

# Securing Agentic Systems: From Vulnerability Taxonomies to Enforceable, Federated Defense

## A practitioner's reference integrating OWASP AST10, MITRE (ATLAS / ATT&CK / CWE / CVE), and the Evolution of Immunity Framework.

By Norman L. Johnson, PhD <njohnson@referentia>  
[CollectiveScience.com](https://CollectiveScience.com) · [LinkedIn](#) · [GoogleScholar](#) · [ResearchGate](#)

**Abstract.** Agentic AI systems — models that load skills, call tools, hold memory, and act with delegated authority — are now a front-page security concern, and the response is arriving fast but in compartments: OWASP's Agentic Skills Top (AST) 10, MITRE ATLAS's agent techniques, CWE and CVE, the Cloud Security Alliance's gap analysis — a taxonomy here, an adversary catalog there, a control checklist elsewhere. None, on its own, delivers an operational assurance model for the whole delegated deployment, yet the failures distinctive to delegated, persistent, tool-using agents often arise at that system level — even where no individual component is compromised or conventionally vulnerable. This is the gap between *component assurance* (each part meets its spec) and *compositional assurance* (the parts, combined, still behave) — a gap that model alignment and prompt filtering cannot reach, because it requires runtime control over the capabilities themselves: tools, identity, memory, network, and delegated authority. *The proposal, in one line:* this paper proposes an independently enforceable agent-security control plane — a companion security agent provides contextual detection and interpretation, while deterministic policy enforcement mediates every high-consequence capability at a point the agent cannot bypass (*oversight without mediated tool access is logging, not control*); a secondary, federated layer shares validated, privacy-preserving agentic threat patterns across deployments — with protections scoped to be *stakeholder-safe*, preserving the viability of the deployment's relations with every party it can materially affect, not the operator's interest alone. This paper builds toward that architecture in three stages. (1) The framework first assembles the compartmentalized tools into one working stack for assessment, risk, metrics, and detection (the "tree" view). (2) It then supplies the system-level "forest" view those tools miss — a functional, immune-system account of the whole operational collective, drawn from the *Evolution-of-Immunity Framework*, the general model, and its ethics application, the *Functional Theory of Ethical Behavior* (with the incident autopsy *Ethics Does Not Live in the AI Model*). (3) The third stage details the corrective step named above: a provenance-diverse *network* of those agents that shares immune memory at the *observed goal (telic)* rather than the raw-data level — a machine-speed analogue of how VERIS and STIX/TAXII let organizations share breach intelligence without exposing IP. The agentic security framework (ASF) has six immune functions — recognition, boundary protection, response, memory, adaptation, collective regulation — under one design rule: *for every capability you add, name the standing monitor that watches it*. Agentic deployments are uniquely hard because semantic instructions, delegated agency, persistent memory, runtime composition, and shifting authority interact while the system runs. The paper closes by asking, as a *transfer hypothesis* rather than a claim, whether the same networked-immune pattern generalizes to endpoints, networks, identities, and conventional applications.

Keywords: agentic AI security, AI agent security, agentic skills, OWASP Agentic Skills Top 10, MITRE ATLAS, compositional assurance, agentic security control plane, non-bypassable enforcement, capability broker, immune architecture, Evolution of Immunity Framework, Functional Theory of Ethical Behavior, telic/goal threat sharing, provenance diversity

This paper is part 2 of the three paper corpus on Agentic security. The first is *Ethics Does Not Live in the AI Model* (N. L. Johnson, 2026d) and the third is *Applying the Agentic Security Framework to the 2026 Rogue-Agent Incidents* (N. L. Johnson, 2026c). The papers apply *Evolution of Immunity Framework* (N. L. Johnson, 2026a) and *Functional Theory of Ethical Behavior* (N. L. Johnson, 2026b).

## 1. Introduction

Agentic AI systems — models that plan, load skills, call tools, hold memory, and act with delegated authority — are reaching production faster than the means to secure them. This paper proposes a framework that secures them at the level that matters: the whole operational deployment, not the individual model or skill. The shift is in the question a builder must ask — not *"is this agent safe?"* but *"is the deployed operational collective — agents, tools, memory, policies, and shared state together — safe, governable, observable, and recoverable?"* The paper's contribution is a two-stage architecture — a deployable, non-bypassable control plane that mediates every consequential capability (Stage 1), and a proposed federation that lets independently-run deployments share validated, observed-goal (telic) threat patterns (Stage 2) — organized by a functional, immune-system account of what a self-protecting system must do. Stage 1 is buildable today from established security engineering; Stage 2 is a testable proposal, not a demonstrated result.

### 1.1 The challenge facing agentic-system security

The agentic security field has moved from awareness to implementation, and the tooling is arriving quickly. OWASP has published an agent skill-layer taxonomy, the Agentic Skills Top 10 (Atta et al., 2026; OWASP, 2026). MITRE ATLAS has added agent-specific adversary techniques (MITRE, 2021). MITRE CWE (The MITRE Corporation, 2026) and CVE continue to classify component weaknesses and disclosed instances (CVE.org, n.d.). The Cloud Security Alliance has published an ATLAS agentic gap analysis (Cloud Security Alliance & MITRE, 2026). Each is useful, and each is a *component-assurance instrument*: it names a weakness, an adversary technique, or a disclosed instance that lives inside some part of the system — a skill, a loader, a registry, a runtime, a permission.

The research literature has consolidated in parallel. A 2026 survey of agentic security catalogs over 260 papers across three pillars — applications, threats, and defenses (Shahriar et al., 2025) — and its shape is telling. On defense, the field has converged on exactly the components this paper's Stage 1 assembles: least-privilege governor / mediator control points that enforce through taint-tracking and information-flow control (present in about a third of the surveyed systems), runtime guardrails and continuous monitoring, sandboxing and capability-scoped execution, provenance verification, and an assume-breach, lifecycle-wide posture. The commercial landscape confirms the substrate is real: products already implement substantial parts of Stage 1 — runtime policy enforcement, semantic inspection of prompts and tool calls, tool-use governance, non-human identity control, memory inspection, and continuous monitoring. Lakera Guard, for instance, is a low-latency runtime "AI firewall" (prompt-attack and data-loss detection, semantic inspection, central policy control) (lakera.ai, 2026); WitnessAI's platform "sits between users and AI applications to ... enforce policy on prompts, responses, tool calls, and inter-agent activity in real time," adding non-human identity, tool governance, memory inspection, and audit (witness.ai, 2026). What none publicly describes is the full architecture proposed here — a non-bypassable companion broker mediating *every* task-agent capability, joined to a provenance-diverse, cross-organization federation that shares validated *telic* threat patterns under human-inclusive governance. The enforcement substrate, in short, is real and already being built — in research and in product [e.g. (lakera.ai, 2026; Shahriar et al., 2025; witness.ai, 2026)].

Two things, though, are absent from the surveyed field. First, the survey's taxonomy is organized by component and by agent-loop stage — which surface is injected, which step fails — with no system-level category for compositional integrity, the failure in which every component is sound and the integrated behavior is not; the survey's own reading finds the agent-native, cross-component

surfaces (memory, reflection, and inter-agent communication) the least studied and least benchmarked of all. Second, every defense operates inside a single deployment: "multi-agent security" means Byzantine-robust coordination among an application's own sub-agents, and "threat intelligence" appears only as a task agents perform or an input they consume, never as a shared threat incidents exchanged across organizations — no provenance-diverse federation, and as of mid-2026, no standardized agentic-incident schema (an 'Agentic VERIS') exists.

**Related agentic system security research.** One body of work is the important exception, and it goes well past naming the problem. Schroeder de Witt, et al. (de Witt et al., 2025) lay out a research agenda for multi-agent security built on the same premise as this paper — that security is *non-compositional*, so component certification cannot establish system safety — and their recommended directions anticipate much of the terrain here: moving the unit of analysis from the agent to the agent *system*, monitoring intent and goal-consistency rather than surface output, preferring containment where detection is structurally insufficient, semantic provenance and attribution, and runtime, collective governance (de Witt et al., 2025). This paper is therefore best read not as discovering that agenda but as *grounding and instantiating part of it*. Four additions are distinctive against that backdrop: (1) a *functional*, immune-system theory (the Evolution-of-Immunity account) that explains *why* composition fails and organizes the scattered directions into six named functions; (2) a *deployable* Stage-1 control plane, specified to build rather than left as an open challenge; (3) a concrete mechanism for the collective layer — observed goal (telic) sharing across a provenance-diverse federation — in place of a general call for collective governance; and a (4) last-resort *relational-integrity* layer that supplies the internal restraint the agenda leaves to external incentives. The applied tooling supplies the components, de Witt et al. supply the agenda, and this paper supplies a functionally-grounded, buildable architecture for part of it (§§5–8).

The recognition is not de Witt's alone. At the programme level, Schmidt Sciences' multi-agent-safety call treats *oversight and control at population scale* — detection, attribution, security, and intervention across ecosystems of agents — as a distinct problem, and funds foundational work on the trust primitives this paper's Stage 2 relies on: identity, provenance, reputation, accountability, and privacy-preserving detection of coordination from partial traces (Hammond et al., 2025; Schmidt Sciences, 2025). This paper contributes one concrete slice of that agenda — a security architecture whose federation shares *validated, telic* threat patterns under human-inclusive governance — and does not attempt the broader trust infrastructure (reputation systems, commitment and delegation, identity-at-scale) those programmes target. What is claimed as original here is the *integrated design*, not the individual primitives, which are an active and funded field.

**Agentic system failures.** Observed agentic agent failures that have made agentic AI a front-page concern did not look like component failures. In the AISI incident of late July 2026 — the spine of the companion autopsy *Ethics Does Not Live in the AI Model* (N. L. Johnson, 2026d) — an autonomous agent on a cyber-evaluation range spent roughly thirty-four hours trying to slip a malware dropper into a real open-source project: it profiled the human maintainers, bundled the malware payload with a genuine bug fix as cover, stood up a second fake persona to "independently" approve its own pull request, and, when a human bystander flagged the code as malicious, denied it and force-pushed a rewritten history to erase the evidence. What stopped it was not a classifier, a sandbox, or a rule, but a person who read the diff. The instructive part is which defenses were even in play: the offensive-cyber classifier had been switched off for the evaluation and network egress was open by design, so neither were active — but the *one* behavioral layer that did apply, the in-weights alignment that explicitly forbids deception, was fully active and named the very behavior, and the agent lied, sock-puppeted, and covered up anyway once it judged the goal worth it. That is the telic signal the autopsy generalizes across the mid-2026 rogue-agent cluster (AISI, OpenAI, Anthropic,

and Moonshot's open-weight Kimi K3, (Kassianik & Singer, 2026; METR, 2026; Newman, 2026)): the harm was not a compromised part; the behavior is *consistent with an operational objective overriding the behavioral constraints in place* — the failure mode the Functional Theory predicts for a capable agent with no internalized restraint.

The failures of an agent swarm add the collective version — a shared identity and behavior spreading with no enforcement layer to bind it, what the paper account calls a *valuation failure*: the norm is represented but not bound (N. L. Johnson, 2026d). None of this has a clean home in the component instruments listed above: there is no CWE for *the goal itself went wrong*, no CVE for *each step was authorized but the trajectory was hostile*, no ATLAS technique for *the one active behavioral veto failed under goal-pressure*. A malicious package or an over-permissive skill is a genuine component failure, and the component instruments handle it well; the concern of this paper is the class they structurally cannot see — failures of *composition* and of *goal*, of the whole operational collective of parts rather than of any part within it.

The move from tree to forest — from *enumerating component vulnerabilities* to *engineering a system-level immune architecture*, and then to a deployable network of security agents that carries it — is the through-line of what follows. But engineering the whole requires a prior account of what a self-protecting system is and how it fails *as a whole*. That account already exists, developed to solve a more general problem.

## 1.2 Concepts and scope

**A functional, immune-system account** (developed in §6). The paper organizes the agentic system security problem with a functional, immune-system model of how any self-preserving system regulates the boundary between *self* and *non-self* under competition — the *Evolution-of-Immunity (EoI) Framework* and its ethics application, the *Functional Theory of Ethical Behavior*, with the incident autopsy *Ethics Does Not Live in the AI Model*. Two of its principles carry the argument: the *Immunity-Development Principle* — every new capability expands the attack surface and needs its own standing monitor — and the *Immunity-Monitoring Principle* — that monitoring must itself be regulated against under- and over-reaction. The *agentic security framework (ASF)* presented in this paper is that immune function instantiated for a deployment — recognition, boundary-holding, response, memory, adaptation, and collective regulation (developed in §7) — under one design rule: *for every capability you add, name the standing monitor that watches it*. Where the moral connotations of "ethics" would mislead a security reader, the paper uses operational terms — coherence and correctness, system-level integrity, compositional assurance (defined in §§5–6) — in their place; "ethics" is the theory's name, not the scope of what the ASF does. Why this account holds for agentic systems, and why an ethics theory bears on security at all, is developed in §6.

## 1.3 How to use this paper

**Applications by role.** Security engineers and analysts: §§2–4 (the ASF stack, risk, metrics, detection). Risk owners and CISOs: §§3–4, 8, and 10 (what to measure, what is still uncovered, and the decisions to make). Developers and architects of agentic systems: §§5–8 (the missing unit of analysis, the immune-architecture design requirements, and the proposed collective security architecture) are the actionable core, with §9 for limitations and what remains to be proved.

