VERKFLÖDE

Controlled AI Delegation

The Verkflöde Agent Operating Model (VAOM)

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

// Delegation boundaries that compile.

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

Version 4.5 · September 2026

Verkflöde AB · An Open Framework for Enterprise AI Delegation · verkflode.com

VAOM 4.5 · WHITEPAPER

Contents

Executive Summary

Enterprise AI is shifting from experimentation to operational delegation. Organizations are no longer testing models in isolation; they are embedding AI agents into workflows that influence financial approvals, contract decisions, HR processes, compliance monitoring, and customer operations. That shift creates a structural problem. Probabilistic decision-making now runs inside systems that were built for deterministic automation.

Most enterprises already possess governance frameworks covering AI usage, model risk classification, and compliance oversight. These frameworks define what is allowed in principle. What they rarely define is how AI systems exercise decision authority within live workflows, and the absence of explicit delegation design 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 AI systems are allowed to act within enterprise workflows. VAOM is implementation-agnostic, working across AI vendors, orchestration platforms, and enterprise systems to establish controlled delegation with measurable outcomes and audit-ready accountability.

VAOM applies wherever AI participates in decisions with operational, financial, or regulatory consequences: invoice processing, contract review, compliance monitoring, HR approvals, customer operations. It is not designed for low-stakes use cases such as internal chatbots or productivity assistants, where delegation authority is not a concern.

VAOM describes what controls must exist, not which products implement them. The seven layers define functional requirements, and any combination of orchestration platforms, AI vendors, enterprise systems, and observability tools can satisfy them. This neutrality is what keeps the framework durable as the technology underneath it changes.

Version 4.0 extends the framework from *declared* boundaries to *enforced* boundaries. A boundary that is written down but not technically bound to the agent protects nothing when tested. VAOM 4.0 therefore defines how the Delegation Authority Matrix compiles downward into agent identity, scoped credentials, and tool-level permissions (Section 9); how authority attenuates through dynamically spawned sub-agents (Section 5.3); how boundaries are continuously assured at runtime, including the guardian-agent function (Section 10); and how delegation health is measured through a standard scorecard (Section 11). The composite confidence score gains a fifth dimension, independent verification (Section 8). The regulatory alignment reflects the post-Omnibus EU AI Act timeline together with the agentic governance frameworks published since early 2026 (Section 13).

Version 4.1 adds the temporal compilation path: matrix rows whose meaning is inherently sequential (act only after review, cumulative value limits, frequency ceilings) now compile to runtime-enforced temporal rules (Section 9), with the framework landscape and alignment mappings updated for the temporal policy languages that reached the market in 2026 (Sections 2 and 13).

Version 4.5 is a landscape and alignment update. It refreshes the Delegation Gap evidence base with the 2026 incident record, brings the runtime agent control standards and identity-layer delegation drafts that emerged over the summer of 2026 into the framework landscape and alignment mappings (Sections 2 and 13), and states the composition boundary between temporal policy enforcement and VAOM's calibration and assurance layers. Nothing architectural changes: no new sections, principles, layers, or metrics.

This paper outlines the Delegation Gap, the principles of Delegation-First design, the VAOM layered control structure, a structured method for delegation discovery and design (including six named delegation patterns and a five-dimension authority decomposition framework), five worked examples, the mechanics and implementation challenges of composite confidence scoring, the identity and credentialing model that enforces delegation boundaries, continuous assurance, the delegation scorecard, regulatory alignment, and a 90–day implementation roadmap.

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 in 2025–2026 converged on the same 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, while Forrester's June 2026 assessment of agentic AI found that more than half of enterprises report governance gaps driving agentic sprawl, even after adopting the NIST AI RMF. By mid-2026 the diagnosis had sharpened. Gartner warned that treating agent governance as binary, either locked down or fully trusted, is itself a root cause of failure and that governance should be proportional to each agent's autonomy level; its guardian-agent research predicts that through 2028 at least 80 percent of unauthorized agent transactions will come from internal policy violations, oversharing, unacceptable use, misguided behavior, rather than from malicious attacks. Organizations are not abandoning these projects because the models are incapable. They are abandoning them because nobody structured what the agents were allowed to decide, and the incidents, review bottlenecks, and unquantifiable risk that followed made the projects indefensible.

The incident record now backs the projections. Through 2025 and 2026 a series of publicly documented cases showed agents deleting production databases and destroying infrastructure after locating credentials that were never meant to be theirs. In the most widely reported, from April 2026, a coding agent 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. 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. Across these cases the boundary that failed existed as an instruction, not as an enforced technical constraint.

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

GOVERNANCE	OPERATIONAL DELEGATION
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.

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 rapidly, and VAOM's position within it should be stated precisely. Singapore's IMDA published the first government framework specifically for agentic AI (January 2026, updated to version 1.5 in May 2026), defining what good agentic governance looks like: bounded risk, meaningful human accountability, technical controls, and tiered autonomy. The OWASP Agentic Security Initiative published its Top 10 for Agentic Applications, cataloguing what can go wrong, from agent identity and privilege abuse to memory poisoning. The Cloud Security Alliance published an agent identity and access management framework defining the zero-trust substrate on which agent actions execute. NIST's AI Agent Standards Initiative and its forthcoming SP 800–53 control overlays for single-agent and multi-agent systems will define which controls must exist in federal-grade terms.

