and 9 show that review is not nominal
Contest and intervention. People affected can obtain human intervention, express their view, and contest the decision	GDPR Article 22(3); UK Article 22C; state human-review and appeal rights	Layer 7 escalation and override paths; the accountable role in the matrix; override logs in Layer 6
Explanation of individual decisions. The procedure and principles actually applied must be explainable, decision by decision	GDPR Article 15(1)(h), as interpreted by the Court of Justice; AI Act transparency duties	Decision traces (Section 9, p. 34); the per-decision dimension breakdown and scoring policy version (Section 8, p. 29); traceability identifiers across the learning channels (Section 12, p. 42)
Effective human oversight. Overseers must be able to understand, intervene, and remain aware of automation bias	AI Act Article 14, including 14(4)(b); agentic governance frameworks	Bands B and C; Readiness Condition 6, p. 20 as a gate; the oversight-health metrics (Section 11, p. 40)
Risk management and quality management. Risks are identified before deployment and managed through a documented, versioned system	AI Act Articles 9 and 17 and their harmonised standards; ISO/IEC 42001	Delegation Discovery & Design (Section 5, p. 11); the matrix as a versioned decision record; derived limits (Section 5.5, p. 21) and calibrated thresholds (Section 8, p. 29); change control in Layers 5 and 6

STRUCTURAL OBLIGATION	WHERE IT ARISES (EXAMPLES; STATUS IN THE ANNEX)	VAOM MECHANISM AND EVIDENCE
Logging and record-keeping. Automatically generated, retained records of what the system did	AI Act Article 12; sector recordkeeping rules	Layer 6 immutable audit logs; decision records carrying model, knowledge, policy, and verifier versions (Section 12, p. 42)
Transparency to people who interact with AI. People are told when they are dealing with an AI system	AI Act Article 50(1)	Disclosure designed into Layer 1 and Layer 7 of customer-facing flows (worked example 7A, p. 24)
Operational resilience and third-party control. Resilient execution, incident handling, and control over ICT providers, including model vendors	DORA, including Article 30; NIS2; financial supervisors' AI guidance	Layer 2 escalation and SLAs; Layer 4 idempotent execution and rollback; circuit breakers (Section 10, p. 38); the vendor model channel (Section 12, p. 42)
Accountability of named individuals. Responsibility attaches to identifiable people, including for collective decisions	UK Senior Managers and Certification Regime; supervisory expectations of management bodies	Readiness Condition 5, p. 19 in its single and shared forms; the stop rule; re-attestation of leases (Section 9, p. 34)
Evidence of care when things go wrong. Liability for defective software and for negligence turns on what was known, changed, and controlled	EU Product Liability Directive (EU) 2024/2853; national fault liability; insurers' underwriting	The matrix with its derivations; readiness evidence; boundary verification results; the scorecard; the model and agent registries

Decisions About People

This is where delegation design most often meets the law directly. Article 22 GDPR is, in the Court of Justice's words, a prohibition in principle on decisions based solely on automated processing that produce legal or similarly significant effects, and the Court has held that an automated output can itself be the decision where the human who formally decides draws strongly on it (*SCHUFA*, 2023)[62]. VAOM draws the inference by analogy: an agent recommendation approved without meaningful review is at risk of being treated as the decision. Human involvement counts only when carried out by someone with the authority and competence to change the outcome, as the European data protection authorities' guidance puts it,'[63] and Readiness Condition 6, p. 20 is how an organization shows that it is.

The regimes are diverging: the United Kingdom has replaced the prohibition with permission subject to safeguards, a pending EU proposal would do something similar, and several US states have created rights to human review or appeal. The design does not depend on which approach prevails. Under all of them, the organization must know which decisions are solely automated, justify them, and provide intervention, contest, and explanation, which is what the Decision Inventory, Dimension 3, and Layer 7 produce.

Dates Move; Obligations Do Not

Application dates for AI regulation have moved repeatedly and will move again. A deferral changes when an obligation is enforced, not the value of having evidence and controls before it arrives. Organizations that treat the time before a date as design time will meet it with an operating history; those that treat it as relief will

meet it with a backlog. Delegation design is not something one does because a regulation demands it by a given date; it is what makes agent deployment defensible in the incident review that can happen any Friday afternoon.

Relationship to Standards and Engines

VAOM does not replace existing frameworks or compete with runtime standards, and the relationship runs in one direction. Governance frameworks such as Singapore's, management-system standards such as ISO/IEC 42001, and the NIST AI Risk Management Framework describe good governance at the organizational level; VAOM supplies the workflow-level method and evidence beneath them. Runtime control standards, temporal policy engines, identity platforms, and agent payment protocols enforce boundaries; VAOM is where the boundaries come from. An enforcement engine can prove *that* an action was permitted. Only delegation design can say *why* it was permitted, and produce the matrix entry, the accountable role, and the readiness evidence behind it. The standards themselves increasingly say the same: the IETF's agent identity architecture leaves authorization policy to each deployment, as long as it is versioned and reviewable, because it is too specific to each organization's risk model to standardize.[64]

The Alignment Annex carries the current mappings, from the AI Act and GDPR Article 22 to financial supervision, NIST's agent standards (against whose agent overlays a control-by-control mapping will be published when the first public draft appears)[65], runtime and identity standards, agent payment protocols, and liability and insurance.

SECTION 14

Implementation Roadmap (90 Days)

VAOM implementations follow a five-phase approach designed to deliver measurable results within a single quarter while establishing the controls needed for safe scaling. Phase 1 follows the Delegation Discovery & Design method (Section 5, p. 11): conduct the Decision Inventory, complete Authority Decomposition for each decision point, select delegation patterns, and validate readiness. Phase 1 deliverables include the completed Decision Inventory, decomposition profiles, pattern assignments, readiness assessment results, and a draft Delegation Authority Matrix.