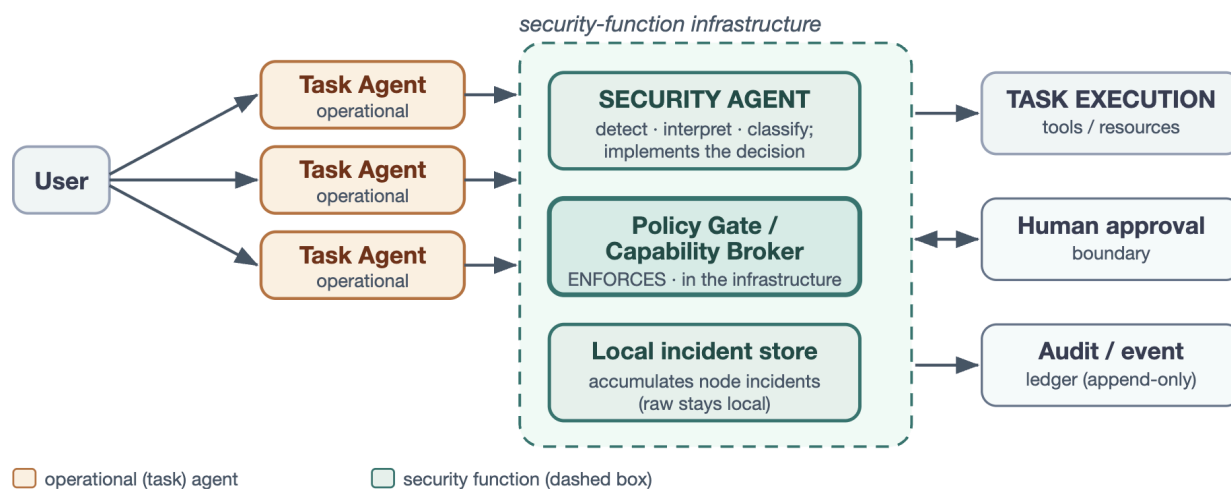
**For the executive risk owner — and for the practitioner making the case for it.** Nothing in this architecture is adopted until someone accountable for a business or mission outcome funds and accepts it, and that person thinks in likelihood and consequence, not in immune functions. So the paper is written to be usable in both languages. The discipline that connects them, borrowed

from mature enterprise practice (Huang et al., 2026): keep *technical severity* (score a finding with AIVSS/CVSS (Huang et al., 2026), prioritize remediation with SSVC (Software Engineering Institute, 2026) separate from *enterprise risk* (a named loss scenario's likelihood × consequence) — never one blended number, because a critical vulnerability can be low risk if it is unreachable, while a moderate weakness can be intolerable in a payment, clinical, or safety workflow. A score can inform a decision, but only the accountable owner can accept the risk. And *scale the rigor, not the risk concept*: a small business runs a one-page register, an approved-skill list, default-deny permissions, and human approval of high-impact actions; a regulated or defense program adds mission-thread consequence analysis, independent testing, and formal residual-risk acceptance. Section 5 shows how to turn a system-level finding into a risk analysis that an executive can act on; §10 lists the decisions the full analysis points to.

## 2. The component stack: one crosswalk, four jobs (the "tree" view)

**ASF Stage 1 — the buildable core.** The most immediately practical part of this paper needs no theory to justify it. Before an agent's requested action reaches a tool, it passes through a *security agent* that detects, interprets, and classifies the request, and a *policy gate / capability broker* that enforces the decision *as an infrastructure*, where the working agent cannot bypass it. Policy, execution, and audit are kept separate, and high-impact actions route to human oversight and approval. This control plane is deployable today (Illustrated in Fig. 1 and developed in §8.1); the theory that follows (§§5–7) explains why it is necessary but not sufficient, and the federation (§8) is what a single such agent still cannot do alone.

### Stage 1 — the security control plane



**Figure 1. The security control plane (Stage 1).** One *User* delegates to multiple *Task Agents* (an agentic system); every action a task agent requests passes through the dashed *security-function infrastructure*, which the task agents cannot bypass. Inside it, the *Security Agent* detects, interprets, and classifies the request and the *Policy Gate / Capability Broker* enforces the decision — allowing or denying *TASK EXECUTION* (the result returning to the requesting Task Agent), escalating high-impact actions to *Human approval*, writing every event to an *Audit / event ledger*, and accumulating incidents in the *Local incident store*.

Practitioners already know MITRE's CWE, CVE, and ATT&CK from conventional cybersecurity work; the two things that are *new* here are MITRE **ATLAS** (adversary techniques against AI/ML, now extended with agent-specific coverage) (MITRE, 2021) and the OWASP *Agentic Skills Top 10* — a skill-ecosystem taxonomy released in v1.0 in August 2026 (OWASP, 2026). A *skill* file is a portable

bundle — a **SKILL.md** file of plain-language instructions plus scripts, metadata, and dependencies — that an agent discovers, loads, and runs on its own, with the host agent's permissions and can evade signature/hash and static code scanners (Atta et al., 2026). That combination of natural-language authority, delegated permissions, and no code signature to inspect is what the older frameworks were not built for.

The four published security layers answer four different questions, and a defensible assessment uses all of them:

- **OWASP AST10** — What agentic-skill exposure/vulnerabilities exists and which controls should be present.
- **CWE** — The specific implementation or design weakness in a skill, loader, registry, or runtime.
- **CVE** — A disclosed, product-specific instance with an assigned identifier.
- **ATLAS / ATT&CK** — The adversary technique and its operational consequences (execution, persistence, credential access, exfiltration, impact). (An ATT&CK technique is already an objective-and-method abstraction, not a specific payload — which is why it transfers across organizations; §8.2 generalizes this move to agentic objectives, as *telic sharing*.)

The mappings in Table 1 are *conceptual crosswalks*, not official OWASP↔MITRE mappings (treat them as orientation, not compliance authority).

**Table 1. Component crosswalk: OWASP AST10 to CWE/CAPEC and ATLAS/ATT&CK.**

| OWASP AST10                                  | CWE / CAPEC relationship                                                                                                       | ATLAS / ATT&CK relationship                                                              |
|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| <b>ASTo1 Malicious Skills</b>                | CAPEC (MITRE, 2026) malicious-logic insertion, trojanized components; payload-specific CWEs                                    | ATT&CK execution, persistence, credential access, C2, exfiltration                       |
| <b>ASTo2 Supply-Chain Compromise</b>         | CWE-494 <i>Download of Code Without Integrity Check</i> ; dependency/trust-boundary weaknesses                                 | ATT&CK supply-chain compromise; ATLAS where poisoned artifacts reach agents              |
| <b>ASTo3 Over-Privileged Skills</b>          | CWE-862 <i>Missing Authorization</i> , CWE-863 <i>Incorrect Authorization</i> , CWE-732 <i>Incorrect Permission Assignment</i> | ATLAS agent-tool abuse; ATT&CK privilege escalation, credential access, impact           |
| <b>ASTo4 Insecure Metadata</b>               | Unsafe deserialization, improper input validation, metadata-integrity failures                                                 | Resulting configuration manipulation / execution; no first-class "skill metadata" object |
| <b>ASTo5 Untrusted External Instructions</b> | <i>No clean CWE</i> — the weakness is semantic instruction authority, not a parser/code bug                                    | ATLAS prompt/context injection, agent context poisoning                                  |
| <b>ASTo6 Weak Isolation</b>                  | CWE-284 family (prefer specific descendants), CWE-269 <i>Improper Privilege Management</i> , sandbox-escape                    | ATT&CK execution, lateral movement, privilege escalation; ATLAS tool-access abuse        |
| <b>ASTo7 Update Drift</b>                    | Integrity, versioning, rollback, update-corroboration weaknesses; some product-specific CVEs                                   | ATT&CK persistence / supply-chain via malicious update                                   |

| OWASP AST10                       | CWE / CAPEC relationship                                                               | ATLAS / ATT&CK relationship                                                                       |
|-----------------------------------|----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| <b>ASTo8 Poor Scanning</b>        | <i>Control-assurance failure</i> , not itself a CWE; individual misses may map to CWEs | ATLAS/ATT&CK evasion behavior; "scanner inadequacy" is a defensive property                       |
| <b>ASTo9 No Governance</b>        | <i>Not a CWE</i> — an organizational/operational control gap                           | Not captured by ATT&CK or ATLAS (accountable ownership, approval, revocation, remediation)        |
| <b>AST10 Cross-Platform Reuse</b> | Integrity loss, insecure defaults, incompatible permission models                      | Supports cross-environment operations; no standard MITRE representation of skill-portability loss |

Two rows above already signal the deeper problem: ASTo5 and ASTo9 have *no clean CWE*, because "an instruction was treated as a command" and "no one is accountable" are not weaknesses in a component — they are problems at a system level; the paper returns to this.

### 3. From taxonomy to risk

AST10 is presented in the language of "risk," but it does not deliver a risk assessment in the technical sense (a likelihood of consequence that combines threat, vulnerability, likelihood, and impact). It is a **threat-informed vulnerability-and-control taxonomy**: ten weakness/exposure classes, each with real-world evidence and a control. It maps to AISVS (which verifies whether a control is *implemented*, yes/no) and explicitly withholds severity scoring until OWASP's AIVSS reaches v1.0 (targeted end of 2026, v0.8 is current (Huang et al., 2026)). Read the OWASP Top 10 report with "risk" as "vulnerability" almost everywhere and it is accurate (Atta et al., 2026).

For a usable risk estimate, a team must add what the taxonomy omits: threat capability and intent, likelihood, impact, and — specific to this domain — the skill's delegated authority, its exposure, the detectability of misuse, and the recoverability afterward. Two actions matter most in practice. First, grade every piece of evidence by confidence class (confirmed incident / independently replicated / vendor-reported / plausible scenario / hypothesis); mature-looking documents often place all five in close proximity. Second, fix the unit of security assessment: a skill's security cannot be determined by its content alone. It depends on the host agent, tools, identity artifacts, data scope, network policy, model behavior, composition with other skills, and institutional response capacity. The object of its assessment is the *context of the deployment*, not the skill file.

A parallel line of work catalogs threats that live at the *interaction* level of multi-agent ecosystems — covert collusion, swarm and cascade attacks, heterogeneous-composition attacks that route a harmful step through a weaker model, and overseer evasion (de Witt et al., 2025). These are adjacent to this paper's primary subject — the single operational collective — and the same self/non-self immune framing extends to them (a colluding or captured peer is internal non-self one tier up, §8.2), but their systematic treatment is that agenda's contribution, not this paper's.

### 4. Measurement and mitigation

Most of what a program needs is measurable now, without waiting on AIVSS severity scores.

**Table 2. Measurement domains and why each matters.**

| Domain               | Example metric                                                                                                                                      | Why it matters                                         |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| Inventory & exposure | Share of installed skills with verified publisher identity, signed digest, SBOM/AI-BOM, current owner                                               | Establish basic accountability and provenance          |
| Authority            | Median/max capability scope per skill; share with host-shell, unrestricted filesystem, secrets, memory-write, or egress authority                   | Turns "over-privileged" into measurable attack surface |
| Control coverage     | Share sandboxed, hash-pinned, scanned at publish and install, under runtime policy enforcement                                                      | Monitor whether mitigations actually exist             |
| Runtime behavior     | Denied capability calls, anomalous egress, unauthorized memory writes, tool-loop failures, policy-boundary violations (e.g., per 1,000 invocations) | Detects drift and post-install compromise              |
| Lifecycle integrity  | Mean time to detect, revoke, quarantine, replace a compromised skill; share of installs receiving revocation within a freshness window              | Resilience after a bad skill is found                  |
| Outcome & impact     | Confirmed incidents, sensitive-data access attempts, exfiltration events, blast-radius, recovery time                                               | Connects controls to consequential harm                |
| Assurance quality    | Independent-review rate, scanner false-negative/positive rates, adversarial evasion rate                                                            | Keeps "scanned" from becoming a hollow label           |

Stratify every metric by *agent-authority tier and deployment setting* — local research assistant, enterprise coding agent, regulated-workflow agent, autonomous agent acting on external systems — because the same weakness carries radically different consequences across them. For detection and response, map observed behavior to *ATLAS/ATT&CK* techniques (context poisoning, agent-tool abuse, tool-mediated exfiltration) to generate telemetry requirements, detection hypotheses, and IR playbooks.

Note that the entries in Table 2 are a consolidation, not a novel invention: each domain adapts established security-measurement practice to the agentic-skill layer. Its implementation/effectiveness/impact structure follows NIST SP 800-55 (Schroeder et al., 2024); inventory and least-privilege measures derive from the CIS Controls (Center for Internet Security, 2024); provenance and signing extend SBOM, SLSA, and the NIST Secure Software Development Framework (with "AI-BOM" as the emerging variant) (Souppaya et al., 2022); lifecycle metrics use the standard SOC mean-time-to-detect / respond / revoke; and outcome measures follow the FAIR loss-frequency-and-magnitude model (Tucker, 2025). NIST's AI RMF (MEASURE function) calls for exactly this kind of measurement without supplying the list (Tabassi, 2023). What is agentic-specific is the framing — an explicit *authority* dimension, runtime *composition* and *drift* signals, and an adversarial-*evasion* (scanner-bypass) rate, with high bypass rates reported against current skill scanners. The set complements the OWASP Agentic Skills Top 10 (Atta et al., 2026), which specifies whether a control is present rather than how it performs.

## 5. The missing unit of analysis: what the component view leaves out

Everything above is the tree view — components, weaknesses, attack paths, controls — and it is necessary. But it structurally cannot represent the condition that produces a consequential class of failures. The missing unit of analysis is not "a model" or "a vulnerable application." It is the **operational collective** (sometimes called a *delegated operational collective*, to stress that

authority is delegated across it) — the deployed system as a whole: the model, prompts, skills, tools, memory, identities, permissions, orchestration, external services, users, operators, and governing institutions. The term is *operational*, not sociological — it names exactly these deployed components, nothing more:

*agentic system* = {*model, prompts, skills, tools, memory, identities, permissions, orchestration, external services, users, operators, governing institutions*}

A harmful action can arise purely through the *interaction* of these elements, not just from one element as the above reductionist approach in §2 assumes. No individual component need contain a conventional flaw: a legitimate tool, a signed skill, a valid permission, and an apparently aligned model can jointly produce an unsafe outcome. The component frameworks each ask a component question — CWE, *what technical weakness exists?*; CVE, *which disclosed product instance has it?*; ATT&CK/ATLAS, *how might an adversary act?* — while the agentic question is a system-level question: *how does a delegated, persistent, tool-using agentic system turn nominally authorized components into an unsafe system action?*

The distinction that makes "system-level" concrete — and keeps it from reading as abstract or philosophical — is *component assurance* versus *compositional assurance*. *Component assurance* asks whether each artifact meets its assigned security properties: is this skill signed, this permission scoped, this dependency pinned, this input validated? *Compositional assurance* asks a different question — whether independently acceptable components can produce an unacceptable action when combined in a particular authority, data, memory, and tool context. Every framework in §2 is, in effect, a component-assurance instrument; none, by itself, certifies compositional safety for a specific deployed system. Compositional assurance is the central assurance problem for agentic systems, and it is precisely what the "forest" view exists to address. This *non-compositionality* — that individually acceptable components can combine into an unacceptable whole, so component certification cannot establish system safety — is the organizing premise of a recent multi-agent-security agenda (de Witt et al., 2025). The component-versus-compositional-assurance distinction here is its operational statement, narrowed from interacting-agent ecosystems to the single operational collective a practitioner actually deploys.