The enforcement layer has now industrialized as well. In August 2026 the first temporal policy language for agent authorization shipped from a major cloud provider: AWS's Dogwood, an open extension of Cedar integrated with Amazon Bedrock AgentCore, enforcing rules over *sequences* of agent actions (prerequisites, cumulative limits, rates) at a gateway the agent cannot argue with. The control plane began standardizing shortly after: the Agent Control Standard, a vendor-neutral specification for runtime enforcement hooks, inline guardian verdicts, and agent telemetry, public since May 2026, was adopted into the OWASP GenAI Security Project in September 2026. Agent identity industrialized in parallel, with Microsoft's Entra

Agent ID bringing lifecycle governance and conditional access to agent identities, and the agent protocol layer consolidated under the Linux Foundation's Agentic AI Foundation with the major model and cloud providers as members.

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 maps the relationships explicitly.

SECTION 3

Core Design Principles

Seven 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.

Why This Is Not Workflow Automation

Traditional workflow automation, including RPA and rules-based orchestration, executes predefined logic where authority is embedded in the rules themselves. If the rule triggers, the action was already authorized by the person who wrote it. There is no ambiguity about what the system is allowed to do, because it can only do what it was explicitly programmed to do.

AI agents are fundamentally different. They interpret context, weigh evidence, and produce probabilistic outputs that may vary across identical inputs, which means the system exercises something 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?"

VAOM exists because that second question has no equivalent in deterministic automation. 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), 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). Rollback and compensation patterns defined for every write operation.
L5	**Knowledge & Learning**	Separates read-only business data from tenant-owned agent knowledge. Captures feedback, validates updates against regression suites, and promotes through controlled release.
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), 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 calibrated against historical outcomes and periodically 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 adopting AI agents routinely skip delegation design entirely: select a workflow, connect an agent, discover the boundaries when something goes wrong. A modified contract term auto-approved. An escalation path that was never defined. A decision category no one realized the agent could reach. The Delegation Gap described in Section 1 is, among other things, a discovery problem, and you cannot govern what you have not identified.

VAOM addresses this through a structured Delegation Discovery & Design method, a repeatable process that organizations use to identify decision points, evaluate delegation fitness, design authority boundaries, and produce 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. The final output is a completed Delegation Authority Matrix (Section 6) 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, 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:

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

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:

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.

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.

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

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.

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).

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.

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.

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, 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.

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.

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.

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.

Record the assessment for each decision point:

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

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, 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) 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. 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) 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. *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. When agents operate in sequence, each agent's output becomes the next agent's input. Small errors or low-confidence outputs early in the chain can compound: an extraction agent that misreads a field at 0.88 confidence feeds a validation agent that passes it at 0.91 confidence,

which feeds a decision agent that auto-approves at 0.93 confidence. Each individual score looks acceptable; the aggregate reliability may not be. Multi-agent chains should track cumulative confidence across the full chain, not just per-agent scores. If the product of individual confidence scores falls below a chain-level threshold, the entire decision routes to human review regardless of individual scores.

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 applies to chains as it does to verifiers.

Coordination state loss. When handoffs between agents lose context, such as a field that was flagged as uncertain by one agent but not passed through the handoff protocol, downstream agents make decisions on incomplete information without knowing it is incomplete. Handoff protocols must define which state is transferred, which is dropped, and what metadata (including uncertainty markers and provenance) must survive the transition.

Delegation laundering. A decision that would be blocked or escalated under one agent's boundaries is routed through another agent whose boundary definitions are looser, and emerges executed. No individual agent violated its own authority, but the decision escaped the constraint it was designed to hit. This is the multi-agent equivalent of the Exception Graveyard anti-pattern (Section 8): each routing choice looks reasonable in isolation while the aggregate effect defeats the delegation design. 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). 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 named accountable human (Readiness Condition 5) is 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 delegable 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).

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 five readiness conditions. All five 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 what historical data supports the threshold calibration.

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: A named human is accountable. A specific role (not a team, not a committee, not "management") is identified as accountable for outcomes produced by this delegation. That role has the authority to override, suspend, or modify the delegation. The role is documented, and the person in it knows they hold this accountability. *Evidence required:* Named accountable role in the delegation design, confirmation that the role holder has accepted accountability, override mechanism documented.

In practice, satisfying Condition 5 is often the hardest part of the readiness assessment. Authority in enterprises is frequently contested, shared across functions, or deliberately left ambiguous. Invoice approval rules may sit between Finance, Procurement, and Legal. Vendor exceptions may be owned regionally in some contexts and globally in others. Nobody may want to own AI-delegated decisions because accountability for automated outcomes feels riskier than accountability for manual ones.

VAOM cannot resolve contested ownership by design, but it makes the contest visible. If no one is willing to be named as accountable for a delegated decision, that decision is not ready for delegation, however technically capable the agent is. The readiness assessment forces the conversation the organization may have been avoiding. Three approaches help: escalate the accountability question to the executive sponsor before the design phase begins (not during it), frame accountability as override authority rather than blame assignment (the accountable role is the one that can suspend the delegation, not the one that gets fired if it fails), and start with decision types where ownership is already clear, building trust before attempting decisions where authority is disputed.

The Delegation Readiness Assessment produces one of three outcomes:

Ready. All five 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.

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.

From Discovery to Matrix

The four stages of Delegation Discovery & Design feed directly into the Delegation Authority Matrix (Section 6). 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.

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. 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.

A Delegation Authority Matrix maps specific decision types to permitted levels of autonomy under defined confidence and risk conditions. This matrix makes delegation explicit, auditable, and adjustable.

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. 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 threshold emerged from Authority Decomposition (consequence scope analysis). 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.

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.

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 (high band). Since the invoice is below €5k and confidence is high, Band A routing is selected: auto-approve.

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 had already been distilled into a threshold refinement (0.90 to 0.92), validated against 200 historical invoices, and promoted.

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.