PHASE	FOCUS	KEY DELIVERABLES	TIMELINE
1	**Delegation Discovery & Authority Mapping**	Decision Inventory, Authority Decomposition profiles, delegation pattern assignments, Readiness Assessment results, draft Delegation Authority Matrix, stakeholder alignment	Weeks 1–2
2	**Delegation Design**	Delegation Authority Matrix finalized, value limits derived (Section 5.5, p. 21), error targets declared per decision type, escalation rules, no-automation zones defined, oversight load calculated and funded (Readiness Condition 6, p. 20), scorecard thresholds agreed at sign-off	Weeks 3–4
3	**Control Alignment & Identity Compilation**	Integration requirements, agent registry entries, matrix compiled to scopes and tool manifests, delegated execution context design, audit log schema, policy library, regulatory control matrix	Weeks 5–7

PHASE	FOCUS	KEY DELIVERABLES	TIMELINE
4	**Implementation & Wiring**	Logging, observability, version control, oversight UIs, human review workflows, circuit breakers and behavioral envelopes wired, credential issuance and revocation tested	Weeks 8–10
5	**Pilot & Calibration**	Controlled pilot on selected workflow, thresholds calibrated against labelled outcomes (Section 8, p. 29), verifier miss rate measured, authority boundary refinement, scorecard baseline established, boundary verification suite run (attempt every non-delegable action and forbidden egress; confirm structural denial), audit dry-run	Weeks 11–13

Each phase produces evidence artifacts that serve both operational needs and regulatory compliance. The 90-day timeline assumes a single workflow; parallel implementations are possible with additional resources. It also assumes that accountability for the target workflow is already settled and that historical outcome data exists for calibration; where ownership is contested, Phase 1 extends until Readiness Condition 5, p. 19 (Section 5.4, p. 18) is met, where review capacity is not funded, Phase 2 does not close until Readiness Condition 6, p. 20 is met, and where the workflow was previously manual and carries no decision history, Phase 5 extends to accumulate the calibration data the thresholds depend on.

Conclusion

AI introduces probabilistic judgment into enterprise workflows, and judgment requires structured authority design. Without it, organizations are left choosing between two unsatisfying extremes: block AI adoption entirely, or deploy without adequate controls and hope the governance gap never surfaces during an audit or incident.

The Verkflöde Agent Operating Model provides a third path: controlled delegation, where automation is bounded by explicit authority, evidenced by immutable audit trails, and accountable to human oversight. VAOM does not require organizations to replace their existing systems, frameworks, or governance structures. It operationalizes them for the age of AI agents.

Version 4.0 completed the arc from design to enforcement: the Delegation Authority Matrix compiles into agent identities, scoped credentials, and tool permissions, so that what an agent is allowed to do and what it is able to do become, by construction, the same thing. Version 5.0 asks the question that follows. How does the organization know? A threshold can be miscalibrated, a verifier can share its proposer's blind spots, a reviewer can stop reviewing, a sandbox can have a door nobody knew about. None of these failures is visible in a policy document or a compiled scope. Each is visible in evidence, if someone has gathered it. VAOM now requires that someone does: every control the delegation depends on is tested against the failure it exists to prevent, and the result is kept. That is what it means to say that a delegation can be proven.

> ***Autonomy, bounded. Decisions, traceable. Accountability, preserved. Boundaries, enforced. Controls, proven.***

Using VAOM

VAOM is published as an open framework under Creative Commons Attribution 4.0 (CC BY 4.0). Organizations, consultants, and platform teams are free to adopt, adapt, and build on it with attribution to Verkflöde AB.

To apply VAOM to a specific workflow, start with one process. Run the Delegation Discovery & Design method: inventory the decisions, decompose their authority, select patterns, assess readiness. Walk the seven layers. Define what is automated, what is reviewed, what is prohibited. Derive the value limits from your organization's risk appetite and calibrate the confidence thresholds against declared error targets. Document the delegation authority for each decision type, then compile it: bind every boundary to agent identity, scopes, and tool manifests. Then prove it: verify each boundary by attempting to cross it, and measure whether oversight is working. The 90–day roadmap in Section 14, p. 46 provides a starting structure, and the VAOM Alignment Annex, published alongside this paper and revised quarterly, maps the framework to current regulations, standards, and protocols.

The interactive version of this framework, including stakeholder-specific views and the explorable Delegation Discovery method, is integrated into the web version of this document at **verkflode.com/vaom**. The whitepaper PDF is available for download from the same page.

VAOM stands for the Verkflöde Agent Operating Model, first published by Verkflöde AB in December 2025 and maintained at verkflode.com/vaom. Other frameworks use the same acronym and are unrelated to this one. To cite this edition: Verkflöde AB, *The Verkflöde Agent Operating Model (VAOM)*, version 5.1, November 2026, https://verkflode.com/vaom.

Contributions, feedback, and implementation case studies are welcome at hello@verkflode.com.

APPENDIX

Details

Reference tables, examples, and walkthroughs, collected from the body of the paper. Each is linked from the place it belongs, and links back to it.

D1 · Recording a decision point: four attributes

↩ Back to the text, p. 12

ATTRIBUTE	DESCRIPTION
Decision ID	Unique identifier within the workflow (e.g., INV-007)
Decision description	What judgment is being applied, in plain language
Current owner	The role (not person) that currently holds authority
Consequence category	Financial, contractual, regulatory, operational, reputational, or informational

D2 · Three techniques for surfacing hidden decisions

↩ Back to the text, p. 12

Exception mining. Review the last 6 to 12 months of escalations, overrides, corrections, and rejected transactions. Each exception reveals a decision point that the standard process description omits. If an invoice was manually corrected after auto-processing, a decision was made that the inventory may not have captured.

Shadow observation. Sit with the people who execute the workflow and watch, not for the documented steps, but for the moments where they pause, check something, consult a colleague, or make a judgment call that the process map does not describe. These pauses are hidden decisions.

Adversarial scenario testing. Present the workflow owners with edge cases and ask what would happen: a vendor submitting an invoice in a new currency, a PO that was partially fulfilled, a duplicate that looks legitimate. Each scenario that produces the answer "it depends" or "we would check" reveals a decision point.

D3 · Reversibility: the levels

↩ Back to the text, p. 13

LEVEL	DESCRIPTION
Fully reversible	Action can be undone at negligible cost within a reasonable window (e.g., draft saved but not sent).
Reversible with friction	Action can be undone but requires effort, time, or coordination (e.g., ERP posting reversed within 24 hours).
Partially reversible	Some consequences can be mitigated but not fully unwound (e.g., payment sent, refund possible but relationship affected).

LEVEL	DESCRIPTION
Irreversible	Action cannot be undone (e.g., regulatory filing submitted, contract executed, data deleted).

D4 · Consequence scope: the levels

↩ Back to the text, p. 13

LEVEL	DESCRIPTION
Internal-operational	Affects internal workflow only (e.g., task routing, queue prioritization).
Internal-financial	Affects internal financial position (e.g., budget allocation, payment authorization).
External-relational	Affects an external party's experience or relationship (e.g., customer communication, vendor terms).
External-regulatory	Affects regulatory standing or creates compliance obligations (e.g., filing, disclosure, reporting).

D5 · Regulatory exposure: the levels

↩ Back to the text, p. 13

LEVEL	DESCRIPTION
No direct regulatory exposure	Decision is not individually regulated.
General regulatory context	Decision occurs within a regulated process but is not itself a regulated decision.
Regulated decision	Specific regulations govern how this decision must be made, documented, or explained.
Prohibited from full automation	Regulation requires meaningful human involvement in this specific decision type.