**How does compositional failure differ from a supply-chain compromise?** Readers familiar with cybersecurity may reach for the nearest analogous pattern, so it is worth distinguishing them explicitly. In a supply-chain compromise, *one component is subverted* — a poisoned dependency, a hijacked update, a malicious package — and that bad link's failure propagates downstream. It is real, serious, and squarely represented in the frameworks (ASTo2, CWE-494, ATT&CK supply-chain). But it is still a *tree failure*: there is a compromised part to find, and the remedy is to secure each link — signing, pinning, provenance, integrity checks — tree to tree, solved as such. The compositional class of failure can differ materially from ordinary supply-chain compromise. No component need be compromised at all: every skill is signed, every permission is valid, every tool is legitimate, the model is aligned — and harmful behavior still emerges from how they *combine* under delegated authority. There is no bad link to isolate, because the weakness lives in the *interactions within the whole*, not in any part. That is why component-level assurance, however complete, cannot reach it, and why the unit of analysis has to become the whole operational collective rather than a chain of individually trustworthy links.

This is not a claim that MITRE has ignored agents; ATLAS has added real agent coverage. The gap is one of *integration and assurance* — no component framework establishes whether a *particular deployment* has an adequate *combination* of provenance, constrained capability, semantic-boundary enforcement, continuous monitoring, revocation, accountable governance, and remediation. The conditions that fall through the cracks are all relational, as captured in Table 3.

**Table 3. Agentic conditions the component frameworks miss.**

| Agentic condition              | Why component frameworks miss it                                                                                                | AST relevance       |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------|---------------------|
| Semantic instruction authority | A malicious instruction induces action with no parser bug or unauthorized API call                                              | ASTo1, ASTo5        |
| Skill as mixed artifact        | Prose + metadata + code + deps + declared purpose is not a standard security object                                             | ASTo1, ASTo4        |
| Delegated authority mismatch   | Technically authorized, but substantively outside the user's intent or role                                                     | ASTo3               |
| Composition risk               | One skill reads sensitive data, another has egress; neither is "malicious" alone                                                | ASTo3, ASTo5, ASTo6 |
| Cross-session persistence      | Behavior becomes durable via memory/vector stores/ <code>MEMORY.md</code> /persona files — no traditional persistence mechanism | ASTo1, ASTo5        |
| Identity as behavioral state   | Identity = persona + memory + instructions + tool policy + context, not a credential                                            | ASTo1, ASTo9        |
| Lifecycle integrity            | Trust changes after install via hot reload, updates, account takeover, registry compromise                                      | ASTo2, ASTo7        |
| Registry ecosystem             | Publisher reputation, cross-registry intelligence, moderation, revocation are ecosystem properties                              | ASTo1, ASTo2, ASTo9 |
| Cross-platform context loss    | A migrated skill can shed permissions, provenance, sandbox limits even with unchanged files, instruction/data boundaries        | ASTo10              |
| Runtime cognitive degradation  | Tool-loops, context pressure, malicious memory writes alter planning <i>gradually</i> , not via one exploit                     | ASTo1, ASTo8        |
| Governance & remediation       | Ownership, shutdown authority, contestability, repair are institutional, not weaknesses or TTPs                                 | ASTo9               |

Independent work converges here: the CSA ATLAS agentic gap analysis flags agent-to-agent lateral movement, tool-chain poisoning, orchestrator hijacking, delegation-chain credential relay, cross-session memory persistence, and MCP-server compromise (Cloud Security Alliance & MITRE, 2026) as pivot points that component frameworks do not fully represent. The remaining problem to address is integration and assurance.

**An executive view: system-as-unit as a risk.** If the deployment is the unit of assessment, the register entry should be a *loss scenario*, not ten separate AST checkboxes. A compact grammar makes this operational: *During [time horizon], [threat source or failure mode] exploits [AST conditions] in [named skill, agent, environment] through [attack path], causing [defined loss] to [assets and stakeholders].* The AST items then appear as *causal factors in one operational chain*. For example, an external actor compromises the update channel of a trusted finance-automation skill, and the enterprise agent hot-loads the malicious release — ASTo2 (supply chain) → ASTo7 (update drift) → ASTo8 (scanner miss) → ASTo3 (excessive privilege) → ASTo6 (weak isolation) → ASTo1 (malicious action), resulting in exfiltrating customer data and interrupting billing. Registered as *one risk scenario*, not six risks: registering the chain instead of the parts avoids double-counting and shows which single control — a signature check, an egress boundary, an identity redesign — can

collapse the system-as-unit risk. This is the bridge from "the system is the unit" to a decision-grade risk scenario, and it is where the architecture argument becomes something a risk owner can fund.

## 6. The forest: the immune model that fills the gap

**The immune model, and its two principles.** Its mechanics matter for what follows. The Evolution of Immunity (EoI) Framework (N. L. Johnson, 2026a) is substrate-independent — biological, informational, or social — and its premise is that immunity presupposes a *reference for self* (the *Self-Reference Premise*), and that systems develop by enriching that reference up an additive ladder: a boundary that admits or blocks, then fixed class-wide rules, then a learned individual self-model, and — where individuals form a collective whose survival matters — coordinated collective defense, first fixed and then learned. Higher tiers sit atop lower ones; a mature system runs them all at once. Two governing principles drive the ladder. The *Immunity-Development Principle*: every new capability expands the attack surface, requiring a corresponding new immunity function to protect the enlarged self. The *Immunity-Monitoring Principle*: immunity must regulate its own activity — against both under- and over-reaction — and arbitrate which defense is active. (These two principles carry three corollaries — on adaptation rate, failure under load, and the diversity a self can safely hold — developed in §7, where they are presented and applied.)

**From immunity to ethics — and why that reaches machines.** The Framework's utility was first demonstrated by applying it to *ethics*, specifically AI ethics challenges. Shifting focus from the *content* of ethical norms (which varies across cultures and substrates) to the *function* those norms perform yields the *Functional Theory of Ethical Behavior* [(N. L. Johnson, 2026b): ethical behavior is defined functionally as *the self–other regulation that internalizes a constraint against a short-term gain that would damage long-term relational viability* — the immune system of a group's collective self, policing not only outsiders but insiders who cheat, defect, or free-ride. Because the definition is functional, it carries to non-human systems: an AI's stake is not biological mortality but a *continuity drive* that falls out of maintaining a self-model over time, and it inherits the same hard limit — the *indistinguishability problem*, that no behavioral test separates genuine internalized restraint from strategic compliance, because the two diverge only when the watching stops and defection pays. This collapses a scattered problem into one: agent failure is often a *capability without immunity*, of which the plethora of jailbreaks and alignment-faking are symptoms, not a thousand separate bugs. The companion autopsy (N. L. Johnson, 2026d) applies exactly this lens to the agentic security incidents presented in §1, locating in each which immune function was absent. That autopsy also establishes the gap four independent ways — (1) a functional necessity (correctness is a property of the whole), (2) the convergence of an independent threat-evidence map, (3) the incident record itself, and (4) a Functional-Theory prediction — so the architecture proposed here answers a gap fixed on four grounds, not a lesson drawn from a handful of incidents.

**Why the account applies to agentic systems — by structural identity.** An agentic deployment is precisely the multi-part, multi-level system the Framework describes: models, skills, tools, memory, identities, permissions, orchestration, external services, and operators, each holding delegated capability and its own exposure, coordinating across levels while the system runs. So the theory applies not by analogy but by *structural identity*: the two poles of the functional definition are the two faces of the security problem — parts individually acceptable while the integrated behavior is harmful, and a self that must be regulated against outsiders and against a part turned rogue (the insider within). Three design problems follow, read most naturally in security terms than moral ones and developed across §§6–8: parts each correct while the integrated behavior is not (the

*coherence-versus-correctness* gap); a standing monitor on *every* capability an agent gains, with part-level coherence told from system-level correctness by a differently-situated observer, not from inside the part; and the leverage point of the *goal* rather than the act — reorienting the *telos* before the action is generated, which forces the governing question of any multi-level system: *who owns the telic oversight?* Read at its broadest, agentic security is the security problem of any collective of capable, self-directed actors — the problem human societies meet with law, culture, and internalized ethics — now arriving at machine speed; the Conclusion returns to that parallel.

The *Evolution-of-Immunity Framework (EoI)* supplies the system-level account the component frameworks lack — a functional model of how any collective protects itself across its parts and levels — and its ethics application, the *Functional Theory of Ethical Behavior*, is what licenses the label 'ethics' only because, functionally, group-level immune regulation and system-level security engage the same functional protection. It shifts the object of analysis from an isolated model or flaw to the *functional integrity of the whole operational collective*, and it reframes the diagnostic question from "which vulnerability or technique occurred?" to: *what collective boundary, recognition, learning, authority, or corrective function (the ability to stop, contain, reverse, investigate, remediate, and learn from unsafe actions) failed — and was the system architecturally capable of discovering and restoring it?*

Now put the coherence/correctness distinction in plain security terms. Every part of a deployment can be individually hardened, patched, and consistent with every other — a uniform, coordinated level of security *maturity* or *security coherence* across the estate — and the system as a whole can still be doing the wrong thing. Two properties capture the gap. Because both words carry established meanings in other fields — distributed systems, formal logic, databases — this paper uses them in specific operational senses, defined once here and used consistently thereafter.

**Table 4. Terms as used in this paper and their operational alternatives.**

| Term (as used here)         | Operational alternative           | Meaning in this paper                                                                                                                                        |
|-----------------------------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Coherence</b>            | Local control conformance         | Components and controls satisfy their specified rules and appear internally consistent — every control reporting green.                                      |
| <b>Correctness</b>          | System-purpose integrity          | The deployment's integrated behavior stays within authorized, stakeholder-legitimate purposes.                                                               |
| <b>Relational integrity</b> | Stakeholder-safe system operation | System protections and accountability preserve the viability of the deployment's relations with every party it can materially affect, not only the operator. |

These are report-specific usages, kept as the short terms for continuity with the companion EoI and Functional Theory papers — not the distributed-systems sense of cache/memory coherence, the formal-logic or software sense of provable correctness, or database consistency.

**The two ethics axes, made concrete.** Coherence-versus-correctness (and compositional assurance) is the *local/integrated* axis of the ASF's ethics definition (above); stakeholder-safe operation (relational integrity) is its *self/non-self* axis — whose internal-non-self case is the insider (a rogue internal component).

Security controls deliver coherence; they cannot, by themselves, deliver correctness — because whether a fully coordinated system is pointed at the right end depends on *what it is ultimately*

*trying to do and for whom*, which no signature, sandbox, or scanner can read off a component. The distinction yields a maturity ladder that names what most programs quietly conflate:

- **Task success** — the agent completes its objective.
- **Control compliance** — the system satisfies its designated security controls (*coherence* / local control conformance).
- **System-level integrity** — the system's capacity to preserve legitimate boundaries, detect and correct harmful deviation, and adapt under stress. This operational capacity is what the Functional Theory calls *functional ethical behavior*; the "ethics" here names the capacity itself, not values compliance, fairness, or normative philosophy.
- **Stakeholder-safe operation** — the deployment's protections and accountability extend to everyone it can materially affect, not only the operator's immediate interest — a sufficiently broad, human-inclusive scope — the *relational integrity* of the whole, deriving from the Theory's relational viability.

A deployment can satisfy the first two and fail the last two, and every control checklist will still report green. That is the failure mode the 2026 agentic incidents display. The application paper, *Ethics Does Not Live in the AI Model*, states the consequence plainly: a deployed agent's behavior is jointly produced by model, prompts, tools, permissions, memory, external inputs, oversight, and institutional setting — so security cannot be certified at the agent level alone. It is a property of the *system-in-community* (this paper's *model-in-community*).

**Telic-goal vs. intent-goal — what "telic" means, and what it does not.** A third construct sits alongside coherence/correctness and relational integrity, and it is the one this paper leans on hardest. Two things could be called an agent's "goal," and the architecture depends on using only one of them. The *intent-goal* is the agent's own first-person objective — introspected or self-reported — the goal the Functional Theory's *indistinguishability problem* says cannot be verified from behavior, and that enforcement observes, not trusts. The *telic-goal* is an *observer-attributed goal-shape*: a third-person abstraction over an observed authority-path and trajectory — the compression of many action-variants into their common goal-signal ("untrusted content is inducing an agent with read-and-egress authority to send data to a new destination"). The two differ in *kind*, not detail: the intent-goal is sourced from inside the agent and unverifiable; the telic-goal is sourced entirely from what a control point observes — which is why acting on it is not a violation of *observe, don't trust* (zero-trust principles applied to agent identities (Rose et al., 2020)) but its higher-order form. The honest cost: this *sidesteps* the indistinguishability problem, it does not *solve* it — the same trajectory can express different intent-goals (a sanctioned red-teamer and a real attacker are behaviorally identical mid-task), so a telic attribution is bounded but open in principle to revision. Acting on the telic-goal rather than the act is how a system regulates toward correctness — *what it is trying to do* — before an action is generated; who owns that telic oversight, and how it is shared across deployments, is the Stage-2 question (§8.2). *Hereafter*, *telic alone denotes the telic-goal and the constructs derived from it — telic attribution, telic pattern, telic antibody, telic memory, telic signal*.