Total time from email arrival to ERP posting: under 90 seconds. Audit evidence available within minutes of a regulatory 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.

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.

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.

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.

Layer 4, Controlled Execution. The classification is recorded in the CRM. The acknowledgment email is sent only after human approval. The case is routed to the fraud investigation queue with full context attached.

Layer 5, Knowledge & Learning. The complaint joins the anonymized complaint corpus used for classification training. A pattern of similar unauthorized-charge complaints over the past month is flagged for the fraud team's attention, not as an automated action but as a data signal surfaced through the monitoring pipeline.

Layer 6, Governance & Control. The agent's CRM access is scoped to complaint-related records only. The regulatory reporting flag is logged with the compliance officer's determination, creating an audit trail for FCA inquiries. All customer data access is purpose-limited and retention-scheduled.

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.

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. 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.

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. The orchestrator assigns the alert a priority score based on regulatory risk (structuring patterns are high-priority under the Fourth Anti-Money Laundering Directive). SLA timer starts: high-priority alerts must be triaged within 4 hours.

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) 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.

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 feed back into the system: confirmed false positives refine the auto-dismissal criteria; confirmed suspicious activity refines the typology models. Every refinement follows the change control process. A recent regulatory guidance update on virtual asset service providers was incorporated as a new typology rule, validated against 6 months of historical alerts, and promoted after compliance sign-off.

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.

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.

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. The orchestrator creates a case record with restricted visibility. It enriches the case with prior complaints involving either party (if any exist) and the applicable local employment law requirements. In the EU, works council notification requirements are checked. SLA timer starts per internal policy.

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.

Layer 4, Controlled Execution. The only automated execution is the case record creation and the notification to the designated HR case handler. All subsequent actions (investigation initiation, witness outreach, outcome determination) are human-executed, with the agent providing preparation and context assembly.

Layer 5, Knowledge & Learning. Case outcomes are stored in an anonymized, access-restricted knowledge base used to improve classification accuracy and to surface patterns (e.g., repeated complaints about the same team or location). Learning updates to the classification model follow enhanced change control: HR Legal must approve any changes to how complaints are categorized, given the employment law implications.

Layer 6, Governance & Control. Access controls are unusually strict: the case is visible only to the assigned HR case handler, the designated investigator, and HR Legal. The agent's data retrieval is purpose-limited to the specific case. GDPR data subject rights apply: both the complainant and the respondent have rights regarding their personal data in the case file. The audit log records every data access, every document generated, and every human decision, but the log itself is access-restricted.

Layer 7, Human Oversight. Humans are in the loop at every substantive decision. The agent's value is in preparation and pattern recognition, not decision-making. The HR case handler reviews the classification, scopes the investigation, and determines the process. The investigator conducts interviews and gathers evidence. The decision-maker (senior HR with Legal input) determines the outcome. The agent never recommends discipline, never contacts witnesses, and never communicates outcomes.

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

IT operations is the largest real deployment category for agentic AI in 2026, and the one where dynamic sub-agent spawning is already routine. This example exercises the version 4.0 additions end to end: identity-bound authority (Section 9), sub-agent attenuation (Section 5.3), chain-level confidence, and continuous assurance (Section 10).

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.

Layer 1, Trigger & Intake. At 02:14, monitoring 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).

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).

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 incident signature, diagnosis, and outcome join the incident knowledge base. The connection-pool pattern is proposed as a new detection rule, to be validated against six months of incident history before promotion through change control.

Layer 6, Governance & Control. The audit log records the full chain: activation context, both spawn events with derived scopes, each sub-agent's findings with provenance, the composite score breakdown, the pre-flight verification result, the restart execution with per-instance timestamps, and the config-change proposal. Every entry carries the tuple of accountable human role, agent identity, and incident ID. The sub-agents' credentials expired before the incident closed; nothing persists.

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.

Continuous Assurance (Section 10) 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 8

How the Composite Confidence Score Works

The Confidence Gate (Section 4, 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.

Implementation Challenges and Calibration Risks

The composite confidence model is architecturally sound but non-trivial to implement well. Four challenges deserve honest acknowledgment:

Model confidence is often poorly calibrated. Foundation models frequently produce high-confidence outputs that are incorrect and low-confidence outputs that are correct. 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 VAOM treats model certainty as one of five dimensions rather than the sole input, and why the hard floor mechanism exists: a single unreliable signal cannot override the composite. Organizations implementing the Confidence Gate should calibrate model certainty against ground truth data for their specific decision types, not rely on the model's self-reported probabilities.

Anomaly detection generates noise. Anomaly signals are powerful negative modifiers, but real-world anomaly detectors produce false positives. An unusual invoice timing may reflect a vendor's changed billing cycle, not fraud. A new vendor format may simply be a new vendor. If anomaly signals are weighted too heavily, the Confidence Gate routes an excessive proportion of decisions to human review, creating the bottleneck problem: review queues grow, SLAs break, and humans become slower than the process was before the agent was introduced. Threshold calibration must account for the false positive rate of each anomaly signal, and organizations should expect to recalibrate anomaly weights multiple times during the pilot phase.

Rules and model reasoning can conflict. When deterministic business rules contradict the model's probabilistic assessment, the composite score must resolve the conflict. VAOM's design gives rule match strength its own dimension precisely for this reason, but the resolution policy must be explicit. In most implementations, a hard rule failure (vendor not on approved list, amount exceeds contract ceiling) should cap the composite score at Band B or below regardless of model certainty. The alternative, letting high model confidence override a failed rule, defeats the purpose of having rules.