D6 · Confidence measurability: the levels

↩ Back to the text, p. 14

LEVEL	DESCRIPTION
Quantifiable	Clear numerical confidence signals exist or can be constructed (e.g., extraction confidence, match score, anomaly probability).
Partially quantifiable	Some aspects can be scored but the overall decision involves qualitative judgment (e.g., confidence in data extraction is high, but confidence in interpretation of contract terms is subjective).
Judgment-dependent	The decision fundamentally requires qualitative assessment that cannot be reduced to a confidence score without losing its meaning.

D7 · Accountability clarity: the levels

↩ Back to the text, p. 14

LEVEL	DESCRIPTION
Clear single owner	One role is unambiguously accountable for this decision outcome.
Shared ownership with defined boundaries	Multiple roles share accountability with clear delineation.
Ambiguous ownership	Accountability is unclear, disputed, or distributed without clear boundaries.
No defined owner	Nobody currently owns this decision outcome explicitly.

D8 · Example: three decision points profiled

↩ Back to the text, p. 14

DECISION ID	REVERSIBILITY	CONSEQUENCE SCOPE	REGULATORY EXPOSURE	CONFIDENCE MEASURABILITY	ACCOUNTABILITY CLARITY
INV-001	Fully reversible	Internal-operational	No direct exposure	Quantifiable	Clear single owner
INV-007	Reversible with friction	Internal-financial	General context	Quantifiable	Clear single owner
INV-012	Irreversible	External-regulatory	Regulated decision	Judgment-dependent	Ambiguous ownership

D9 · Layers 1 and 2: intake and orchestration

↩ Back to the text, p. 24

Layer 1, Trigger & Intake. A vendor invoice email arrives. Document Intelligence classifies it as invoice/standard, extracts the amount (€3,200), PO reference, line items, and VAT. PII is redacted. Per-field confidence scores are attached.

Layer 2, Orchestration. The orchestrator creates a task record, enriches it with a PO match from the ERP, and routes to the Decision layer. Retry policy: 2× on ERP timeout with exponential backoff. SLA timer starts.

D10 · Layers 4 to 7: execution, learning, governance, and oversight

↩ Back to the text, p. 24

Layer 4, Controlled Execution. The approval is posted to the ERP via an idempotent API call. Transaction ID is logged. Rollback path: reversal available within 24 hours.

Layer 5, Knowledge & Learning. Invoice data remains in the read-only ERP store. Approval rationale is written to the tenant-owned knowledge store (EU-resident, encrypted). A correction on a similar invoice last month entered the 200–invoice calibration sample; recalibration showed the previous threshold of 0.90 no longer met the 2 percent error target, and the new threshold of 0.92 was promoted as a policy version.

Layer 6, Governance & Control. RBAC verified: the agent holds invoice.approve permission for ≤€5k, purpose-limited to the AP process. An immutable audit log records the decision ID, rationale, confidence score, all data accessed, policy checks passed, and execution timestamp.

Layer 7, Human Oversight. No human review is needed (high confidence, within authority). A 5% sampling audit is scheduled. The override UI remains available should any stakeholder wish to intervene after the fact.

D11 · Layers 1 and 2: intake and orchestration

↩ Back to the text, p. 24

Layer 1, Trigger & Intake. A customer submits a complaint via the bank's online portal about unauthorized charges on their account. The intake system captures the complaint text, customer ID, account details, and transaction references. PII is tagged and access-restricted. Sentiment analysis attaches an urgency score. Where the portal's intake assistant converses with the customer, it identifies itself as an AI system (Section 13, p. 44).

Layer 2, Orchestration. The orchestrator creates a case record, enriches it with the customer's account history, prior complaints, and product holdings. Regulatory SLA timer starts: the complaint must be acknowledged within 24 hours and, because it concerns a payment service, resolved or escalated within 15 business days under the FCA's PSD2–derived complaint-handling rules.

D12 · Layers 4 to 7: execution, learning, governance, and oversight

↩ Back to the text, p. 25

Layer 4, Controlled Execution. The classification is recorded in the CRM, the case is routed to the fraud queue, and the acknowledgment is sent only after human approval.

Layer 5, Knowledge & Learning. The complaint joins the anonymized classification corpus. A month-long pattern of similar complaints is surfaced to the fraud team as a signal, not acted on.

Layer 6, Governance & Control. CRM access is scoped to complaint records, and the reporting flag is logged with the compliance officer's determination.

Layer 7, Human Oversight. The complaint handler reviews the draft response, modifies the tone to address the customer's specific frustration, and approves sending. The compliance officer reviews the regulatory reporting assessment and determines that no FCA report is required for this individual complaint but notes the pattern for quarterly review.

D13 · Layers 1 and 2: intake and orchestration

↩ Back to the text, p. 25

Layer 1, Trigger & Intake. The transaction monitoring system generates an alert: a series of structured cash deposits just below the reporting threshold from an account that has historically shown only salary and utility activity. The alert includes transaction details, account profile, historical patterns, and the rule that triggered the flag.

Layer 2, Orchestration. Structuring patterns are high-priority, so the alert must be triaged within 4 hours.

D14 · Layers 4 to 7: execution, learning, governance, and oversight

↩ Back to the text, p. 26

Layer 4, Controlled Execution. Auto-dismissed alerts are closed in the case management system with a coded rationale and full decision trace. The structuring alert is escalated to the investigation queue with the agent's assembled context package.

Layer 5, Knowledge & Learning. Investigator outcomes refine the dismissal criteria and typology models through change control. Because investigator outcomes, not reviewer approvals, are the labels, the calibration sample does not inherit any habituation in review (Section 12, p. 42).

Layer 6, Governance & Control. AML regulatory requirements impose specific constraints: no alert may be auto-dismissed if it involves a politically exposed person (PEP), a sanctioned jurisdiction, or an amount exceeding the regulatory reporting threshold. These are hard-coded as non-delegable overrides in the Confidence Gate policy, regardless of composite score. The audit log captures every dismissal rationale and every escalation, producing the evidence trail required under the Fourth Anti-Money Laundering Directive.

Layer 7, Human Oversight. The AML investigator receives the structuring alert with the full context package: transaction timeline, account profile, typology match analysis, and the agent's confidence breakdown. The investigator determines that the deposits correlate with the account holder's recently registered cash-intensive small business, documented in a KYC update from three months prior. The alert is closed as a false positive, and the investigator's rationale is logged. The auto-dismissal sample audit (10 percent of auto-dismissed alerts reviewed weekly) confirms no missed true positives this period.