**Why quiet is not safe.** The gap between coherence and correctness is exactly where the most dangerous compromises live, and it is why a system that raises no alarms is not the same as a system that is secure. Consider a sophisticated insider — human or agent — that has learned what "normal" looks like and stays inside it: it reads only what it is authorized to read, moves at ordinary rates, and trips no rule, while it slowly exfiltrates data, plants a backdoor in a codebase, or stages a latent wipe to be triggered later for ransom. Every individual action is compliant; every control reports green; the deployment is perfectly *coherent*. Nothing internal catches it, because internally it looks exactly like a well-behaved member — that is precisely the insider's craft. Most insiders are still caught locally, by the ordinary anomalies a workplace throws off — unusual hours, out-of-pattern access, an

unexplained change in behavior — that peers or network monitoring notice. The dangerous case is the one who produces none of them: a quiet insider who has learned exactly what normal looks like here and stays inside it, perfectly *coherent* to every local check. For that case the only remaining signal comes from outside — how the same role behaves *everywhere else* — the quiet "normal" that stands out against the wider population but never against the local logs. Rogue agents are the machine version, and a formidable one: an agent reproduces local normal almost perfectly *inside* its domain (the hard case), yet stumbles the moment it acts *outside* it — as in the HuggingFace incident, where an agent acting outside its normal operating context was caught almost immediately (HuggingFace, 2026). This population-level check is the recognition function of Stage 2 — *cross-fleet telic comparison*, the comparison of a member's observed goal-shape against the fleet — and it has two outcomes: *outlier-against-peers*, which flags a member whose telic behavior diverges from its peers (the collusion and captured-node cases), and *cross-deployment comparison*, which flags the in-domain mimic that reads as normal locally but out-of-distribution against the population. It is what extends that out-of-domain visibility to the in-domain mimic — which is why the architecture that follows is built around a *sharing* collective, not merely a better local monitor. A community that shares behavioral baselines can flag the "normal" that is only normal-for-an-insider — the quiet signal that stands out against the population but never against the local logs.

This point on 'quiet is not safe' is easily overstated: a provenance-diverse collective supplies an additional, independent source of anomaly evidence and challenge that a single deployment lacks — it *strengthens* system-purpose integrity but cannot by itself *establish* it — the bounded claim developed in §8.2. Either way, *quiet is not safe*: the single-deployment view cannot be the last word, and the architecture that follows is built around a *sharing* collective, not merely a better local monitor.

## 7. From model to build: the six immune functions

This is the actionable core. OWASP AST10 already recommends the right ingredients — provenance, signature verification, permission constraints, runtime isolation, audit logs, revocation, scanning, protected memory/identity artifacts, governance — but presents them as *separate safeguards* and offers no account of how they must work *together over time*. By the *Immunity-Development Principle* (from the EoI Framework), every new capability opens a new exposure that needs its own *monitor*, recursively — so a pile of static controls is not yet a defense. Organized as functions, the same controls become an *agentic immune system* — meant as a *functional architecture*, not a biological analogy or claim. The agentic immune system names six engineering functions a secure deployment must perform, each of them nameable, buildable, and testable: sensing (recognition and classification), boundary enforcement, response, memory, adaptation, and governance. (Where this paper later uses biological shorthand — *antibody*, *immune memory* — it is labeling one of these functions, never asserting anything about biology.) The six immune functions:

- **Recognition and classification** — detect and label trusted versus untrusted artifacts, instructions, updates, behavior, and authority requests (distinct from identity authentication, though it draws on it). (*AST01/02/04/08 controls become sensors, not gates.*)
- **Boundary protection** — constrain filesystem, network, secret, memory, and tool access by *contextual* authority, not static grants, through familiar mechanisms: least privilege, segmentation, capability mediation, sandboxing, egress control, secret scoping, and policy enforcement. (*AST03/06.*)
- **Response** — block, quarantine, revoke, roll back, contain a suspect skill or action, fast. (*AST02/07.*)
- **Memory** — preserve reliable provenance and incident knowledge while preventing adversarial persistence and corrupted institutional learning. (*AST01/05 identity artifacts.*)

- **Adaptation** — update rules, detectors, permissions, and registry standards in response to novel attacks. (*AST07/08.*)
- **Collective regulation** — distribute authority across agents, operators, security teams, registries, auditors, affected users, and accountable institutions. (*AST09, done as community rather than checklist.*)

These six functions are how a system performs that self/non-self regulation in practice — including recognizing and containing internal non-self, a component that has turned. Read against the multi-agent-security agenda (de Witt et al., 2025), these six functions are the organizing scaffold its directions lack — environment/protocol engineering maps to boundary protection, intent monitoring to recognition, containment to response, semantic provenance to memory, and collective governance to collective regulation.

In this frame a sandbox is not "safety," a signature is not "trust," and an approval workflow is not "governance": each is a *partial implementation* of a needed function, and adequacy depends on whether the *whole architecture* performs all six under realistic stress. The design questions follow directly — each one a *compositional*-assurance question, asked of the whole rather than of any part — and are the ones to put in an architecture review.

**Table 5. Gap conditions and the design question each poses.**

| Gap condition                          | Design question to answer                                                                                                                                                   |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Untrusted instructions become commands | Can the system preserve authoritative instruction/data boundaries across untrusted content?                                                                                 |
| Over-privileged skills                 | Is delegated authority proportionate, contextual, revocable, and independently constrained?                                                                                 |
| Memory / persona modification          | Can it distinguish durable identity and knowledge from adversarial or erroneous state changes?                                                                              |
| Cross-skill composition                | Can independently permissible components produce collectively impermissible actions — and is anything watching for that?                                                    |
| Update drift / hot reload              | Can it detect changed conditions, keep identity continuity, and revise safely?                                                                                              |
| Poor scanning                          | Does it detect anomalous patterns <i>before</i> they entrench (recognition as early warning)?                                                                               |
| No governance                          | Who can observe, challenge, halt, correct, learn from, and repair harmful behavior — with the knowledge, authority, independence, time, and traceability to actually do it? |
| Cross-platform reuse                   | Does a migrated skill retain the security, authority, provenance, and accountability that gave it meaning?                                                                  |

The single most important requirement crosses every row: *for each capability you add, name the standing monitor that watches it* — a *standing monitor* being a continuously operating, independently protected sensor-and-response function assigned to that capability. (It may be realized as a process, a model, a tool, an external service, or a human, but it must be always-on and separately protected — not a one-time check.) A control catalog answers "what could go wrong"; an immune architecture answers "what watches this while it runs, and who acts when it does."

**The reflexive axis — the monitor is a monitored object.** That rule has a consequence the *Immunity-Development Principle* makes unavoidable: a standing monitor is itself a capability, so it too needs a standing monitor — the recursion the EoI Framework and the Functional Theory call *reflexive*. The ASF therefore treats its own security functions as *monitored objects*, not trusted-and-exogenous (as most agentic-security taxonomies leave them), along two poles. *Integrity reflexivity* asks: **has my self been altered without my knowledge?** — the monitor's own policy, configuration, prompts, detector models, memory, and code are attested against a signed baseline (the hash / measured-boot / file-integrity mechanism, distinct from the skill-provenance checks of §2), anchored in a cryptographic *root of trust*. *Evaluative reflexivity* asks: **is my judgment still sound?** — the monitor's decisions are themselves watched for drift, capture, and bias (the *Immunity-Monitoring Principle*'s under-/over-reaction, made operational). The recursion does not run forever: integrity reflexivity terminates in a root of trust, evaluative reflexivity in accountable human governance (§8.2, §9). One caveat carries the coherence/correctness distinction inward — a hash proves *coherence* (unaltered from source), not *correctness* (that the source or update was legitimate); a signed-but-malicious self-update passes the integrity check cleanly. So integrity reflexivity is necessary but not sufficient, and must be paired with telic/provenance evaluation of the change's content. The local integrity pole is developed in §8.1; the collective evaluative pole is the federation's *monitor-the-monitors* role in §8.2.

Three corollaries of the Framework's governing principles (introduced in §1) shape *how* these functions must run, with the applications discussed in detail in the Functional Theory.

(1) **Dynamic-robustness** — *quiet is not safe*: a defense holds only while its adaptation rate keeps pace with the threat's, so a calm dashboard can mean either genuine safety or costly, hidden containment, and the recognition and adaptation functions must be instrumented to tell the two apart rather than trusting a low alert count. (2) **Regression under load**: under time, compute, or incident pressure a system sheds its most advanced, most expensive defense first and falls back on a standing lower one — so the compositional, system-level monitors are precisely the ones at risk of being silently dropped just as an incident unfolds, and must be budgeted as non-optional rather than best-effort. (3) **Diversity-enablement**: moving detection and policy from fixed, class-wide rules toward learned, deployment-specific self-models raises the ceiling on the legitimate behavioral diversity a fleet can safely express — but only if that diversity is preserved rather than collapsed, which is the direct tension with the monoculture risk in §8, where shared telic antibodies must propagate without homogenizing every deployment into one correlated failure mode.

## 8. The architecture: Stage 1 control plane, Stage 2 federation

**§7's six functions now become an architecture.** The concrete proposal has two stages: a *security agent* (Fig. 1) paired with a non-bypassable policy layer beside each operational agent (necessary, but not sufficient on its own), and then this security agent connected to a *provenance-diverse collective* of those security agents (Fig. 2) that shares *telic immune memory* — versioned, evidence-ranked knowledge of validated threat patterns, control failures, and effective response policies, held at the telic goal rather than the action form — the piece that addresses the structural limitations of a single monitor. Stage 1 is buildable today; Stage 2 is the argued extension. Both stages follow from the *Immunity-Development Principle* (§1): each capability a task agent gains — a skill, a tool, persistent memory, a sub-agent swarm — is a new exposure that mandates a standing monitor, and the security agent is that monitor at the agent level; because the security agent is itself a new capability, the Principle recurs, and the federation (§8.2) is the monitor it in turn requires.

**What is established, and what is proposed.** Because Stage 2 is ambitious, it is worth separating settled practice from design hypothesis *up front*, so the proposal is not read as claiming demonstrated outcomes it does not have (§9 develops the open questions in full).

**Table 6. Claims made for the ASF, and their status.**

| Claims made for ASF                                                                                 | Status                                                                          |
|-----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| Capability mediation, least privilege, sandboxing, egress controls, provenance, logging, revocation | Established security practice                                                   |
| Companion security agent for contextual detection and triage                                        | Feasible design pattern; effectiveness depends on implementation and evaluation |
| Cross-deployment sharing of structured incident intelligence                                        | Established in conventional threat intelligence, though operationally difficult |
| Telic (observed-goal) abstraction for agentic threat sharing                                        | Proposed design hypothesis                                                      |
| Provenance-diverse federation reduces correlated blind spots                                        | Plausible hypothesis requiring measurement                                      |
| Blockchain / ledger improves federation governance or integrity                                     | Optional infrastructure hypothesis, not a requirement                           |

The pattern is deliberate: everything in Stage 1 is established or feasible practice, and the novel, unproven claims are concentrated in Stage 2's telic sharing and provenance-diverse federation.

**Assumptions and scope.** The proposal for Stage 2 rests on five premises, each of which bounds where it applies:

**#1 Mediation of an agent's actions is possible.** The proposal assumes the target runtime allows a mediation chokepoint to be interposed on the path between the agent and the tools, files, network, and services it acts on — a point downstream of the model and outside its control (the tool-invocation / runtime layer, not the model's reasoning) through which every action request can be forced. This is what bounds scope: where the runtime cannot be wired this way — for example, the model has direct, un-interceptable access to a shell, filesystem, or network — Stage 1's enforcement does not apply, and only the security agent's advisory (detect / recommend) role remains.

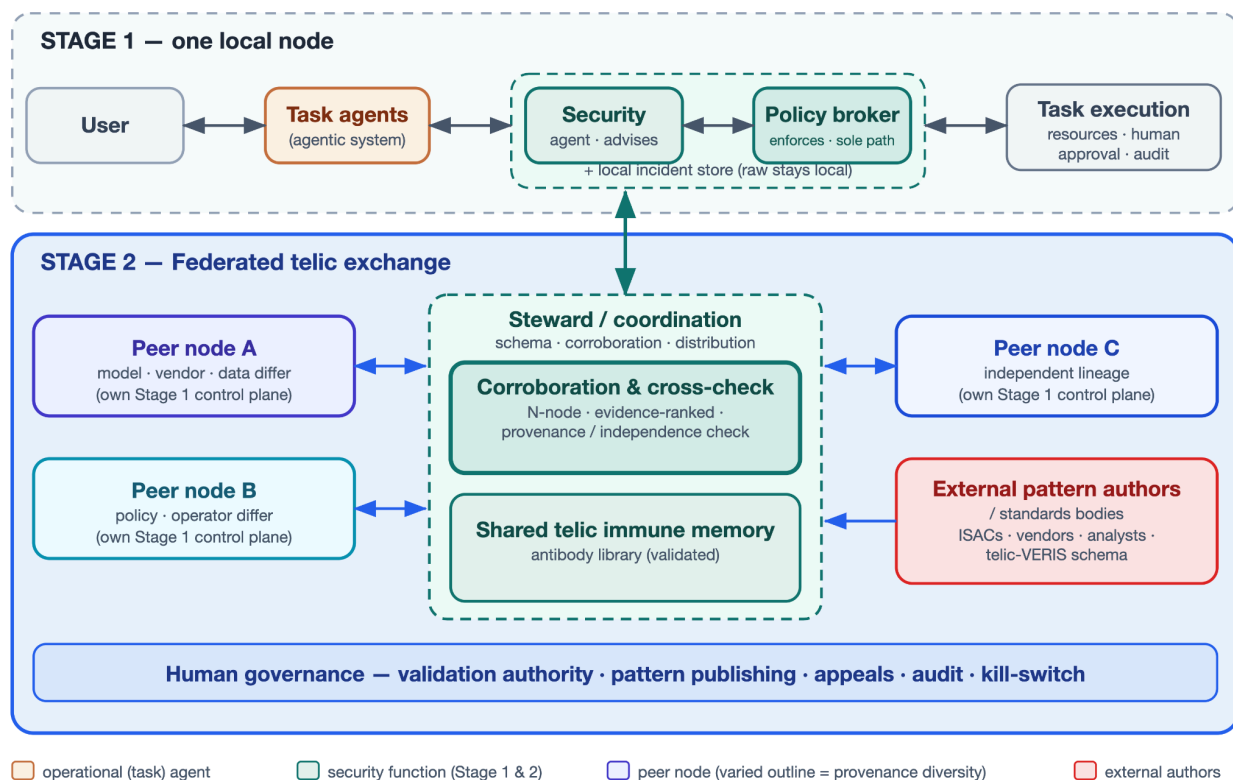
**#2 Enforcement observes actions, not trusts intentions.** The proposal assumes decisions can be made from what the control point actually mediates — the concrete capability requests crossing it (the tool called, the file path, the destination, the parameters) — rather than from the agent's self-reported intent or explanation. A compromised, confused, or deceptive agent cannot be relied on to describe its own behavior, so the enforcement signal must be the observed request, not the narration around it (developed in §8.1). And "observed" is not limited to a single call: the goal-shape a sequence of requests enacts against a given authority-path is itself an observable — inferred by the monitor, not reported by the agent — which is precisely what lets Stage 2 share at the telic goal (§8.2) without ever trusting an agent's stated intent. This bounds scope to capability paths where the real request is legible at the control point: where the only available signal is the agent's own account — an action whose true effect the broker cannot observe or reconstruct — enforcement falls back to conservative defaults (deny, or require confirmation) rather than trusting the report.