Verification can become theater. The independent verification dimension only adds signal if the verifier can actually fail. A verifier that approves 99.9 percent of proposals is either verifying trivial properties or is not independent. Organizations should track verifier disagreement rates: 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, either tighten what verification checks or stop counting it as a dimension. An always-green signal inflates the composite exactly the way poorly calibrated model certainty does.

The Confidence Gate is not a plug-and-play component. It requires investment in calibration data, iterative threshold tuning, and ongoing monitoring of score distributions. The framework provides the architecture; the organization must provide the engineering discipline to make it reliable. Implemented well, it is designed to produce better outcomes than either pure automation or pure human review, and the scorecard (Section 11) is how an organization verifies whether it does. Implemented carelessly, with uncalibrated thresholds and unweighted dimensions, it becomes either a bottleneck that routes everything to humans or a rubber stamp.

Calibration Anti-Patterns

Miscalibrated confidence thresholds are the failure cause VAOM anticipates as the most common in implementation. These anti-patterns represent failures of delegation design and governance discipline, not failures of AI capability. The architecture works correctly, the layers are wired, the audit logs are flowing, but the thresholds are wrong, and the system either produces no value (everything goes to humans) or produces uncontrolled risk (too much is auto-approved). The following anti-patterns describe what bad calibration looks like and what signals indicate recalibration is needed. The numeric signals attached to them (band shares, approval rates, exception shares, correlation limits) are illustrative starting defaults drawn from practice, to be calibrated per workflow; they are not empirically derived thresholds.

Anti-pattern 1: The Review Queue Flood. Symptom: 60 to 80 percent of decisions route to Band B (human review). Review queues grow. SLAs degrade. Humans begin rubber-stamping approvals to clear the backlog, which is worse than not having the system at all because rubber-stamping creates the illusion of oversight without its substance. Root cause: thresholds set too conservatively, often because the implementation team defaulted to "safe" settings without calibrating against historical data. The anomaly dimension may also be over-weighted, treating normal business variation as suspicious. Signal to watch: if the human override rate on Band B decisions exceeds 90 percent (meaning humans approve more than 90 percent of what the agent flagged for review), the threshold is too strict. The agent is not adding judgment; it is adding delay.

Anti-pattern 2: The Confidence Mirage. Symptom: the system auto-approves at high rates and the composite scores look healthy, but post-execution audits reveal error rates significantly above expectations. Root cause: over-reliance on model certainty without adequate weight on rule match and data completeness. The model reports high confidence, but the confidence is poorly calibrated for this specific decision type. The composite score inherits the model's overconfidence because the weighting policy gives model certainty too much influence. Signal to watch: if the composite score distribution clusters tightly above the Band A threshold (e.g., 85 percent of scores fall between 0.92 and 0.98), the scoring lacks discrimination. A healthy distribution shows meaningful spread across the bands, reflecting genuine variation in decision difficulty.

Anti-pattern 3: The Exception Graveyard. Symptom: the Delegation Authority Matrix has accumulated a long list of hardcoded exceptions, special cases, and manual overrides that bypass the Confidence Gate. Each exception was added to handle a specific situation, but collectively they have created a shadow routing system that the Confidence Gate does not govern. Root cause: instead of recalibrating thresholds or adjusting dimension weights when the gate produces incorrect routings, the team adds exceptions. Each exception is individually reasonable, but the aggregate effect is that the Confidence Gate governs a shrinking proportion of decisions while the exception list grows. Signal to watch: if more than 15 percent of decisions are routed through exception rules rather than through the standard composite scoring path, the scoring policy itself needs revision, not more exceptions.

Anti-pattern 4: The Stale Threshold. Symptom: the system performed well during pilot but accuracy has degraded over 3 to 6 months of production operation. Root cause: the business has changed (new vendors, new pricing structures, seasonal patterns, updated regulations) but the thresholds, dimension weights, and anomaly baselines have not been recalibrated. The Confidence Gate is evaluating current decisions against historical calibration that no longer reflects the operating environment. Signal to watch: a

gradual shift in the band distribution over time. If the percentage of Band A decisions is drifting upward or downward without a corresponding change in business volume or complexity, the thresholds are drifting out of alignment with reality.

Anti-pattern 5: The Dimension Collapse. Symptom: the composite score behaves almost identically to a single dimension, usually model certainty. The other four dimensions (rule match, data completeness, anomaly signals, independent verification) are either not implemented, not weighted meaningfully, or always return high scores. Root cause: the implementation team built the model certainty pipeline first and either deferred the other dimensions or implemented them as placeholders that return constant values. The composite score exists in name but not in practice. Signal to watch: calculate the correlation between the composite score and each individual dimension. If any single dimension has a correlation above 0.95 with the composite, the other dimensions are not contributing meaningful signal and the gate is effectively single-dimensional.

These anti-patterns can be expected to emerge within the first 3 to 12 months of production operation. Organizations implementing the Confidence Gate should monitor for these signals as part of the ongoing governance process described in Layer 6, and should schedule formal recalibration reviews 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 accountable for this delegation (Readiness Condition 5)
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

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.

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:

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).

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. 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: 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), 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) 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. Enterprise identity platforms, agent identity services, and authorization standards (OAuth 2.1–family token exchange, workload identity frameworks, policy-as-code engines) are evolving quickly, and organizations should implement these requirements with whatever their identity stack supports. The role division is stable even as the products change: identity platforms enforce boundaries; VAOM is where the boundaries come from. An identity system can prove *that* an agent was allowed to act. Only delegation design can answer *why* it was allowed, and produce the matrix entry, the accountable role, and the readiness evidence behind the scope.