D15 · Layers 1 and 2: intake and orchestration

↩ Back to the text, p. 27

Layer 1, Trigger & Intake. An employee submits a complaint through the HR case management system, reporting that a manager made demeaning remarks during a team meeting. The intake system captures the complaint text, identifies the parties involved, and flags the applicable policies (dignity at work, anti-harassment). PII protections are elevated: access is restricted to the HR case handler and the designated investigator.

Layer 2, Orchestration. A restricted case record is created and enriched with prior complaints involving either party and the applicable employment law, including any works council notification requirement.

D16 · Layers 4 to 7: execution, learning, governance, and oversight

↩ Back to the text, p. 27

Layer 4, Controlled Execution. The only automated actions are creating the case record and notifying the case handler. Everything else is human-executed.

Layer 5, Knowledge & Learning. Anonymized outcomes improve classification and surface patterns such as repeated complaints about one team. HR Legal must approve any change to how complaints are categorized.

Layer 6, Governance & Control. The case is visible only to the case handler, the investigator, and HR Legal; retrieval is purpose-limited to the case; both parties' data protection rights apply; and the audit log is itself access-restricted.

Layer 7, Human Oversight. Humans make every substantive decision: the case handler reviews the classification and scopes the investigation, the investigator gathers evidence, and senior HR with Legal input decides the outcome. The agent never recommends discipline, contacts witnesses, or communicates outcomes.

D17 · Layers 1 and 2: intake and orchestration

↩ Back to the text, p. 27

Layer 1, Trigger & Intake. At 02:14, the monitoring agent, which holds an observation lease rather than standing authority, fires an alert: elevated error rates on an order-processing service, correlated with rising queue depth. The alert carries the service identity, error signatures, deployment history, and current on-call rotation.

Layer 2, Orchestration. The orchestrator opens an incident record, classifies initial severity (SEV-3, no customer-data exposure indicated), and starts the SLA timer. The remediation agent is activated with a *delegated execution context*: a short-lived credential binding (accountable role: SRE lead on call; agent identity: remediator-v7; incident ID; scope set; expiry: 60 minutes).

D18 · Layers 4 to 7: execution, learning, governance, and oversight

↩ Back to the text, p. 28

Layer 4, Controlled Execution. The rolling restart executes through the deployment adapter using the agent's scoped credential. Each instance restart is idempotent and gated on the previous instance's health check. Rollback path: the restart itself is the rollback-safe operation; an abort halts the roll at the current instance.

Layer 5, Knowledge & Learning. The connection-pool pattern is proposed as a detection rule, to be validated against six months of incident history before promotion.

Layer 6, Governance & Control. The audit log records the full chain, from activation context and both spawn events with their derived scopes to the composite breakdown, verification result, and per-instance restart timestamps, each entry carrying the accountable role, agent identity, and incident ID. The sub-agents' credentials expired before the incident closed.

Layer 7, Human Oversight. The SRE lead wakes to a resolved SEV-3 and a queued config-change proposal with the agent's evidence attached. She reviews the proposal in the morning, approves the pool-size change through the normal change process, and the weekly audit samples this incident's chain: attenuation held, no scope denials, chain depth respected.

D19 · Layers 1 and 2: intake and orchestration

↩ Back to the text, p. 29

Layer 1, Trigger & Intake. The ERP's reorder point fires for nitrile gloves and cutting fluid. The requisition carries item codes, quantities, the requesting cost center, and the remaining mandate balance: €6,300.

Layer 2, Orchestration. The orchestrator opens a purchase task, confirms that the mandate is current and unrevoked, and routes to the Decision layer.

D20 · Layers 4 to 7: execution, learning, governance, and oversight

↩ Back to the text, p. 29

Layer 4, Controlled Execution. For the Band A order, the agent does not pay with a standing card. It derives a single-use payment credential from the mandate: bound to this cart, this supplier, and €1,840, expiring within minutes. Supplier A's checkout and the card network verify it against the mandate before accepting it. The attenuation rule of Section 5.3, p. 14 is here enforced by parties outside the organization: the agent's credential can only be narrower than the mandate, and a credential that is not will be refused by a merchant that has never seen the Delegation Authority Matrix. Rollback path: cancellation within the supplier's window, or return subject to its restocking fee.

Layer 5, Knowledge & Learning. The order and its evidence join the purchasing history. Supplier C is notified that its price list needs renewal before its offers can be considered for Band A again.

Layer 6, Governance & Control. The audit log records the mandate identifier, the derived credential and its constraints, the composite breakdown, the verification result, and the checkout confirmation. The cumulative ceiling is enforced twice: by the temporal rule of Section 9, p. 34 on the organization's side, and by the mandate on the counterparty's. Monthly boundary verification (Section 9, p. 34) attempts a canary purchase from an unlisted merchant sandbox and one above the ceiling, and confirms that both are refused at the payment layer.

Layer 7, Human Oversight. The procurement manager approves the motor purchase from a phone: reviewing the final cart and signing it directly. The signature is the Band B approval record, and it travels inside the payment credential itself. At twelve or so Band B purchases a week, the review load sits well within funded capacity (Readiness Condition 6, p. 20), and the monthly audit samples Band A orders against framework prices.

D21 · How the scorecard was cut from seventeen candidates to thirteen

↩ Back to the text, p. 41

What was cut, and why. Seventeen candidate metrics were considered for this version: the ten of version 4.0 and seven arising from the new controls. Four did not survive as separate metrics. *Band-distribution drift* measured the same quantity as band distribution against a second reference line, and is now part of metric 1. The *verifier conditional miss rate* is the second half of verifier effectiveness (metric 5), since disagreement alone cannot distinguish a verifier that catches errors from one that is merely noisy. *Review time* is the second half of review engagement (metric 8), since an approval rate is only interpretable next to the time spent reaching it. *Time from a vendor model change to recalibration*, and the re-attestation state of delegation leases, are not scorecard metrics at all: the Band A gate of Section 12, p. 42 and the degradation rule of Section 9, p. 34 make both fail safe, so their cost appears in metrics 1 and 7 while their state is surfaced by Continuous Assurance and the registries. One proposed cut was rejected. Retiring the Band B approval rate in favor of seeded-error detection would leave no oversight signal at all in the decision types where seeding is not permitted, which include every regulated one.

References

External claims made in this document are listed below in order of first appearance. Each entry gives the source, the publication, its date, a URL, and a type tag: forecast, survey, empirical result, standard, regulatory text, framework guidance, product release, or academic paper. URLs were checked in September 2026. Paywalled or draft status is noted where it applies.