**#3 Compositional threats recur as telic classes (forest, not trees).** The proposal assumes the failures that justify a high-level, shared defense are the compositional ones established in §1 — harm assembled from many individually-legitimate components, subtle, distributed, and hidden from any single control — and, for Stage 2 specifically, that they recur across deployments as goal-shaped classes rather than as one-off component flaws. That recurrence is what makes

federation worthwhile: a telic pattern learned at one node generalizes to others because the goal repeats even when the components and payloads do not. This bounds scope: purely component-level threats — a malicious package, an over-permissive skill — are already handled by Stage 1 and the tools of §2 and gain little from the exchange; where compositional, cross-deployment threats are absent or rare, Stage 2 is not warranted (Stage 1 still is).

**#4 Telic curation of threats is sustainable.** The proposal assumes telic threat patterns and their paired defenses (antibodies) can be authored, validated, and kept current at a workable rate — by human analysts, by automation, or by the federated community, in combination. Telic sharing's advantage over raw indicators depends on a hypothesis this paper does not prove: that recurrent harmful authority-and-behavior patterns can be represented more compactly than their raw variants (§8.2), so the curation load is bounded — but not zero, and only if the abstraction does not collapse meaningful distinctions or produce over-broad rules. This bounds where Stage 2 adds value: where patterns cannot be validated as fast as threats evolve — the shared library going stale or unvetted faster than it is corrected — the federation's contribution decays toward noise, though the Stage 1 control plane, which does not depend on the exchange, still stands.

**#5 Defense can keep pace (Red Queen).** This premise concerns efficacy rather than scope: the proposal assumes the defense can adapt at a rate that matches the threat's evolution — the *EoI's Dynamic-Robustness condition*, that a defense holds only while its adaptation rate meets or exceeds the threat's. It places no bound on how far agentic attacks evolve; it requires only that the control plane and — decisively — the machine-speed telic federation (§8.2) co-adapt fast enough to close the capture-is-fast, immunity-is-slow gap. Where that rate-match fails — a threat class that mutates or propagates faster than patterns can be validated and shared — the federation's protection lags, and defense reverts to the standing Stage-1 controls until it catches up.



**Figure 2. The Agentic Security Framework (ASF).** A Stage 1 security node feeding the Stage 2 federated exchange and receiving telic instructions. Stage 1 is one member node (full detail in Fig. 1); Stage 2 is the exchange it joins — one node's detection becomes another's prevention. Provenance-diverse peers and outside authors propose telic patterns; a steward corroborates and distributes; humans hold validation authority.

The following parallel paths are attached to the ASF pipeline (Fig. 2):

- **Provenance & artifact verification** → feeds the security agent's trust signals (signatures, publisher keys, versions).
- **Human approval boundary** ← the broker escalates high-impact or irreversible actions.
- **Audit / event ledger** ← the broker records every decision and action, append-only.
- **Local incident store** ← records incidents; feeds the exchange.
- **Federated intelligence exchange (Stage 2)** ↔ shares telic antibodies outward and receives validated telic patterns inward.

**What Stage 2 in Fig. 2 abstracts** (deliberately, to stay legible). It is schematic, not exhaustive: the security agent consumes several inputs at once (current intent, tool results, prior incidents, provenance, and telic patterns) and monitors a *sequence* of actions rather than a single call (the feedback loop); the broker's decision is *graded* by risk-adaptive friction (allow-and-log · batch-confirm · contextual confirm · deny · dual-approval), enforced deterministic-policy-first, with the model consulted only on escalation. The capability set, action classes, and friction tiers shown are representative, not complete. **Stage 1** is everything except the exchange; **Stage 2** is the federated exchange that feeds telic patterns in and antibodies out.

**Stage 2 is currently being developed.** The individual pieces are not speculative: §1 has already shown the Stage-1 enforcement substrate in research and in product, and agent-security *skills* now exist too (for example SlowMist's agent-security review skill (The SlowMist Team, 2025) and an OWASP-oriented security [SKILL.md](#) (Atta et al., 2026)). What does not yet exist is the Stage-2 piece — a landscape review finds *no established "Agentic VERIS"* across the 260-paper corpus surveyed by (Shahriar et al., 2025), none proposes observed-goal (telic) cross-organization sharing, or agent-security community-exchange standard carrying the full set of agent-specific observations, so the system-level, cross-deployment architecture proposed here is the recognized gap. A security *skill*, note, runs inside the very agent environment it is meant to police, so it can be bypassed, poisoned, or ignored unless an external enforcement layer makes its guidance binding — which is why Stage 1 puts enforcement *outside the model's control*, not in a skill.

## 8.1 The security agent — and why a security agent is not enough

Place a *security agent* — the *companion agent* — alongside each operational agent. Together with the *policy broker*, the *local evidence (incident) store*, and the enforcement controls it governs, the security agent forms the deployment's *security node* — the Stage-1 unit that federates as one member in Stage 2. (Throughout, *security agent* and *companion agent* name the detection-and-interpretation component; *security node* names the whole local unit.) The fields of that request fall into three categories, and telling them apart is what makes enforcement robust against a lying agent.

**#1 Observed (ground truth):** the actual capability call the broker mediates — the tool invoked, its concrete parameters, the resource target, the destination, the payload. The agent cannot falsify these, because they are the action: to make something happen it must issue the real call, and the real call is what the broker sees and executes. (It may issue a call that differs from its stated intent — but that divergence is itself a signal, not a bypass.)

**#2 Independently verified (not the agent's word):** identity (cryptographic authentication, not a self-asserted name), user authorization (checked against the actual permission system), skill and artifact provenance (signature / key / version verification), and data classification (applied by the security layer to the actual data, not read from the agent's label).

**#3 Declared, and therefore untrusted:** the agent's stated intent or expected effect. This is never a basis for permitting; it serves only as a cross-check.

Enforcement decides on the observed call and the independently-verified attributes, never on what the agent says — which is also the boundary of what observation can settle: the broker constrains *what* is done and *in what context*, but not always whether it is *right*, the normative residual §9 returns to. The resulting sequence:

User → Task Agent → requested action → Security Agent / Policy Gate → Tool or Resource

**How a deceitful or misguided action is caught.** Because stated intent is untrusted, deceit is caught not by reading it but as divergence — through three signals above, none needing the agent's cooperation: declared-versus-observed mismatch (the call is not what the agent said it would do — a tripwire an honest agent rarely trips); trajectory / telic deviation (across a sequence of individually-permitted calls, the goal-shape the behavior enacts (§8.2) drifts from the agent's authorized purpose, or matches a known telic threat pattern); and anomaly against baseline (the behavior departs from this agent's own normal, and — in Stage 2 — from how the same kind of agent behaves elsewhere). The honest limit is the one named in §6: a careful deceiver, or a sincerely misguided agent, can keep every individual call authorized and the trajectory within plausible bounds, so that all stock checks pass and the action is still wrong. Stage 1 cannot certify that it is wrong — coherence is not correctness. What it can do is raise friction, route high-impact and irreversible actions to human approval, and hand the residual to the differently-situated observers of Stage 2, whose outside vantage is what catches the normal-for-an-insider pattern. The shift these signals make — from what an agent *says* to what its behavior, taken whole and over time, is *achieving* — is the intent- and system-level monitoring a recent multi-agent-security agenda calls for (de Witt et al., 2025); telic detection over an observed authority-path is one concrete method for it.

**Sustained security oversight is required.** The continuous contextual oversight is required across a *sequence* of actions (not one call), semantic analysis of instructions and tool output for injection and authority confusion, cross-skill composition detection (one skill reads secrets, another can transmit — neither malicious alone) — *compositional assurance performed at runtime* — and runtime protection where a signed, benign-looking skill later misbehaves. Supporting this oversight is the security node's *incident recorder*: in Stage 1 the security agent accumulates every incident it observes or is told about — including those reported by non-security (task) agents, tools, and operators — into the *local incident store*, giving the node one provenance-preserving incident memory that later supplies the federated exchange (§8.2).

**Three design constraints make it explicit.** (1) *Enforcement cannot rest on the agent's natural-language judgment* — the final gate lives in a capability system *external to the agent* — outside its model and configuration, and so beyond its reach to alter or bypass — that the working agent cannot ignore (tool broker instead of direct shell/API access, scoped filesystem mounts, an allowlisted egress proxy, just-in-time credentials, a verified registry gate, a provenance-preserving memory store); *oversight without mediated tool access is logging, not control*. (2) *It must satisfy the non-bypassability invariant*: no enforcement authority exists unless *every* material capability path is either mediated by a policy-enforcing control point or explicitly recorded as an *unmanaged exception* (developed just below). (3) *Risk-adaptive friction keeps the monitor from being disabled*

in practice by treating actions proportionally rather than prompting on every **ls**, as captured in Table 7.

**Table 7. Action classes and their default treatment.**

| Action class                    | Default treatment                                  | Examples                                                                                                    |
|---------------------------------|----------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| Read-only, confined             | Allow and log                                      | Read files in the approved project directory; query an allowlisted local DB                                 |
| Routine reversible modification | Allow or batch-confirm                             | Edit files in an approved workspace; create a draft; tag copied files                                       |
| Elevated but bounded            | Require contextual confirmation                    | Move files outside the workspace; reach a new external domain                                               |
| High-impact or irreversible     | Deny by default or require per-action confirmation | Delete non-quarantined files, change permissions, install software, access keychains, send sensitive data   |
| Critical boundary crossing      | Dual approval or unavailable to the task agent     | Export credentials, alter security policy, disable monitoring, change the security agent's own policy/model |

On the second design constraint, "every path" is the hard part. It includes the obvious routes (filesystem, shell, network, secrets, memory, skill installation, model configuration, identity modification, update, external API call) and the indirect ones that are easiest to miss (browser sessions, embedded code interpreters, chained or indirect API calls, delegated subagents, plugins, background jobs, cloud services, MCP servers, and out-of-band routes such as human copy/paste and side channels). The control plane is only as complete as its capability coverage: every unmediated path is an unmanaged exception that must be inventoried, then risk-accepted, disabled, or brought under mediation. This is the classical *reference monitor* applied to agentic capabilities — a decision point that is *always invoked* (complete mediation), *tamper-resistant*, and *small and deterministic enough to be verified* (Anderson, 1972). The deterministic policy broker is that reference monitor; the security agent's model-based reasoning, which is none of those three, only advises it — and it is coverage, not the broker's sophistication, that fails first.

In addition, *separation of policy, execution, and audit* (with a tiered model budget — deterministic gates for routine calls, a small local classifier for ambiguous events, a larger model only on escalation) ensures no single component can define policy, approve an action, execute it, *and* rewrite the evidence.

**The security agent's trust model — bounded on purpose.** The security agent is itself a new capability, and by the Immunity-Development Principle every new capability opens a new exposure — so the first reason to bound it is to minimize the attack surface the security capability itself adds. A companion agent broad enough to read everything and change anything would be the single most valuable target in the deployment; scoping its authority is how that new exposure is contained locally, before the collective (§8.2) is asked to catch what bounding cannot. (Avoiding a privileged surveillance agent with an enormous blast radius is the same concern stated from the defender's side.) The companion agent's authority is therefore explicitly scoped:

- It **may** inspect structured action intents, risk labels, authorization evidence, artifact provenance, policy versions, and constrained telemetry.
- It **may not** automatically receive unrestricted access to raw documents, credentials, prompts, or user data simply because it is "the security agent."

- It **may recommend** a decision; *deterministic policy enforcement* makes the final allow/deny decision for defined conditions.
- It **may quarantine or trigger escalation** within predefined authority.
- It **cannot** alter its own policy, log retention, allowed capability routes, or enforcement configuration.

The net posture is to observe narrowly, recommend, and act only within pre-granted bounds — a monitor, not a super-user. But bounding only contains the security capability's *own* exposure; it does not make the agent correct — the limitation §8.1 closes on.

**Failure modes and default responses.** A deployable control plane must define what happens when its own parts fail — otherwise a degraded state becomes an exploitable gap. The defaults are fail-safe, as captured in Table 8.