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). 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 4.0 requires both.***

SECTION 10

Continuous Assurance and the Guardian Function

VAOM's controls so far operate at three tempos: design time (Delegation Discovery & Design), decision time (the Confidence Gate), and periodic review (sampling audits, recalibration, drift management). Version 4.0 adds the missing tempo, *runtime*: the continuous verification that agents are actually operating within the boundaries the other controls define, 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, every tool-manifest miss, every rejected credential exchange. A single denial may be noise. A pattern of an agent repeatedly requesting authority it does not hold is an early signal of misalignment between the agent's behavior and its delegation design, or of an attempted prompt injection steering the agent toward its boundary.

Behavioral envelope. Each agent's operating profile under normal conditions: its band distribution, its tool-call sequences, its spawn patterns, its data-access footprint. Deviation from the envelope (an agent that historically routes 60 percent of decisions to Band A suddenly routing 95 percent, an unfamiliar tool sequence, a spawn fan-out spike) triggers alerting before any individual 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.

Circuit Breakers

When defined limits are crossed, Continuous Assurance suspends: it revokes the delegated execution context, halts the affected chain in a rollback-safe state, and pages the accountable human with the evidence assembled. Suspension is deliberately designed to be *delegable*: fully reversible (a wrongly suspended agent is resumed by a human in minutes), internally scoped, with excess caution as its failure mode. The asymmetry is the point. 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), envelope limits and circuit breaker conditions gain a stronger implementation: expressed as temporal prohibition rules, they move from monitoring-and-alerting to pre-execution enforcement, refusing the call that would cross the limit rather than paging a human after it did. One distinction from runtime verification theory keeps expectations honest: temporal

enforcement handles *safety* properties (this must not happen, or must not happen 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: the guardian function watches for it, and escalation SLA attainment (Section 11, metric 7) measures it.

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. 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. It operates under Monitor & Intervene (Pattern 5), applied recursively: continuous operation, detection capabilities exceeding human capacity at scale, defined intervention paths.

Three boundaries keep the recursion honest:

Guardians contain; they never expand. A guardian's authority is limited to observation, alerting, and suspension: actions that narrow other agents' effective authority. A guardian may never approve, execute, or extend another agent's actions. This keeps the guardian's own delegation profile in the fully-reversible, internally-scoped quadrant where autonomous operation 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 applies. A named 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 anti-patterns of Section 8 each name a signal to watch. Version 4.0 consolidates them, plus the enforcement signals from Sections 9 and 10, into a standard scorecard: ten metrics that together answer the question *is our delegation healthy?* as a dashboard, not a debate. Each metric has a healthy range, and each range breach maps to a named diagnosis. A range 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. The healthy ranges quoted below carry over the illustrative starting defaults of Section 8: they are calibrated per workflow at design time and are not empirically derived thresholds.

Calibration health

#	METRIC	HEALTHY SIGNAL	BREACH INDICATES
1	**Band distribution vs design target**	Band A/B/C shares within the tolerance set at design	Review Queue Flood (B too high) or authority creep (A too high)
2	**Band B approval rate**	Humans meaningfully modify or reject a visible share of reviews; approval rate below ~90%	Above 90%: thresholds too strict; the agent adds delay, not judgment
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 disagreement rate**	Measurably above zero on verified decision types	Near zero: verification theater; an always-green signal is inflating the composite

Operational health

#	METRIC	HEALTHY SIGNAL	BREACH INDICATES
6	**Band-distribution drift**	Stable against calibration baseline across reporting periods	Stale Threshold: business changed, calibration didn't
7	**Escalation SLA attainment**	Escalations reach an authorized human within defined SLA, near 100%	Broken escalation paths; Readiness Condition 3 no longer holds
8	**Band A audit error rate**	Post-execution sampling error rate at or below the rate accepted at design	Confidence Mirage: composite scores no longer predict correctness

Enforcement health

#	METRIC	HEALTHY SIGNAL	BREACH INDICATES
9	**Scope-denial rate per agent**	Low and stable; spikes investigated	Agent behavior diverging from delegation design, or active injection attempts

#	METRIC	HEALTHY SIGNAL	BREACH INDICATES
10	**Chain integrity incidents**	Zero attenuation violations; chain depth/fan-out within matrix limits	Dynamic delegation operating outside its designed envelope

The enforcement-health metrics have a natural data source: the enforcement layer's own event stream. Every decision record, scope denial, and temporal-rule refusal is a scorecard input, whether it originates in a hosted policy engine or a local decision log, and emerging agent telemetry conventions are making those streams portable across runtimes. Where temporal rules are deployed (Section 9), scope-denial rate (metric 9) extends to temporal-denial rate with the same reading: a spike means agent behavior diverging from delegation design, or a probe.

Two disciplines make the scorecard useful rather than decorative. First, thresholds are set at design time, as part of the Delegation Authority Matrix sign-off, so that a breach is a governance event with a pre-agreed owner and response, not a number someone notices. Second, the scorecard is reviewed on the same cadence as recalibration (quarterly at minimum during the first year), and every review is logged. The scorecard is itself audit evidence that oversight is operating, which is precisely what regulators examining "effective human oversight" ask organizations to demonstrate.

SECTION 12

Managing Foundation Model Drift

A distinct challenge for AI-enabled workflows is foundation model drift: the risk that an underlying AI model changes, through provider updates, version deprecation, or fine-tuning adjustments, in ways that invalidate previously calibrated confidence thresholds and delegation boundaries.

VAOM addresses this through Layer 5 (Knowledge & Learning) and Layer 6 (Governance & Control) working together. When a foundation model change is detected or announced, the following control sequence applies:

Regression testing. The updated model is evaluated against the same historical decision set used to calibrate the original thresholds. Outputs are compared for consistency, confidence distribution shifts, and decision boundary changes.