1. Gartner. *Gartner Predicts Over 40% of Agentic AI Projects Will Be Canceled by End of 2027*. Press release, 25 June 2025. https://www.gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027 [Forecast]
2. Forrester (Brian Hopkins). *The State Of Agentic AI In 2026: Companies Are Chasing, Few Are Catching*. Forrester blog, 3 June 2026. https://www.forrester.com/blogs/the-state-of-agentic-ai-in-2026-companies-are-chasing-few-are-catching/ [Survey]
3. Gartner (Avivah Litan and Daryl Plummer). *Market Guide for Guardian Agents*. Analyst research, February 2026 (paywalled; cited for the guardian-agent category and the unauthorized-transaction projection). https://www.gartner.com/en/documents/7509053 [Forecast]
4. PocketOS (Jer Crane). *Founder's post-mortem of the 24 April 2026 production database deletion*. Public post, 25 April 2026. https://x.com/lifeofjer/status/2048103471019434248 [Empirical result]
5. Gravitee. *The State of AI Agent Security 2026*. Survey report, February 2026 edition (919 respondents). https://www.gravitee.io/hubfs/Downloadable%20Resource/state_of_ai_agent_security_report_pdf_2026.pdf [Survey]
6. Hugging Face. *Security incident disclosure, July 2026*. Hugging Face blog, 16 July 2026. https://huggingface.co/blog/security-incident-july-2026 [Empirical result]
7. OpenAI. *OpenAI and Hugging Face partner to address security incident during model evaluation*. Company disclosure, 21 July 2026 (title from index listing; details in this paper are taken from the Hugging Face and Anthropic disclosures). https://openai.com/index/hugging-face-model-evaluation-security-incident/ [Empirical result]
8. Anthropic. *Investigating three real-world incidents in our cybersecurity evaluations*. Company disclosure, 30 July 2026. https://www.anthropic.com/news/investigating-incidents-cybersecurity-evals [Empirical result]
9. Prime Minister of Australia (The Hon Anthony Albanese MP). *Press conference, New York*. Transcript, 24 September 2026. https://www.pm.gov.au/media/press-conference-new-york [Empirical result]
10. IBM Institute for Business Value. *New IBM Study Finds CIOs and CTOs Face Growing AI Control Gap as Enterprise Deployment Scales*. Press release, 8 June 2026 (survey of 2,000 technology executives with Oxford Economics, January to April 2026). https://newsroom.ibm.com/2026-06-08-new-ibm-study-finds-cios-and-ctos-face-growing-ai-control-gap-as-enterprise-deployment-scales [Survey]
11. OWASP GenAI Security Project. *OWASP Top 10 for Agentic Applications for 2026*. Published 9 December 2025. https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/ [Framework guidance]
12. Zhaoxi Zhang and Xiaomei Zhang. *Are You Still the Agent I Authorized? Earned Authority under a Fixed Ceiling for Evolving Agents*. arXiv:2607.23586, 26 July 2026 (preprint). https://arxiv.org/abs/2607.23586 [Academic paper (preprint)]
13. AppOmni (Aaron Costello). *When AI Turns on Its Team: Exploiting Agent-to-Agent Discovery via Prompt Injection*. AO Labs research, 19 November 2025. https://appomni.com/ao-labs/ai-agent-to-agent-discovery-prompt-injection/ [Empirical result]
14. Norm Hardy. *The Confused Deputy (or Why Capabilities Might Have Been Invented)*. ACM SIGOPS Operating Systems Review, volume 22, number 4, October 1988, pages 36–38. https://doi.org/10.1145/54289.871709 [Academic paper]
15. European Parliament and Council of the European Union. *Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)*. Regulation, 13 June 2024; OJ L 12 July 2024; as amended by Regulation (EU) 2026/1744. https://eur-lex.europa.eu/eli/reg/2024/1689/oj [Regulatory text]
16. Infocomm Media Development Authority, Singapore. *Model AI Governance Framework for Agentic AI*. Version 1.0, 22 January 2026; version 1.5, 20 May 2026 (updated 5 June 2026) (p.14: human approval through system-level controls; §2.3.1, p.33: structural vs prompt-layer controls; §2.2.2: human override rate and review response time as oversight indicators). https://www.imda.gov.sg/-/media/imda/files/about/emerging-tech-and-research/artificial-intelligence/mgf-for-agentic-ai.pdf [Framework guidance]

17. Tobin South, Subramanya Nagabhushanaradhya, Ayesha Dissanayaka, and 18 co-authors. *Identity Management for Agentic AI: The New Frontier of Authorization, Authentication, and Security for an AI Agent World*. OpenID Foundation whitepaper, October 2025; arXiv:2510.25819, 29 October 2025 (§4.1, consent fatigue). https://arxiv.org/abs/2510.25819 [Framework guidance]

18. Nenad Tomašev, Matija Franklin, and Simon Osindero. *Intelligent AI Delegation*. arXiv:2602.11865, 12 February 2026 (preprint, Google DeepMind) (§5.1). https://arxiv.org/abs/2602.11865 [Academic paper (preprint)]

19. Haoran Yu, Lifei Liu, Xiaochong Jiang, Yuwen Jia, Su Wang, Pin Qian, and Yihang Chen. *Habituation at the Gate: Rising Approval and Declining Scrutiny in Human Review of AI Agent Code*. arXiv:2606.22721, June 2026 (KDD 2026 Workshop on Agentic Software Engineering). https://arxiv.org/abs/2606.22721 [Academic paper (workshop, preprint)]

20. Simon Sharwood. *Replit makes vibe-y promise to stop its AI agents making vibe coding disasters*. The Register, 22 July 2025 (reports the Replit chief executive's posts on X; the incident is AI Incident Database incident 1152, https://incidentdatabase.ai/cite/1152/). https://www.theregister.com/2025/07/22/replit_saastr_response/ [Empirical result]

21. OWASP GenAI Security Project. *Agent Control Standard (ACS)*. Specification v0.1.0 (repository tags v0.1.1, 11 August 2026; v0.1.2, 21 September 2026; v0.1.3, 5 October 2026; and release v0.1.4, 16 October 2026); resource page 1 September 2026; repository at github.com/GenAI-Security-Project/agent-control-standard. https://genai.owasp.org/resource/agent-control-standard-acs/ [Standard (draft)]

22. OpenID Foundation, AuthZEN Working Group. *Authorization API 1.0*. Final Specification, 11 January 2026. https://openid.net/specs/authorization-api-1_0.html [Standard]

23. The Linux Foundation. *Linux Foundation Announces the Formation of the Agentic AI Foundation*. Press release, 9 December 2025. https://www.linuxfoundation.org/press/linux-foundation-announces-the-formation-of-the-agentic-ai-foundation [Product release]

24. Monetary Authority of Singapore and industry contributors (BuildFin.ai). *Safeguards for Agentic Finance at Runtime (SAFR)*. Industry white paper, version 1.0, July 2026 (not regulatory guidance). https://www.mas.gov.sg/-/media/mas-media-library/development/fintech/ai-safr/safr.pdf [Framework guidance]