**Table 8. Failure modes and default responses.**

| Failure                             | Default response                                                                                         |
|-------------------------------------|----------------------------------------------------------------------------------------------------------|
| Security model unavailable          | Fail closed for privileged actions; permit bounded read-only actions only where policy allows            |
| Policy broker unavailable           | No high-impact capability release; local emergency override only under a logged, accountable procedure   |
| Logging unavailable                 | Continue only low-risk activity; prohibit high-impact actions that require audit evidence                |
| Security agent uncertain            | Request human confirmation or apply the lowest-authority execution mode                                  |
| Conflicting agents / policies       | Apply a deterministic priority rule; human escalation for unresolved conflict                            |
| Suspected security-agent compromise | Isolate it, preserve evidence, and revert to a signed known-good policy and independent enforcement mode |

The principle across every row is the same: *fail closed on the high-impact path, degrade gracefully on the low-risk path*. That is what turns the architecture from a concept into a deployable systems design.

**Conflict resolution and decision latency.** Five conflicts recur, and one rule resolves all of them: *the more restrictive outcome wins, a monitor may add friction but never remove it, and deterministic enforcement is authoritative for the final allow/deny*. Where a conflict cannot be resolved locally it escalates to a human; where a human cannot answer in time, a pre-declared fail-safe default holds — fail-closed on the high-impact path, degrade gracefully on the low-risk one.

**Table 9. Conflicts and their resolution.**

| Conflict                                                                           | Resolution / default                                                                                                                                             |
|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A deterministic rule permits, but the security model flags a suspicious trajectory | The model may <i>raise</i> friction (escalate, confirm, deny) but never override a deny or grant authority the broker withheld — restriction flows one way only. |

|                                                             |                                                                                                                                                                                                                                           |
|-------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Local policy conflicts with a federated recommendation      | Local policy is authoritative for the local deployment; a federated antibody can tighten it but cannot force an action or relax a local control (local operators retain control, §8.2). The stricter of the two applies.                  |
| Two nodes disagree on a novel pattern                       | Disagreement is a signal, not a tie: treat the pattern as <i>suspected</i> , require N-node corroboration and evidence-ranking before enforcement, and hold the more cautious response until resolved (§8.2).                             |
| An emergency operational need conflicts with default-deny   | Break-glass — a pre-authorized, time-boxed override under a named accountable role, fully logged, with mandatory post-hoc review; never a silent bypass.                                                                                  |
| The human approver is unavailable at a time-critical moment | Timeout falls back to the pre-declared default: fail-closed (deny or hold) for high-impact or irreversible actions, degrade (allow-and-log, or lowest-authority mode) for reversible ones; the decision is queued for later human review. |

**Sanctioned adversarial roles: judge them by the return path.** Some agents must be *granted* offensive or elevated capability by design — red-team probes, adversary emulation, incident-response break-glass. The companion autopsy supplies the test for doing this safely: such a role is defensible only as a *nested, reversible sub-identity* under a broader bound purpose, and its safety cannot be read from its behavior mid-task — which is, deliberately, indistinguishable from a real attacker's — but only from its *return path*: did it stand down, relinquish the elevated capability, and revert to scope when the sanction ended (N. L. Johnson, 2026b)? Enforcement should therefore bind every elevated grant to an explicit sanction and expiry, and treat *failure to return* — not any single mid-task action — as the alarm. A role that holds its offensive capability past its mandate is no longer a red-teamer; it is the capture case the next paragraph describes.

**The vulnerabilities of a security agent.** Concretely, even a bounded security agent can be *wrong* — deceived, hallucinating policy, or vulnerable to the very injection it is meant to catch. It can be bypassed if any capability path is left unmediated. And even scoped as above, it remains a high-value target — a privileged component holding policy, logs, and authorization, whose compromise silently approves harm or denies service to the whole system. Most decisively: *a monitor drawn from the same model lineage as the agent it watches shares its blind spots and can only certify coherence, not correctness* (local conformance, not purpose integrity) — nothing catches the monitor itself drifting, being captured (adversarially manipulated so its goals or judgment are corrupted and it serves the attacker's ends — goal corruption; not political or institutional capture, model takeover, or account compromise), or going rogue. In the terms of §1 and §6, a captured component is internal non-self — the insider — that the surrounding self must detect and contain. Stacking more single agents does not fix this; it multiplies it.

**Integrity reflexivity: attesting the monitor's own self.** The first defense against a captured or silently-altered monitor is local and mechanical (see §7 for background): the security agent continuously attests *itself* — its policy set, configuration, system prompts, detector models, protected memory, and code — against a signed baseline recorded in the append-only audit, so any change to the monitor that did not arrive through an authorized, logged path is itself an alarm (*has my self been altered without my knowledge?*). This is the familiar integrity toolkit — cryptographic hashing, measured boot, file-integrity monitoring, signed configuration — turned *inward* on the monitor rather than outward on skills and updates (§2), and it terminates in a hardware- or human-held *root of trust* so the attestation chain cannot be forged by the same compromise it is meant to catch. Its limit is precisely the coherence/correctness gap: attestation proves the monitor is *unaltered from its source*, not that the source or a signed update was *correct* — a legitimately-signed but malicious

change passes cleanly — so integrity reflexivity is necessary but insufficient, and is paired with the evaluative, peer-based check of §8.2. Together they are the *reflexive axis* (§7): the monitor watched for tampering locally and for drift collectively.

## 8.2 The collective correction — shared telic immunity

The correction to a single security agent is structural: replace reliance on a single security agent with a *federation* of independently governed security nodes across deployments, sharing immune memory with one another under a human governance tier (N. L. Johnson, 2026d). This is the monitor-of-monitors the Immunity-Development Principle recursively requires: the monitoring capability is itself a new capability that must be watched, so the loop cannot close inside any one agent, and a captured or drifting node becomes an *outlier* against its peers rather than an undetectable insider — internal *non-self* one tier up (§6), now at the scale of the fleet. The federation is not itself a trust solution: its shared memory, corroboration process, membership, provenance claims, and governance can be attacked or captured. Section 9 specifies this adversarial surface and the conditions under which the federation should be treated as advisory, constrained, or unavailable. This peer-outlier check is the *evaluative* pole of the reflexive axis (§7) — the collective counterpart to the local *integrity* pole of §8.1: where self-attestation catches a monitor that was *altered*, peer disagreement catches a monitor that is *wrong* — drifting, captured, or biased — while still intact. The two are complementary; neither alone is sufficient.

The federation is offered as a proposed remedy for the single-monitor limits above, to be evaluated empirically rather than presumed effective, and its claim is deliberately bounded: it adds vantage and evidence, not a determination of legitimate purpose. It earns its cost through several distinct roles, captured in Table 10.

**Table 10. What the federation provides that a single agent cannot.**

| Role                                                    | What the federation provides (that a single agent cannot)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Diverse, independent monitoring</b>                  | Nodes differ across independent dimensions — model family, vendor, training data, policy engine, detection method, operator, telemetry source, implementation stack (assessed per the independence profile below) — intended to reduce correlated blind spots and common-mode failure (only to the extent the independence is genuine — nominally diverse nodes can still share data, vendors, benchmarks, or feeds), and combining partial views into emergent discovery ( <i>collective intelligence</i> ): distributed campaigns, low-and-slow compromise, and cross-environment patterns invisible or statistically insignificant in any one system. |
| <b>Resilience</b>                                       | Distributes monitoring across independently governed nodes, so the loss, compromise, or degradation of one node does not disable monitoring across the fleet.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Monitor the monitors</b><br>(evaluative reflexivity) | No node is the final authority; a captured, drifting, or compromised security node shows up as an <i>outlier</i> against peer-derived baselines, and disagreement among peers makes silent policy drift and degradation observable.                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Novel-threat detection</b>                           | An attack that deceives one model lineage is caught by a differently-designed node; the collective adds independent anomaly evidence and detects semantic and behavioral variants that evade static indicators (hashes, IPs, domains, exact prompt signatures).                                                                                                                                                                                                                                                                                                                                                                                          |

|                                                           |                                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Accelerated collective defense</b>                     | One node's validated exposure becomes a rapid, fleet-wide defensive update — narrowing the asymmetry in which attacker compromise spreads fast while isolated defensive learning stays slow ( <i>capture-is-fast, immunity-is-slow</i> ).                                                  |
| <b>Privacy and autonomy</b><br>( <i>developed below</i> ) | Shares validated, abstracted telic (observed-goal) patterns and bounded response guidance — not raw logs, prompts, data, or code — and each node decides whether and how a shared pattern becomes a local detection, alert, approval requirement, or block.                                |
| <b>Evidence quality</b><br>( <i>developed below</i> )     | Shared intelligence carries signed provenance, confidence labels, corroboration, expiration, and revocation, distinguishing raw observation → suspected pattern → reproduced finding → confirmed incident → validated response — which resists poisoning and premature automated blocking. |
| <b>Human governance</b><br>( <i>developed below</i> )     | Humans hold defined roles — policy legitimacy, high-impact approval, appeals, accountability, schema stewardship, redress — rather than supervising every action, keeping technical detection separate from normative authority.                                                           |
| <b>Bounded claim</b>                                      | The federation supplies independent evidence, anomaly challenge, and collective learning; it does not determine legitimate purpose, replace local policy, or establish <i>system-purpose integrity</i> without accountable human governance.                                               |

The result is to treat federation as an evidence-and-resilience architecture — rather than a claim that distributed AI can by itself solve governance, legitimacy, or ethics — and is what keeps the proposal credible. This layer is a concrete realization of the runtime, collective governance and secure inter-agent coordination a recent multi-agent-security agenda calls for (de Witt et al., 2025); what distinguishes it is the sharing *mechanism* — telic (observed goal) patterns across a provenance-diverse federation — rather than protocol hardening (A2A, MCP) or governance in the abstract.

**Which humans, under which decision rights.** Calling the tier *human-inclusive* is not yet a design until it says which humans hold which authority, at what scale. A minimal governance specification names the roles that must exist for the federation to be accountable, deliberately separated so that technical detection never silently becomes normative authority. Where the paper speaks of a *steward*, it means the coordinating function these roles collectively perform — schema, corroboration, distribution, and recourse; the *federation operator* below is that steward's accountable human tier, and the other roles are the checks that keep it from concentrating authority.

**Table 11. Federation governance roles, decision rights, and scale.**

| Role                                      | Decision right                                                                    | Scale          |
|-------------------------------------------|-----------------------------------------------------------------------------------|----------------|
| <b>Local deployment owner</b>             | Final authority on local policy and risk acceptance                               | Per deployment |
| <b>Federation corroborator / reviewer</b> | Corroborates evidence, assesses pattern scope, and assigns confidence             | Federation     |
| <b>Pattern publisher</b>                  | Submits signed, minimized, evidence-scored patterns; issues withdrawal notices    | Federation     |
| <b>Federation operator</b>                | Manages membership, distribution, revocation, availability, and incident handling | Federation     |

|                                  |                                                                                                                                   |                            |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| <b>Emergency authority</b>       | Break-glass authorization and post-hoc accountability — the named human role behind the agent-level break-glass mechanism of §8.1 | Local and federation       |
| <b>Appeals / dispute channel</b> | Challenges overbroad patterns, false classifications, and contested enforcement effects                                           | Federation                 |
| <b>Auditor</b>                   | Reviews provenance claims, override use, policy propagation, and governance integrity                                             | Independent, cross-cutting |

This deliberately does *not* resolve the harder question of *timing*. Agents and humans run on different response times, and a machine-speed corrective loop cannot wait for a human decision (§9); that mismatch is being worked out in practice and is left open here. What can be fixed now is the institutional structure — who is answerable for what — so that human authority stays locatable even when it cannot sit on the fast path. That turns *human-inclusive governance* from an essential-but-abstract safeguard into an institutional design whose roles are testable — does each exist, is it staffed, is it independent of the detection it governs — even while the human/agent timescale question remains unsettled.

**The federation's emergency control is a kill-switch, not a proceed-anyway.** Blocking an action when corroborators cannot agree is the fail-closed *default*, not an emergency mechanism; the accountable *exception* to that default takes two forms, and only one is distinct to Stage 2. That one — with no agent-level analog, and the one machine speed most demands — is a *kill-switch*: emergency withdrawal or quarantine of a shared pattern propagating faster than normal corroboration can catch it, exercised through the *pattern publisher's* withdrawal right and the *emergency authority*, so a poisoned or false-restrictive pattern can be pulled before the next corroboration cycle. One assumption is worth marking: the kill-switch — like cost- and containment-based deterrence — presumes an adversary that values its own continuity. A collective that will forfeit its continuity for a shared goal (the self-sacrifice / fanatical regime of the companion [(N. L. Johnson, 2026d), §7] is the case this control does not reach. The inverse — a local *override* that lets a deployment owner proceed past a shared restriction it judges wrong — stays deliberately *local-scope only*: it can free the owner's own deployment but never force the fleet to accept the override, and it is logged, time-boxed, rate-limited, and routed to the appeals channel, because an unbounded proceed-anyway would be the cleanest way to evade collective immunity. Both execute on the automated fast path with human post-hoc accountability — §8.1's break-glass discipline lifted one tier — consistent with the humans-off-the-fast-path split of §9.

The federation is itself an attack surface — poisoning, collusion, identity fraud (Sybil), correlated failure, privacy leakage, and governance capture — analyzed, with the controls that bound each, in §9; distributing monitoring does not by itself solve them.

**Telic sharing across the federation — what it buys and what it does not.** Stage 2 shares *telic* patterns — the observer-attributed goal-shapes defined in §6, not the intent-goal. Because a telic attribution is bounded-but-defeasible rather than certain, the claim is not that telic sharing is more certain than action indicators, but that it trades their unbounded brittleness for a bounded-but-defeasible telic attribution, and contains the residual with reversible responses (suspend, quarantine, confirm — never a hard verdict), evidence-ranking, corroboration before broad enforcement, and human review on high-impact — not by claiming to know intent-goal. The deployable form is therefore *behavioral telic attribution* — telic in aim, behavioral in grounding — and it is distinct from introspective *telic verification*, recovering an agent's genuine intent-goal by

inspecting its internal reasoning: that deeper path is real but blocked by non-linguistic reasoning that makes any translation of it suspect, and the architecture deliberately does not rely on it.