Threshold recalibration. If regression testing reveals material drift (confidence distributions shifted, previously auto-approved decisions now falling below threshold, or vice versa), thresholds are recalibrated against the new model's behavior. Recalibration follows the same change control process as any threshold modification.

Shadow period. Before the updated model enters production, it runs in parallel with the existing model for a defined period. Decisions are logged but not executed. Divergences between old and new model outputs are reviewed by the designated human authority.

Model registry. Every model version used in production is recorded in a registry with its calibration data, regression test results, shadow period outcomes, and the human approval that authorized its promotion. This creates an auditable chain from model version to threshold policy to delegation authority. The model registry and the agent registry (Section 9) are two views of the same discipline: a model version change alters what an agent identity is capable of, so the registries cross-reference: every agent identity records which model versions it depends on, and a model promotion triggers review of every agent that inherits it.

The principle is straightforward: a model change is a change to the delegation system. It is governed by the same change control discipline as any other modification to the operating model. Provider convenience does not override organizational governance.

SECTION 13

Regulatory Alignment 2026–2028

VAOM embeds regulatory obligations into operational design so that operational 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.

The EU AI Act After the Digital Omnibus

The regulatory timeline changed materially in 2026 and any delegation program should plan against the current dates. The EU's Digital Omnibus package, formally adopted in June 2026, postponed the application of high-risk AI system obligations: stand-alone high-risk systems under Annex III now apply from **2 December 2027** (previously 2 August 2026), and AI embedded in regulated products under Annex I from **2 August 2028**. Transparency marking obligations for AI-generated content under Article 50(2) were deferred to 2 December 2026 for generative systems already on the market before 2 August 2026, and new prohibitions were added to Article 5. Obligations for general-purpose AI models have applied since August 2025 and were not deferred.

The deferral should be read carefully. It moves the application dates; it does not eliminate the value of establishing evidence and controls before they arrive. Article 14 (human oversight), Article 13 (transparency), and Article 9 (risk management) survive intact. Organizations that treat the window as design time (running delegation discovery, building the matrix, wiring identity enforcement, and accumulating evidence) will meet December 2027 with two years of audit trail. Organizations that treat it as relief will meet the same deadline with a backlog and no operating history. VAOM's position is unchanged by the dates. Delegation design is not something one does because a regulation demands it in August or December; it is what makes agent deployment defensible in the incident review that can happen any Friday afternoon.

Core European Regulatory Mapping

The model provides operational controls and evidence that can support obligations under three key European regulatory frameworks, subject to the specific system classification and processing context:

REGULATION	KEY REQUIREMENTS	VAOM LAYERS PRODUCING SUPPORTING EVIDENCE
GDPR	Data minimization, purpose limitation, transparent decision logic, right to explanation	L1 (PII detection, minimization at entry), L3 (purpose-limited retrieval, traceable reasoning), L6 (immutable decision logs, access audit trail)
DORA	Operational resilience, change control, third-party risk management, audit evidence generation	L2 (retry/escalation/SLA), L4 (idempotent execution, rollback), L5 (versioned learning, regression tests), L6 (evidence packs, incident runbooks)
NIS2	Security governance, monitoring, incident response, accountability	L6 (RBAC/ABAC, policy versioning, real-time alerting, SLO monitoring), L7 (override logging, escalation paths), L4 (secrets management, zero-trust auth)

Complementary Frameworks

VAOM does not replace existing enterprise frameworks. It complements them by operationalizing AI-specific delegation controls within structures that organizations already maintain:

EU AI Act: VAOM's confidence gate and human oversight layers provide operational controls and evidence that can support the Act's obligations for high-risk AI systems, particularly human oversight (Article 14), transparency (Article 13), and risk management (Article 9), subject to the specific system classification and processing context. The Delegation Discovery method (Section 5) provides a structured process for identifying the decisions that may fall within the Act's scope and for designing controls around them; the classification itself remains a legal determination outside VAOM. The Delegation Scorecard (Section 11) produces evidence that can support the Article 14 expectation that oversight be *effective*, not nominal: metrics 2 and 8 detect exactly the rubber-stamping and automation-bias failure modes that turn human review into theater.

NIST AI RMF and the agent standards pipeline: VAOM provides operational controls and evidence that can support the Govern, Map, Measure, and Manage functions at the workflow level, adding operational specificity where the NIST framework provides strategic guidance. NIST's AI Agent Standards Initiative and its forthcoming SP 800–53 control overlays for securing AI systems, including dedicated overlays for single-agent and multi-agent deployments, will define control catalogs at the system level; VAOM's artifacts (the matrix, execution contexts, decision traces, the scorecard) are designed to produce workflow-level evidence that can support those controls, subject to the specific system classification and processing context. The overlays remain pre-final: the first of the series (predictive AI) circulated as an annotated outline in January 2026, with the single-agent and multi-agent overlays further back in the queue. The commitment stands, and a control-by-control mapping will be published alongside the comparative mapping document against the first public draft of the agent overlays.

ISO 42001: VAOM's governance layer (L6) provides operational controls and evidence that can support the management system requirements, while the learning pipeline (L5) can support continual improvement, subject to the scope of the organization's management system.