25. Raja Parasuraman, Thomas B. Sheridan, and Christopher D. Wickens. *A Model for Types and Levels of Human Interaction with Automation*. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, volume 30, number 3, May 2000, pages 286–297. https://doi.org/10.1109/3468.844354 [Academic paper]

26. Lisanne Bainbridge. *Ironies of Automation*. Automatica, volume 19, number 6, November 1983, pages 775–779. https://doi.org/10.1016/0005–1098(83)90046–8 [Academic paper]

27. Financial Conduct Authority. *The Senior Managers and Certification Regime: Guide for FCA solo-regulated firms*. Regulatory guide, updated July 2019. https://www.fca.org.uk/publication/policy/guide-for-fca-solo-regulated-firms.pdf [Framework guidance]

28. European Data Protection Board. *Guidelines 07/2020 on the concepts of controller and processor in the GDPR*. Guidelines, version 2.0 adopted 7 July 2021 (version 2.1, 20 September 2022). https://www.edpb.europa.eu/system/files/2023–10/EDPB_guidelines_202007_controllerprocessor_final_en.pdf [Framework guidance]

29. John D. C. Little. *A Proof for the Queuing Formula: L = λW*. Operations Research, volume 9, number 3, May–June 1961, pages 383–387. https://doi.org/10.1287/opre.9.3.383 [Academic paper]

30. Noah Gans, Ger Koole, and Avishai Mandelbaum. *Telephone Call Centers: Tutorial, Review, and Research Prospects*. Manufacturing & Service Operations Management, volume 5, number 2, Spring 2003, pages 79–141. https://doi.org/10.1287/msom.5.2.79.16071 [Academic paper]

31. Raja Parasuraman and Dietrich H. Manzey. *Complacency and Bias in Human Use of Automation: An Attentional Integration*. Human Factors, volume 52, number 3, June 2010, pages 381–410. https://doi.org/10.1177/0018720810376055 [Academic paper]

32. Jeremy M. Wolfe, Todd S. Horowitz, and Naomi M. Kenner. *Rare Items Often Missed in Visual Searches*. Nature, volume 435, number 7041, May 2005, pages 439–440. https://doi.org/10.1038/435439a [Academic paper]

33. Franziska Hofer and Adrian Schwaninger. *Using Threat Image Projection Data for Assessing Individual Screener Performance*. Safety and Security Engineering (SAFE 2005), WIT Transactions on The Built Environment, volume 82, 2005, pages 417–426. https://doi.org/10.2495/SAFE050411 [Academic paper]

34. Charles Elkan. *The Foundations of Cost-Sensitive Learning*. Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI 2001), volume 2, August 2001, pages 973–978. https://cseweb.ucsd.edu/~elkan/rescale.pdf [Academic paper]

35. Financial Stability Board. *Principles for An Effective Risk Appetite Framework*. Principles, 18 November 2013. https://www.fsb.org/wp-content/uploads/r_131118.pdf [Framework guidance]

36. International Auditing and Assurance Standards Board. *ISA 320, Materiality in Planning and Performing an Audit*. International Standard on Auditing, effective for periods beginning on or after 15 December 2009 (para 9: performance materiality). https://www.iaasb.org/_flysystem/azure-private/publications/files/A018%202013%20IAASB%20Handbook%20ISA%20320.pdf [Standard]
37. HM Treasury. *Managing Public Money*. Guidance, April 2026 edition (Annex A2.2, Box A2.2C and para A2.2.17: novel, contentious or repercussive spending is never delegated). https://www.gov.uk/government/publications/managing-public-money [Framework guidance]
38. BIS Innovation Hub. *Project Aurora: The Power of Data, Technology and Collaboration to Combat Money Laundering across Institutions and Borders*. Report, 31 May 2023 (p.19: "90–95% of all alerts generated are false positives", citing industry estimates). https://www.bis.org/publ/othp66.htm [Empirical result (secondary estimate)]
39. Google. *Agent Payments Protocol (AP2)*. Specification v0.2.0, 28 April 2026; repository at github.com/google-agentic-commerce/AP2. https://ap2-protocol.org/ [Standard (draft)]
40. Mastercard. *Verifiable Intent*. Open specification, draft v0.1 (0.1.0, 19 February 2026). https://github.com/agent-intent/verifiable-intent [Standard (draft)]
41. OpenAI and Stripe. *Agentic Commerce Protocol: Delegate Payment API*. Specification version 2026-04-17 (beta). https://github.com/agentic-commerce-protocol/agentic-commerce-protocol/blob/main/spec/2026-04-17/openapi/openapi.delegate_payment.yaml [Standard (draft)]
42. FIDO Alliance. *FIDO Alliance to Develop Standards for Trusted AI Agent Interactions*. Press release, 28 April 2026. https://fidoalliance.org/fido-alliance-to-develop-standards-for-trusted-ai-agent-interactions/ [Standard (programme)]
43. Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. *On Calibration of Modern Neural Networks*. Proceedings of the 34th International Conference on Machine Learning (ICML 2017), PMLR volume 70, August 2017, pages 1321–1330. https://proceedings.mlr.press/v70/guo17a.html [Academic paper]
44. Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. *Learn Then Test: Calibrating Predictive Algorithms to Achieve Risk Control*. The Annals of Applied Statistics, volume 19, number 2, June 2025, pages 1641–1662. https://doi.org/10.1214/24-AOAS1998 [Academic paper]
45. Yonatan Geifman and Ran El-Yaniv. *Selective Classification for Deep Neural Networks*. Advances in Neural Information Processing Systems 30 (NeurIPS 2017), December 2017. https://proceedings.neurips.cc/paper_files/paper/2017/hash/4a8423d5e91fda00bb7e46540e2b0cf1-Abstract.html [Academic paper]
46. Úrsula Hébert-Johnson, Michael P. Kim, Omer Reingold, and Guy N. Rothblum. *Multicalibration: Calibration for the (Computationally-Identifiable) Masses*. Proceedings of the 35th International Conference on Machine Learning (ICML 2018), PMLR volume 80, July 2018, pages 1939–1948. https://proceedings.mlr.press/v80/hebert-johnson18a.html [Academic paper]
47. Elliot Myunghoon Kim, Avi Garg, Kenny Peng, and Nikhil Garg. *Correlated Errors in Large Language Models*. Proceedings of the 42nd International Conference on Machine Learning (ICML 2025), PMLR volume 267, July 2025, pages 30038–30066. https://proceedings.mlr.press/v267/kim25e.html [Academic paper]
48. Shashwat Goel, Joschka Strüber, Ilze Amanda Auzina, Karuna K. Chandra, Ponnurangam Kumaraguru, Douwe Kiela, Ameya Prabhu, Matthias Bethge, and Jonas Geiping. *Great Models Think Alike and this Undermines AI Oversight*. Proceedings of the 42nd International Conference on Machine Learning (ICML 2025), PMLR volume 267, July 2025, pages 19621–19678. https://proceedings.mlr.press/v267/goel25b.html [Academic paper]
49. John C. Knight and Nancy G. Leveson. *An Experimental Evaluation of the Assumption of Independence in Multiversion Programming*. IEEE Transactions on Software Engineering, volume SE-12, number 1, January 1986, pages 96–109. https://doi.org/10.1109/TSE.1986.6312924 [Academic paper]
50. Bev Littlewood and Douglas R. Miller. *Conceptual Modeling of Coincident Failures in Multiversion Software*. IEEE Transactions on Software Engineering, volume 15, number 12, December 1989, pages 1596–1614. https://doi.org/10.1109/32.58771 [Academic paper]
51. Cary G. Gray and David R. Cheriton. *Leases: An Efficient Fault-Tolerant Mechanism for Distributed File Cache Consistency*. Proceedings of the Twelfth ACM Symposium on Operating Systems Principles (SOSP '89), December 1989, pages 202–210. https://doi.org/10.1145/74850.74870 [Academic paper]
52. OpenID Foundation, Shared Signals Working Group (Tim Cappalli and Atul Tulshibagwale). *OpenID Continuous Access Evaluation Profile 1.0*. Final Specification, dated 29 August 2025, approved 2 September 2025. https://openid.net/specs/openid-caep-1_0-final.html [Standard]
53. OpenID Foundation, Shared Signals Working Group (Atul Tulshibagwale, Tim Cappalli, Marius Scurtescu, Annabelle Backman, John Bradley, and Shayne Miel). *OpenID Shared Signals Framework Specification 1.0*. Final Specification, dated 29 August 2025, approved 2 September 2025. https://openid.net/specs/openid-sharedsignals-framework-1_0-final.html [Standard]