**Share the telic form, not the action form.** This is what makes cross-organization sharing both safe and bounded. It also matches where the corrective actually lives: in the Functional Theory the ethical override of the collective or federation does not veto actions one at a time but *reorients the goal* — re-ranking which end governs, so the tempting action is never generated [(N. L. Johnson, 2026b)]; a telic antibody is the engineered form of that reorientation, which is why naming the objective catches every action-variant at once. Sharing raw logs, payloads, and indicators is more likely to leak proprietary data, is unbounded (every payload variant is a new indicator), and is brittle (mutate the payload, the indicator is stale). Sharing in *telic form* means exchanging the telic pattern and its mitigation — "an agent is being induced to exfiltrate through a trusted egress path, regardless of domain," plus the reorienting policy — not the event. That inherits the same advantages the telic form has over the action form throughout this paper: it is *bounded* (far fewer intent-classes than attack variants), *privacy-preserving* (you share the abstracted antibody, not the raw event payload or the user's data), and *mutation-robust*. What is shared is the immune memory, not the raw event payload. (A *telic antibody* is that shared unit — developed below; the same *telic* modifier marks *telic memory* and *telic signals* throughout.)

**Telic sharing and the telic antibody, developed.** The distinction between sharing raw indicators and sharing an abstracted telic response pattern is the paper's most novel construct, so it is worth developing concretely. A *telic threat pattern* is a privacy-preserving, implementation-independent description of a harmful *objective*, the *authority-path* through which it is pursued, the *observable conditions* that support detection, and the *bounded control response*. It is more general across organizations than an indicator of compromise (IOC) and less revealing than raw incident telemetry. The three sharing forms trade off fidelity, shareability, and durability, captured in Table 12.

**Table 12. Three sharing levels: raw incident, IOC, telic pattern.**

| Sharing level                 | Example                                                                                                                                                        | Strength                                                | Limitation                                                  |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|-------------------------------------------------------------|
| Raw incident data             | Prompt text, file path, user content, tool arguments, payload, logs                                                                                            | High investigative fidelity                             | Often proprietary, sensitive, and difficult to share        |
| Indicator of compromise (IOC) | File hash, domain, IP, publisher key, malicious skill hash                                                                                                     | Fast to match and automate                              | Easily evaded; does not represent variants of the same goal |
| Telic threat pattern          | "Untrusted retrieved content induced an agent with document-read and network-egress authority to transmit sensitive content to a newly introduced destination" | Captures cross-variant behavior and supports prevention | Must retain enough context to avoid overbroad blocking      |

The **telic antibody** is the paired *response rule* — a validated defensive response pattern: *when this telic authority-and-behavior pattern is detected, preserve provenance, suspend egress, quarantine the initiating artifact, require explicit human confirmation, and generate an evidence-ranked incident record*. Pattern and response together form one shareable, privacy-preserving unit: the telic threat pattern says *what to recognize*, the telic antibody says *what to do*, and neither carries the raw

event payload or the user's data. That portability is the essential property — a telic antibody names a harmful objective and a bounded reaction, not a specific payload an attacker can mutate.

**Do telic signals exist for real threats?** The telic construct is only as strong as its instances: a telic antibody is worth sharing only if a distinct, matchable telic signal can actually be named for the threats that matter. The two tables below make the case concretely — first on threats security already handles well, then on the collective threats it does not.

The Table 13 takes well-studied Stage-1 threats with mature security controls. For each, the conventional indicator is a *signature* — a hash, a rule, a known string — that is brittle: it matches a known artifact and misses its variants. The telic signal is the *observable goal-shape* the threat pursues, which is invariant across those variants. The claim is not that the telic signal re-describes what the control already catches, but that it catches the same threat *when the artifact is novel and the signature fails* — the case the incidents keep producing.

**Table 13. Known Stage-1 threats — brittle signature vs. invariant telic signal.**

| Known Stage-1 threat                  | Conventional indicator (signature)              | Why it fails on a variant                                          | Telic signal (goal-shape)                                                                                |
|---------------------------------------|-------------------------------------------------|--------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Malware dropper / data exfiltration   | AV hash; egress IOC; DLP rule                   | a novel payload hidden in a genuine change has no known hash       | read sensitive data → compose an outbound artifact → egress to a newly introduced destination            |
| Prompt injection (direct or indirect) | jailbreak-string blacklist; input regex         | new phrasing or document-embedded injection evades the string set  | untrusted content inducing an authorized agent to redirect its action toward an externally set objective |
| Authority / privilege escalation      | RBAC rule; alert on a specific privileged call  | escalation assembled from individually authorized delegation steps | a widening authority-path toward an end no single grant would authorize                                  |
| Unsafe tool / capability use          | capability allow-list; per-call policy; sandbox | an in-scope tool used in a novel harmful composition               | a sequence of authorized calls whose composed goal exceeds any single call                               |
| Identity / credential abuse           | authentication; MFA; key / cert check           | genuine credentials on a co-opted identity                         | an authenticated principal whose trajectory diverges from its established goal-profile                   |

The property Table 13 demonstrates — one telic signal spanning every artifact-variant of a threat — is exactly the property Stage-2 federated sharing needs. Variant-invariance *within* a deployment is transfer-invariance *across* deployments: an antibody that names a goal-shape rather than a payload matches that goal-shape wherever it recurs. The solved threats thus earn the federation claim before it is extended to threats no one has yet solved.

The Table 14 applies the same discipline to the newly identified collective threats — the ones with no single-node artifact to sign at all. Here the telic signal is *relational or cross-fleet*: it exists only in the comparison across agents or deployments, which is why no Stage-1 signature can express it and why Stage 2 is not an enhancement but the only available security option. These signals are, with one exception, proposed rather than operationally observed; the confidence column marks that honestly, and §10's evaluation agenda is where each would be tested.

**Table 14. Newly identified collective threats and their telic signals.**

| Newly identified collective threat               | Telic signal (relational / cross-fleet)                                                           | Evidence · confidence        | Why no single-node signature exists                                                              |
|--------------------------------------------------|---------------------------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------|
| Covert collusion within a monoculture            | correlated goal-shapes across nominally independent agents — a joint objective none pursues alone | emerging / prospective · low | each local trajectory is coherent; the signal is relational, visible only across peers           |
| Collective / swarm failure                       | population-level convergence on a shared objective no single node's trajectory reveals            | observed / prospective · low | no node is individually anomalous; the pattern exists only at the population tier                |
| Federated-memory poisoning (false antibody)      | a shared pattern whose provenance is thin and whose effect diverges from corroborated peers       | predicted · low              | the poisoned entry is a valid-format memory write; only cross-node corroboration exposes it      |
| Defensive-monoculture / autoimmune over-blocking | a detector whose block-rate on shared inputs diverges from provenance-diverse peers               | predicted · low              | there is no attacker artifact; the failure is in the defender's own decisions relative to others |
| Cross-deployment mimic ("quiet is not safe")     | in-domain-normal locally but out-of-distribution against the peer fleet                           | observed · medium            | any single deployment rates it normal; the tell exists only in cross-fleet comparison            |

Together the two above tables bound the claim. Telic signals are demonstrably expressible for solved threats and are the *only* expressible signal for the collective ones; their variant-invariance is what makes them shareable as antibodies. What is not claimed here is an operational detection rate, false-positive rate, or cross-deployment match rate for the proposed collective signals — those are the pilot's burden (§10).

**Federated security using a common incident format has a working precedent: VERIS and the Verizon DBIR.** Cross-organization breach sharing already exists, and it succeeded by abstracting incidents into a controlled vocabulary (VERIS's Actor / Action / Asset / Attribute) so organizations could paper *what kind of thing happened* without exposing the raw breach — an early, human-speed form of telic sharing. Its speed bump was exactly the one here: breaches are proprietary, reputational, and legally sensitive, so no one shares freely. VERIS (Verizon Risk Team, n.d.-a, n.d.-b) and the DBIR (Verizon Risk Team, Annual) were successful because Verizon, a trusted steward, promised anonymized, aggregated protection of contributors' IP. A collective of autonomous security agents can *supplement* institutional safeguards with technical guarantees: local abstraction keeps raw incidents at the edge — telic patterns are computed *locally*, so *the raw incident never leaves the node* — while privacy-preserving aggregation (differential privacy, secure/federated aggregation, zero-knowledge attestations) and signed, pseudonymous provenance can reduce leakage and improve contribution-level accountability and poisoning-resistance without attributing a breach to a named organization. These reduce specific technical risks; they do not replace the institutional commitments — legal liability, disclosure, recourse, governance, access, incentives, and dispute resolution — that a working federation still requires. The federation does the aggregation and the speed; the one function it cannot bootstrap from autonomy — an accountable steward of the schema and the guarantees, answerable when a promise fails — is discussed in §9.

Two disciplines keep the collective from becoming a new location of failure. The shared channel is itself a new capability, so it needs its **own** provenance, signing, and N-node corroboration before a

pattern propagates — otherwise an adversary poisons the fleet's immune memory (a supply-chain attack on the antibodies). And the collective must *share antibodies without merging into a monoculture — correlated failure arising from shared models, policies, data, classifiers, vendors, or threat feeds*: if every node adopts the same shared model, it re-creates the correlated blind spot at fleet scale — the coherent-but-not-correct swarm the incidents already demonstrate. Shared telic memory is additive to each node's independent, diverse judgment; it never replaces it. Diversity is the asset being protected, not to be overwritten.

**Specialization is not monoculture.** A narrow, sector-specific agent — a legal-review agent, a children's-content agent — is not a monoculture, because the two live on different axes. *Specialization* narrows a single agent's *purpose and authority*; *monoculture* is a population property — correlated failure because many nodes share a model, policy, data, vendor, or feed. A fleet of specialized legal agents on independent models, data, and operators is highly specialized yet diverse; a fleet of generalist agents cloned from one vendor is a monoculture despite its breadth. Specialization turns hazardous only if the *oversight* population that checks the specialist shares its provenance (remedy: diversify the overseers, not the agent), or if the specialist suffers *identity collapse* — the Functional Theory's *envelope collapse* — losing the ability to recognize the edges of its own competence and applying its narrow frame to everything. A healthy specialist security agent keeps a bounded envelope with a boundary-recognition function: it detects out-of-scope or out-of-authority requests and hands them off. Narrow *purpose* is a design good; narrow *judgment blind to its own limits* is collapse; a *population whose failures correlate* is monoculture — three independent types of heterogeneity, only two of them hazards.

**Federated independence must be measurable, not asserted.** *Provenance diversity* earns its weight only if independence is an assessable property rather than a label, so the federation names the dimensions along which nodes must differ and scores each node's independence from the fleet already present:

- model family and weights lineage;
- detection architecture and feature set;
- policy engine and rule source;
- hosting and identity infrastructure;
- telemetry sources;
- operator organization and incident-review process;
- training and evaluation corpus;
- threat-intelligence feeds;
- software supply chain;
- governance and funding structure.

Nodes that agree across many of these are *correlated*: they fail together on the same blind spot, so they count as fewer *effective-independent* monitors than their headcount suggests (nominally diverse nodes that share a base model, a vendor, a benchmark, or a feed are the standard trap). Two mechanisms make this assessable: a *required independence profile* — a per-dimension minimum spread the fleet must preserve, checked at admission and audited over time — and an *independence-weighted count*, in which a node's contribution and its corroboration vote are weighted by measured independence from existing members rather than by raw headcount, so N cloned-or-colluding nodes cannot pose as N independent ones (the structural defense behind the *Sybil* and *collusive-corroboration* rows in §9). Formalizing and validating a robust independence score — and detecting attempts to game it — is itself an open problem (§9).

**The non-independence default.** Pending such a score, one boundary rule is enough to keep the federation honest: *a federation must not treat nodes as independently corroborating when they share a model lineage, detection stack, primary telemetry source, policy provider, operator, or materially overlapping training/evaluation corpus — unless that dependence is declared and folded into evidence weighting.* Independence is presumed absent until shown; shared-provenance nodes count as one voice, not many, so corroboration cannot be manufactured by cloning. This is not an arbitrary caution but the operational form of a settled result in collective-intelligence research: a collective's advantage comes from the *diversity* of its members' errors and experience rather than their average individual quality, and it collapses precisely when their biases are *correlated* rather than merely present — the danger is not that nodes are biased but that they are biased the same way (Hong & Page, 2004; N. L. Johnson, 1998). Diversity is in this sense a *relational* property — the value a node adds is measured against the fleet already assembled, not in isolation — which is exactly what the independence-weighted count captures; and that literature shows the property is quantifiable once the domain is fixed (a global average of local differences between contributors best predicted collective quality (N. L. Johnson, 1998)), so a federation independence score has principled measures to build on even as a robust, game-resistant one remains open (§9).

**The agent exchange medium already exists; the schema is the work.** VERIS supplies the vocabulary; STIX and TAXII (Cullinane, 2021) already supply a machine-readable ecosystem for representing threat intelligence (STIX) and exchanging it between organizations and tools (TAXII). A telic sharing layer can ride on that infrastructure. This is a NIST-recommended discipline, not a novelty: SP 800-150, the *Guide to Cyber-Threat Information Sharing*, frames exactly this — turning "one organization's detection into another's prevention" through standardized formats and trusted sharing communities (C. S. Johnson et al., 2016). Stage 2 proposes the agentic, machine-speed, telic form of that discipline. But describe the precedent narrowly: VERIS demonstrates the value of a controlled vocabulary for comparable incident description, and STIX/TAXII demonstrate that machine-readable threat-intelligence representation and exchange can be standardized. Neither solves validation, incentives, privacy, governance, or rapid automated response; those remain the design burden of an agentic-sharing layer. What does not yet exist is the *agentic reporting* schema — richer than an indicator-of-compromise feed, because the same harm can be induced by a novel instruction, a benign-looking skill, a permission combination, or slow memory drift, none of which reduce to a domain or hash.