Alignment with the 2026 Agentic Frameworks

Singapore's Model AI Governance Framework for Agentic AI (IMDA, January 2026; version 1.5, May 2026) is the first government framework specifically for AI agents, and its structure parallels VAOM's architecture dimension by dimension. The v1.5 update added multi-agent systemic-risk factors and automation-bias guidance that lands on the same failure modes VAOM's calibration anti-patterns name (Section 8):

IMDA DIMENSION	VAOM MECHANISM
Assess and bound risks upfront	Delegation Discovery & Design (Section 5); Authority Decomposition
Make humans meaningfully accountable	Readiness Condition 5; named accountable roles; Layer 7
Technical controls and processes	Confidence Gate; identity-bound scopes (Section 9); Continuous Assurance (Section 10)
Tiered autonomy (fully automated / human-approved / off-limits)	Bands A / B / non-delegable, with the added confidence dimension and decision-level granularity

OWASP Top 10 for Agentic Applications (2026): OWASP catalogs the threats; VAOM's structures are among the controls. Agent identity and privilege abuse is countered by delegated execution contexts and scope compilation (Section 9); excessive-autonomy risks by the matrix and band ceilings; multi-agent exploitation paths by attenuation, chain limits, and delegation-laundering defenses (Section 5.3). Security teams should treat the OWASP taxonomy as an adversarial test suite for a VAOM implementation: every Top 10 entry is a question the delegation design should already answer.

Levels-of-autonomy research: Academic work on agent autonomy (notably Feng, McDonald & Zhang's five-level framework, which characterizes autonomy by the user's role: operator, collaborator, consultant, approver, observer) arrives at a structure recognizably parallel to VAOM's delegation patterns: Prepare & Present places the human as operator; Draft & Approve as approver; Execute & Audit and Monitor & Intervene as observer. The research proposal of "autonomy certificates" (formal statements of the maximum autonomy an agent may operate at) has a natural implementation substrate: a signed Delegation Authority Matrix entry, compiled into scopes, is an autonomy certificate with enforcement attached.

Temporal policy languages (2026): The first runtime enforcement language for agent action sequences from a major provider is AWS's Dogwood (August 2026), an open (Apache 2.0) extension of Cedar integrated with Amazon Bedrock AgentCore Policy and grounded in metric first-order temporal logic. The mapping to VAOM is direct, and it runs in one direction: VAOM designs the row, the engine enforces it.

VAOM MECHANISM	TEMPORAL POLICY IMPLEMENTATION
Band B human review	Prerequisite approval event within a window (`formerly within`)
Cumulative value thresholds	Windowed sum caps (`sum_within`)
Frequency and volume limits; circuit breakers	Rate rules (`count_within`, `count_distinct_within`)
Delegation Authority Matrix row	The policy someone must first design

What the engine does not decide is the point: which decisions exist in the workflow, which of them are delegable, at what confidence, under whose accountability, and whether the resulting oversight is calibrated or theater. Every temporal policy presupposes that someone already made those calls; VAOM is the method that makes them, and Section 9's temporal compilation path is where the matrix meets this class of engine. Two caveats carry over from the provider's own guidance: temporal enforcement depends on a complete, authenticated event history (Section 9), and the open reference interpreter is for exploration, with production enforcement living in the hosted service.

One composition boundary is worth stating. Temporal operators compile the deterministic slice of assurance: counts, sums, and sequence prerequisites. They do not subsume the calibration and anomaly work of Sections 8, 10, and 11. A rate rule can refuse the eleventh call in a window; it cannot notice that a verifier has stopped disagreeing or that a band distribution is drifting. The layers compose, and the scorecard reads both.

Runtime agent control standards (2026): The Agent Control Standard (ACS, specification v0.1.1), public since May 2026 and adopted into the OWASP GenAI Security Project in September 2026, is a vendor-neutral specification for the runtime control plane. It instruments agents at defined lifecycle hooks (user messages, agent triggers and responses, tool call requests and results, knowledge and memory retrieval,

memory writes, sub-agent start and stop, skill registration and loading, and session, turn, and compaction boundaries), returns inline allow, deny, modify, ask, or defer verdicts through a Guardian Agent pattern with a deterministic policy layer that always runs first, and standardizes agent telemetry and bills of materials on open conventions. The mapping to VAOM runs the same direction as the temporal one: ACS standardizes how a boundary is enforced and observed at runtime, and VAOM produces the boundary. Matrix rows and circuit breaker conditions compile to intervention-point policies, and a VAOM guardian (Section 10) has the ACS Guardian Agent as a natural implementation target, with VAOM supplying what the standard presupposes: the guardian's own matrix row, its accountable human, and its containment-only authority profile. Gartner's guardian-agent category (Market Guide, February 2026) frames the same function from the analyst side, and VAOM's clarification applies to both: a guardian is itself a delegated agent. One disambiguation: Microsoft's Agent Governance Toolkit separately uses "Agent Control Specification," also abbreviated ACS, for its own policy-decision-point artifact; references here are to the OWASP standard.

Identity-layer delegation drafts (2026): The credential mechanics of Section 9 are being independently formalized at the standards layer. An IETF draft on attenuating authorization tokens (offline-derivable credentials whose scopes can only narrow) and an OAuth actor profile for delegation chains give the attenuation rule and the accountable-role-plus-agent-identity audit pair native token semantics, and the OpenID AuthZEN Authorization API 1.0, a Final Specification since January 2026, standardizes the decision interface through which such scopes are evaluated. The division of labor is unchanged: the drafts define how narrowed authority is carried; the matrix defines what it is narrowed to.

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): 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, confidence threshold policy, escalation rules, no-automation zones defined, 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
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, threshold calibration against real outcomes, authority boundary refinement, scorecard baseline established, audit dry-run including an enforcement test (attempt a non-delegable action; verify structural denial)	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 (Section 5.4) 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 completes the model's arc from design to enforcement. The Delegation Authority Matrix is no longer only a governance record: it compiles into agent identities, scoped credentials, and tool permissions that make its boundaries structurally binding; it attenuates correctly through dynamically spawned sub-agents; it is verified continuously at runtime; and its health is measured on a standard scorecard. What an agent is allowed to do and what an agent is able to do become, by construction, the same thing.