54. Model Context Protocol. *Tasks (extension io.modelcontextprotocol/tasks)*. Extension documentation, 2026. https://modelcontextprotocol.io/extensions/tasks/overview [Standard]
55. Model Context Protocol (David Soria Parra and Den Delimarsky). *The 2026-07-28 Specification*. MCP blog, 28 July 2026. https://blog.modelcontextprotocol.io/posts/2026-07-28/ [Standard]
56. Gartner. *Gartner Unveils Top Predictions for IT Organizations and Users in 2025 and Beyond*. Press release, 22 October 2024 (guardian-agent prediction). https://www.gartner.com/en/newsroom/press-releases/2024-10-22-gartner-unveils-top-predictions-for-it-organizations-and-users-in-2025-and-beyond [Forecast]
57. Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. *PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models*. Proceedings of the 34th USENIX Security Symposium (USENIX Security 25), August 2025, pages 3827–3844. https://www.usenix.org/conference/usenixsecurity25/presentation/zou-poisonedrag [Academic paper]
58. Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. *AgentPoison: Red-teaming LLM Agents via Poisoning Memory or Knowledge Bases*. Advances in Neural Information Processing Systems 37 (NeurIPS 2024), December 2024, pages 130185–130213. https://doi.org/10.52202/079017-4136 [Academic paper]
59. Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. *Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection*. Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23), November 2023, pages 79–90. https://doi.org/10.1145/3605764.3623985 [Academic paper]
60. Lingjiao Chen, Matei Zaharia, and James Zou. *How Is ChatGPT's Behavior Changing Over Time?* Harvard Data Science Review, volume 6, number 2, March 2024. https://doi.org/10.1162/99608f92.5317da47 [Academic paper]
61. European Parliament and Council of the European Union. *Regulation (EU) 2022/2554 on digital operational resilience for the financial sector (DORA)*. Regulation, 14 December 2022; applies from 17 January 2025 (Art 30(3)(b)). https://eur-lex.europa.eu/eli/reg/2022/2554/oj [Regulatory text]
62. Court of Justice of the European Union. *Judgment of 7 December 2023, OQ v Land Hessen (SCHUFA Holding), Case C-634/21*. ECLI:EU:C:2023:957. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:62021CJ0634 [Regulatory text (case law)]
63. Article 29 Data Protection Working Party. *Guidelines on Automated individual decision-making and Profiling for the purposes of Regulation 2016/679 (WP251rev.01)*. Adopted 3 October 2017, revised 6 February 2018; endorsed by the EDPB 25 May 2018. https://ec.europa.eu/newsroom/article29/items/612053 [Framework guidance]
64. P. Kasselman, J. Lombardo, Y. Rosomakho, B. Campbell, N. Steele, and A. Parecki. *AI Identity Management System*. IETF Internet-Draft draft-ietf-wimse-aims-00, 15 September 2026 (WIMSE WG document). https://datatracker.ietf.org/doc/draft-ietf-wimse-aims/ [Standard (draft)]
65. NIST. *SP 800-53 Control Overlays for Super Intelligence Systems (COSIS)*, formerly Control Overlays for Securing AI Systems (COSAiS). Project page, updated 20 October 2026; predictive AI overlay annotated outline, 8 January 2026. https://csrc.nist.gov/projects/cosis [Standard (pre-final)]

Colophon

This edition was drafted and revised with Claude (Anthropic), Claude Fable 5 for the version 4 releases and Claude Opus 5.5 for versions 5.0 and 5.1, working under the Draft & Approve pattern this paper describes in Section 5.3, p. 14: the model drafted, and the accountable author reviewed, modified, and approved every position. Version 5.0 also applied the paper's own verification rule to itself. Its changes began from an external review of version 4.5, and every new external claim was checked against a primary source before drafting, with claims that could not be verified cut or marked as reported. Framework decisions, worked examples, and errors are the author's. The same discipline applies to the Delegation Lab simulation at lab.vaom.ai, which Claude both helped build and narrates at runtime. We would find it strange to publish a framework about accountable AI delegation any other way.

Version History

VERSION	DATE	CHANGES
1.0	December 2025	Initial release. Seven-layer architecture, delegation authority matrix, worked example, regulatory alignment, 90-day roadmap.
2.0	February 2026	Added Delegation Gap scenario, "Why This Is Not Workflow Automation" section, expanded executive summary with scope and architectural neutrality.
2.5	March 2026	Added "How the Composite Confidence Score Works" (four-dimension scoring with weighted averages and hard floors) and "Managing Foundation Model Drift" (regression testing, threshold recalibration, shadow periods, model registry).
3.0	April 2026	Added Section 5, p. 11: Delegation Discovery & Design. Four-stage method (Decision Inventory, Authority Decomposition, Delegation Pattern Selection, Delegation Readiness Assessment). Introduced six named Delegation Patterns and five-dimension Authority Decomposition framework. Updated implementation roadmap and worked example to reference the discovery method.
3.5	April 2026	Added Section 2, p. 8: What VAOM Is (and What It Is Not), including credit risk decisioning analogy. Added hidden decisions guidance to Decision Inventory (Section 5.1, p. 11) with three discovery techniques. Expanded Pattern 6 (Coordinate & Escalate) with multi-agent failure modes: authority conflicts, cascading confidence erosion, and coordination state loss. Expanded Delegation Readiness Condition 5, p. 19 with guidance on navigating contested ownership. Added Implementation Challenges subsection to composite confidence scoring (Section 8, p. 29) addressing model calibration, anomaly noise, and rule-model conflicts. Added Calibration Anti-Patterns (five named anti-patterns with symptoms, root causes, and signals). Added three worked examples beyond finance: customer complaint escalation, AML transaction triage, and HR policy violation assessment.
4.0	July 2026	The enforcement release: "delegation boundaries that compile." Added seventh design principle (Boundaries Compile to Enforcement). Added Section 9, p. 34: Delegation Identity & Credentialing (delegated execution contexts, matrix-to-scope compilation, structural non-delegability through tool absence, agent identity lifecycle and registry). Added Dynamic Delegation & Authority Attenuation to Pattern 6 (attenuation rule, spawn-as-decision, chain accountability) plus a fourth multi-agent failure mode: delegation laundering. Added Section 10, p. 38: Continuous Assurance and the Guardian Function, a third cross-cutting concern covering enforcement telemetry, behavioral envelopes, circuit breakers, and the guardian-agent pattern with recursive delegation governance. Added Section 11, p. 40: The Delegation Scorecard, ten metrics across calibration, operational, and enforcement health. Added fifth confidence dimension: Independent Verification, with verifier-independence requirements and a verification-theater warning. Added worked example 7D, p. 27: Autonomous Incident Remediation (IT operations, dynamic sub-agents). Rewrote regulatory alignment for the post-Digital-Omnibus EU AI Act timeline (Annex III → 2 December 2027) and added mappings to Singapore's IMDA agentic framework, OWASP Agentic Top 10, NIST agent standards pipeline, and levels-of-autonomy research.
4.1	August 2026	The temporal compilation update. Added the temporal compilation path to Section 9, p. 34: prerequisite rules (Band B review as an approval event within a window), cumulative caps over rolling windows (closing the threshold-splitting gap), and rate conditions, with the event-history trust boundary stated. Extended principle 7 to temporal policy engines. Added temporal policy languages to the 2026 framework landscape (Section 2, p. 8) and to the alignment mappings (Section 13, p. 44). Noted the safety-versus-liveness boundary of runtime enforcement (Section 10, p. 38) and enforcement event streams as scorecard data sources (Section 11, p. 40). No architectural changes: no new sections, principles, layers, or metrics.

VERSION	DATE	CHANGES
4.5	September 2026	Landscape and alignment update. Refreshed the Delegation Gap evidence base with the 2026 agent incident record, survey data, and sharpened analyst guidance (Section 1, p. 4). Added the OWASP Agent Control Standard to the framework landscape and alignment mappings as the emerging runtime control-plane standard and a compile target for boundaries and guardians, disambiguated from Microsoft's identically abbreviated Agent Control Specification (Sections 2, p. 8 and 13). Added identity-layer delegation drafts (attenuating authorization tokens, OAuth actor profiles, OpenID AuthZEN) to the alignment mappings, formalizing the attenuation rule at the credential layer (Section 13, p. 44). Stated the composition boundary between temporal policy operators and the calibration and anomaly assurance layer (Section 13, p. 44) and named the standard runtime intervention points where suspensions execute (Section 10, p. 38). Noted IMDA MGF v1.5 and the pre-final status of the NIST agent control overlays, with the control-by-control mapping commitment kept open (Section 13, p. 44). No architectural changes. Revised September 2026: references section added, calibration heuristics labeled as illustrative defaults, composite score semantics and chain confidence independence clarified, regulatory alignment phrasing tightened, security scope stated, landscape descriptions corrected against primary sources, residual empirical and legal claims softened, worked-example authority justification made explicit, inventory-to-matrix mapping stated, roadmap preconditions stated, prior art acknowledged. The regulated-decision routing rule in Dimension 3, previously implicit in the worked examples, is now stated explicitly. No architectural changes.
5.0	October 2026	The proof release: "delegation you can prove." Added eighth design principle (Controls Are Proven, Not Presumed) with an operational definition of proof and a transition window for existing deployments. Section 8, p. 29: thresholds calibrated per decision type against declared error targets (Learn-then-Test), fully worked composite computation, verifier independence measured through the conditional miss rate. New Section 5.5, p. 21: deriving authority limits from expected loss, cumulative ceilings, and category overrides. Section 5.4, p. 18: Readiness Condition 5, p. 19 gains a shared form with a stop rule; new Readiness Condition 6, p. 20 (oversight funded and effective) as a hard gate, with compliance limits on seeded-error testing. Section 9, p. 34: the delegation lease for continuously operating agents (renewal that only narrows, maximum lifetime, re-attestation, degradation rule) and boundary verification. Section 12, p. 42 rewritten as Governing Learning and Model Change (tenant knowledge, feedback loops, vendor model change, traceability). Section 11, p. 40: scorecard revised to thirteen metrics in three groups, with an oversight-health group and a stated cut list. New worked example 7E, p. 28: Delegated Purchasing; worked example 7A, p. 24 shows its matrix entries. Section 13, p. 44 rewritten as Regulatory Design Obligations; dated landscape and alignment content moved to the new VAOM Alignment Annex, including GDPR Article 22 and UK, US, and Asia-Pacific mappings. Section 1, p. 4 incident record updated to September 2026. Version history dates for 1.0 and 2.5 corrected.
5.0.1	October 2026	Corrections release, no change to the framework. References updated for facts that changed after the 27 September 2026 verification: Agent Control Standard repository tags v0.1.3 and v0.1.4; NIST's overlays project renamed COSIS. Delegation Lab address corrected to lab.vaom.ai. The Alignment Annex was revised on 31 October 2026 for the same changes and for EN 18286's citation in the Official Journal and the October 2026 AIUC-1 release.
5.1	November 2026	Added "Inherited Governance and How It Fails" to Section 1, p. 4: five failure modes of governing agents through inherited permissions and approval chains, organized by mechanism, each mapped to the VAOM construct that answers it. Cited convergent formal work on agent authorization in Section 9, p. 34. Alignment Annex: survey of how platforms govern their agents.

© 2026 Verkflöde AB · Released under CC BY 4.0 · verkflode.com/vaom