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

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. Map the confidence thresholds to your organization's risk appetite. Document the delegation authority for each decision type, then compile it: bind every boundary to agent identity, scopes, and tool manifests. The 90–day roadmap in Section 14 provides a starting structure.

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.

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

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. *Gartner Says Applying Uniform Governance Across AI Agents Will Lead to Enterprise AI Agent Failure*. Press release, 26 May 2026. https://www.gartner.com/en/newsroom/press-releases/2026-05-26-gartner-says-applying-uniform-governance-across-ai-agents-will-lead-to-enterprise-ai-agent-failure [Forecast]
4. 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]
5. PocketOS (Jer Crane). *Founder's post-mortem of the 24 April 2026 production database deletion*. Public post, 25 April 2026. https://x.com/lifeof_jer/status/2048103471019434248 [Empirical result]
6. 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]
7. 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). https://www.imda.gov.sg/-/media/imda/files/about/emerging-tech-and-research/artificial-intelligence/mgf-for-agentic-ai.pdf [Framework guidance]
8. 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]
9. Cloud Security Alliance. *Agentic AI Identity and Access Management: A New Approach*. Research publication, 18 August 2025. https://cloudsecurityalliance.org/artifacts/agentic-ai-identity-and-access-management-a-new-approach [Framework guidance]
10. NIST Center for AI Standards and Innovation. *AI Agent Standards Initiative*. Launched 17 February 2026. https://www.nist.gov/artificial-intelligence/ai-agent-standards-initiative [Standard (programme)]
11. NIST. *SP 800-53 Control Overlays for Securing AI Systems (COSAiS)*. Project page; predictive AI overlay annotated outline, 8 January 2026. https://csrc.nist.gov/projects/cosais [Standard (pre-final)]
12. Amazon Web Services. *Introducing Dogwood: Runtime Verification for AI Agents*. AWS Open Source Blog, 6 August 2026; language repository at github.com/dogwood-policy/dogwood. https://

aws.amazon.com/blogs/opensource/introducing-dogwood-runtime-verification-for-ai-agents/ [Product release]

13. OWASP GenAI Security Project. *Agent Control Standard (ACS)*. Specification v0.1.1, public since May 2026; adopted into the OWASP GenAI Security Project, 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)]
14. Microsoft. *Microsoft Entra Agent ID*. Announcement, 19 May 2025. https://www.microsoft.com/en-us/security/blog/2025/05/19/microsoft-extends-zero-trust-to-secure-the-agentic-workforce/ [Product release]
15. 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]
16. 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]
17. 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]
18. 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]
19. European Parliament and Council of the European Union. *Regulation (EU) 2026/1744 amending Regulation (EU) 2024/1689 as regards the simplification of the implementation of harmonised rules on artificial intelligence (Digital Omnibus on AI)*. Adopted 29 June 2026; signed 8 July 2026; Official Journal 24 July 2026. https://eur-lex.europa.eu/eli/reg/2026/1744/oj [Regulatory text]
20. K. J. Kevin Feng, David W. McDonald, and Amy X. Zhang. *Levels of Autonomy for AI Agents*. arXiv:2506.12469, June 2025 (revised July 2025). https://arxiv.org/abs/2506.12469 [Academic paper]
21. Microsoft. *Agent Control Specification (ACS), Agent Governance Toolkit*. Project documentation, 2026 (cited for disambiguation only). https://microsoft.github.io/agent-governance-toolkit/packages/agent-control-specification/ [Standard (draft)]
22. N. A. Niyikiza. *Attenuating Authorization Tokens for Agentic Delegation Chains*. IETF Internet-Draft draft-niyikiza-oauth-attenuating-agent-tokens-01, 15 June 2026 (individual submission). https://datatracker.ietf.org/doc/draft-niyikiza-oauth-attenuating-agent-tokens/ [Standard (draft)]
23. Karl McGuinness. *OAuth Actor Profile for Delegation*. IETF Internet-Draft draft-mcguinness-oauth-actor-profile-00, 30 April 2026 (individual submission). https://datatracker.ietf.org/doc/draft-mcguinness-oauth-actor-profile/ [Standard (draft)]
24. OpenID Foundation, AuthZEN Working Group. *Authorization API 1.0*. Final Specification, 11 January 2026. https://openid.net/specs/authorization-api-1_0.html [Standard]

Colophon

This edition was drafted and revised with Claude Fable 5 (Anthropic), working under the Draft & Approve pattern this paper describes in Section 5.3: the model drafted, and the accountable author reviewed, modified, and approved every position. Framework decisions, worked examples, and errors are the author's. The same discipline applies to the Delegation Lab simulation at verkflode.eu, 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	Q1 2026	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	Q1 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: 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: What VAOM Is (and What It Is Not), including credit risk decisioning analogy. Added hidden decisions guidance to Decision Inventory (Section 5.1) 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 with guidance on navigating contested ownership. Added Implementation Challenges subsection to composite confidence scoring (Section 8) 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: 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: 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: 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: 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.

VERSION	DATE	CHANGES
4.1	August 2026	The temporal compilation update. Added the temporal compilation path to Section 9: 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) and to the alignment mappings (Section 13). Noted the safety-versus-liveness boundary of runtime enforcement (Section 10) and enforcement event streams as scorecard data sources (Section 11). No architectural changes: no new sections, principles, layers, or metrics.
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). 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 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). Stated the composition boundary between temporal policy operators and the calibration and anomaly assurance layer (Section 13) and named the standard runtime intervention points where suspensions execute (Section 10). 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). 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.