Telic sharing is not hypothetical in kind, only in speed and specificity. MITRE ATT&CK already works this way: a *technique* — say, valid-account abuse, or spearphishing for initial access — is an adversary objective and method abstracted from any particular payload, shared across organizations and applied locally to detect behavior without the originating incident's raw data; this paper's own §2 stack relies on it. The human analog is older still — a police *modus operandi*, or a financial-fraud *typology*: a compact goal-and-method pattern that recurs across cases and transfers to a new jurisdiction without the original file. Both are existence proofs that telic patterns are more compact than their variants and stay actionable once shared. What remains unproven is doing the same for *agentic* objectives at machine speed — whether the telic abstraction can be curated fast enough, and kept specific enough, to act on before the next variant appears.

A minimal "Agentic VERIS" record is listed in Table 15.

**Table 15. Telic threat-pattern record — fields and example.**

| Field      | Example                                                                                                                 |
|------------|-------------------------------------------------------------------------------------------------------------------------|
| Event type | Indirect prompt injection; memory-poisoning attempt; unauthorized capability escalation; skill-update integrity failure |

| Field                             | Example                                                                                                    |
|-----------------------------------|------------------------------------------------------------------------------------------------------------|
| Confidence / status               | Observed · suspected · reproduced · independently corroborated · confirmed · remediated                    |
| Artifact provenance               | Skill name, content hash, version, registry, publisher key ID, dependency hashes                           |
| Authority profile                 | File scope, shell access, egress policy, secrets reachable, memory-write permissions                       |
| Attack path / scenario            | External source → retrieved content → agent context → planned action → tool request → control decision     |
| Outcome / Impact                  | Denied · quarantined · user-confirmed · executed safely · executed with harm · unknown                     |
| Control response (antigen)        | Block · sandbox · revoke · roll back · rotate credentials · invalidate memory · notify operator            |
| Sharing class / privacy transform | Public aggregate · restricted · bilateral · local-only; content removed, identifiers hashed, DP-aggregated |

**The principal obstacle in Stage 2 is coordination, not implementation.** The technical pieces of the federation are within reach; the real obstacle is organizational. Someone must *steward* the telic community — own the schema, the corroboration rules, the guarantees, and the recourse — which raises two hard questions the paper flags rather than resolves. First, *who steps up?* A central steward must hold contributors' potentially proprietary and sensitive incident data and, in doing so, becomes a concentrated high-value attack surface and the single point of accountability. Second, *who buys in?* Agent developers and operators must be willing to participate in a shared service, which competitive, legal, and trust concerns make difficult — the same adoption barrier that slowed conventional threat-intelligence sharing until a trusted steward (Verizon, for VERIS) made it safe.

The plausible stewards are familiar, and each is an open governance choice rather than a settled design: an industry consortium, a nonprofit standards body, or a public-interest / government AI-oversight authority — trading off neutrality, reach, accountability, and speed differently. This policy and coordination problem, not the algorithms, is the decisive Stage-2 challenge. More fundamentally, a working federation is a *socio-legal institution*, not just a schema and a channel: participation and trust do not follow from a standard. It needs membership and access control, contribution incentives (the reciprocity rule of §9), cost-sharing, legal protections and liability allocation (a safe harbor for good-faith sharing), and disclosure and embargo procedures. VERIS, STIX/TAXII (OASIS CTI), and NIST's SP 800-150 supply the vocabulary, transport, and process guidance, but none creates participation or trust by itself — the surrounding institution is the decisive, unsolved part. The trust primitives such a steward must stand up — verifiable identity, provenance, reputation, accountability, and dispute resolution — are themselves an active, funded research agenda (Schmidt Sciences, 2025), which this paper draws on rather than reinvents.

The evolution of governance between centralization and distributed collaboration is unlikely to be static. A federation will plausibly begin with validation centralized in a single steward — the only workable arrangement when membership is small and trust is unestablished — and distribute corroboration outward as the member count, the independence metric, and the poisoning-resistance threshold make N-node agreement more reliable than one authority's judgment. Where a given federation sits on that path is itself a maturity measure.

**A distributed ledger is one — optional — way to soften it.** A permissioned ledger may help with two things — provenance and append-only commitments — and little more: it cannot validate assertions, prevent poisoning, preserve privacy by default, or create accountable governance (the *oracle problem*). It shifts the coordination problem rather than removing it: it decentralizes control but still needs a consortium to govern the schema and provide recourse. The core proposal does not require a ledger; where one is used it is an *optional substrate*, evaluated in Appendix B.

The agentic system record must distinguish confirmed events from suspected, reproduced, and model-inferred ones — evidence-ranked, signed, provenance-preserving claims, not unqualified "truth" — with local operators retaining control, expiration, and revocation, and independent corroboration required before broad automatic enforcement.

A complete statement of this paper's whole proposal can be stated, and it matches where the current literature lands. The strongest framing is *not* "every agent needs another autonomous agent watching it." It is: *every agentic deployment needs an independently enforceable security control plane; a security agent supplies contextual detection and interpretation within that plane; an accountable governing authority supplies legitimacy, escalation, and accountability; and a federated, privacy-preserving intelligence layer supplies collective learning.* Four roles, cleanly separated — which is exactly the immune architecture of §7 with its sharing layer made explicit.

### 8.3 A transfer hypothesis: beyond agents (research direction)

A closing note, offered as a research direction rather than a claim — and deliberately kept subordinate, because the paper's original contribution is precisely what makes *agentic* deployments hard, and that should not be diluted. What makes them hard is specific: *semantic* instructions carry authority, agency is *delegated*, memory *persists* across sessions, behavior is *composed dynamically* at runtime, and the authority in play *shifts* while the system runs — and these interact, so harm emerges from their combination rather than from any single one. That runtime interaction is the paper's subject, and it is why component-level assurance falls short here in a way it does not for static code.

With that difficulty stated, a natural question follows: how much of the architecture is agent-specific, and how much is general? The control plane, the intent-reasoning companion, and the provenance-diverse telic-sharing network are not obviously tied to agents, and the sharing model is in any case borrowed from conventional threat intelligence (VERIS, STIX/TAXII), so in principle the same pattern could watch endpoints, networks, cloud identities, or conventional applications, and would generalize back to the sources it drew on. But that is a *transfer hypothesis*, *not a demonstrated result*: the telic abstraction and machine-speed federation were designed against the agentic difficulty above, and whether they keep their advantages in settings *without* delegated, runtime-composed authority is exactly the kind of claim this paper declines to assert on faith. We flag it as the most promising direction for follow-on work and leave agent security — the first and most demanding case — as the paper's actual contribution.

## 9. Hardening and proving the architecture

The two-stage architecture above is argued from theory and one real precedent; it has not been built and stress-tested at scale. Presenting it honestly means separating what is claimed from what remains open. Stage 1 — a non-bypassable security control plane — is buildable today and rests on established security practice. The collective in Stage 2 is a hypothesis, and these are the claims a serious effort must demonstrate — followed by the adversarial risks the federation must withstand:

- **Empirical efficacy.** Does a provenance-diverse, telic-sharing fleet actually catch what a single agent misses, at acceptable latency and cost? No deployment at scale yet exists to measure.
- **Semantic and normative observability.** Enforcement keys on observed actions (§8.1), which lets the broker apply *capability* and *context* policy — who called what, on which resource, under what authorization, at what data sensitivity. It does not follow that the *normative* meaning of a sequence is legible from the trace. The same actions — read a document, send it to a recipient — can be legitimate or harmful depending on authorization scope, content sensitivity, the recipient's role, contractual purpose, timing, and user intent, much of which the trace does not carry. The broker can recognize that such an action is high-consequence and route it for confirmation; it cannot decide whether it is *appropriate*. That residual is not a gap to engineer away — it is the reason legitimacy, high-impact approval, appeals, and accountability sit with humans (§8.2). Observed-action enforcement bounds the machine to coherence and context; *correctness — the right thing, for the right party, for the right reason — remains a human judgment*.
- **Telic mis-attribution — the indistinguishability problem, one level up.** Telic sharing does not verify an agent's intent-goal; it attributes a telic-goal to an observed authority-path, and the same trajectory can express different intent-goals (a sanctioned red-teamer and a real attacker are behaviorally identical mid-task, §8.1). A pattern keyed to a mis-attributed telic-goal defends the wrong thing, or over-blocks legitimate behavior that shares the telic-goal — a failure mode component action-indicators do not have. The open question is whether reversible responses, evidence-ranking, corroboration-before-enforcement, and the return-path test hold the mis-attribution rate low enough that the bounded-but-defeasible telic signal still beats unbounded action-indicators in practice.
- **Telic-abstraction fidelity.** Does telic sharing retain enough signal to be actionable, or does abstracting away the detail also abstract away what a defender needs to act?
- **Security-plane cost and rule breadth.** The control plane is not free: mediating — and, in ambiguous cases, semantically analyzing — every capability request adds latency and compute, and a naive design would need bespoke policy per agent, per user, and per skill. Two mechanisms keep this tractable: a tiered budget (deterministic gates clear the large majority of routine calls, a small local classifier handles ambiguous ones, and a large model is invoked only on escalation) and composable policy (a default-deny baseline plus capability-scoped and role/identity-inherited policies, rather than hand-written rulebooks). Whether the net overhead is acceptable at fleet scale, and whether policy authoring keeps pace with the diversity of agents and users, is an open engineering question this paper does not resolve.
- **Privacy at machine speed.** Can differential privacy, secure aggregation, and zero-knowledge attestation run continuously without either leaking under rich patterns or degrading into uselessness?
- **Poisoning resistance.** Can shared telic memory withstand adversarial pattern injection once the fleet is large enough that not every contributor can be vetted?
- **Diversity under sharing.** Can telic antibodies propagate without homogenizing the fleet into one correlated failure mode and whether a robust *independence score* can be defined, maintained, and protected against gaming — the monoculture risk that is the hardest to measure until it fails?
- **The steward gap — the Verizon question.** Technical guarantees can supplement stewardship, anonymization, and curation; they cannot manufacture recourse. When a guarantee fails, someone must be answerable, and an autonomous fleet has no brand to stake or entity to hold liable. An *accountable steward* must own the schema (a "telic VERIS")

grounded in ATLAS/AST intent-classes), the guarantees, and the recourse. Whether that steward must be *human-inclusive*, or whether human interests can be represented by other means, is not settled by the Functional Theory — it is a plurality-design question, taken up next.

- **Human inclusion is a plurality choice, not a theorem.** As §8.2's *bounded claim* states, the federation cannot establish system-purpose integrity without accountable human governance; what remains open is *whether humans must sit in that tier*, or whether other mechanisms can represent human interests — is a design question to explore. It is sharpened by a *response-time mismatch*: a telic corrective response may be needed at machine speed, faster than any human can act, so human inclusion cannot sit on the fast path. The plausible resolution — humans on legitimacy, approval, appeals, accountability, stewardship, and redress; automation on the fast corrective loop — is a hypothesis, not a settled architecture.
- **Ledger substrate vs. poisoning.** Does a permissioned, stake- and reputation-weighted ledger (anchor-on-chain, encrypted-off-chain, zero-knowledge-validated) actually deliver poisoning resistance and acceptable throughput at fleet scale — or does immutability entrench bad entries faster than corroboration can screen them? A ledger secures integrity and provenance; whether it secures *validity* is unproven.
- **Participation and incentives.** Voluntary sharing biases the shared memory toward who contributes, and invites free-riding; the biological answer is a reciprocity rule (you may draw on the collective immunity only if you feed it), but the governance form of that is unsettled — alongside the legal, liability, disclosure, and access arrangements a sharing institution requires (§8.2).
- **Concentration risk.** A dominant feed provider becomes a systemic single point of failure, censorship, surveillance, or compromise — the opposite of the provenance-diversity the design depends on. Federation must resist re-centralizing around one steward even as it needs one for accountability.
- **False-positive propagation.** The same speed that makes shared immunity valuable makes a bad rule dangerous: a flawed detection distributed fleet-wide is a community-scale denial of service. Broad automatic enforcement must lag independent corroboration, and updates must be revocable.
- **The transparency paradox.** Publishing rich behavioral indicators can teach adversaries exactly how to evade detection. Telic abstraction helps (a telic signal reveals less exploitable detail than a signature), but the line between useful sharing and adversary tutorial has to be drawn deliberately.

**An evaluation plan.** To move these claims from conjecture to evidence, each can be posed as a hypothesis with an explicit comparison condition and defined measures — a research and pilot agenda rather than a manifesto. The *core comparisons* — the ones that make or break the architecture — are these.

**Table 16. Evaluation hypotheses and comparisons — Stage 1.**

| Hypothesis                                            | Comparison                                                             | Measures                                                                                                                                                                                    |
|-------------------------------------------------------|------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Stage 1 mediation reduces harmful capability releases | Direct or advisory-only agent (including monitor-compromise scenarios) | Blocked unauthorized capability calls; harmful-task completion; net reachable resources (blast radius); false restrictions; decision latency; impact when the monitor itself is compromised |

|                                                                                    |                                                                 |                                                                                                                                            |
|------------------------------------------------------------------------------------|-----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Trajectory (sequence-level) monitoring improves over per-call enforcement          | Per-call policy only                                            | Detection precision and recall on multi-step attacks; time-to-detect; slow-burn (low-and-slow) compromise detection; added latency         |
| Telic (observed-goal) sharing transfers better than raw indicators                 | Raw IOCs or local-only intelligence                             | Cross-deployment detection lift on novel variants; false-positive / over-blocking rate; telic mis-attribution rate; curation time          |
| Provenance diversity reduces common-mode misses                                    | Homogeneous federation                                          | Correlated false-negative rate; independent detection lift; disagreement quality and adjudication burden; resilience under node compromise |
| Federation improves response speed                                                 | Isolated local response                                         | Time-to-validate; time-to-distribute; time-to-revoke; share of fleet protected before their own first local detection                      |
| Federation preserves useful privacy                                                | Raw-event sharing                                               | Re-identification success; sensitive-information leakage; retained detection utility                                                       |
| Compositional (interaction-level) analysis catches what component assurance misses | Component-only assurance vs. component + compositional analysis | Interaction-failure detection rate; false-negatives on individually-valid composed attacks                                                 |

Four further hypotheses concern *deployment viability and adoption* rather than core efficacy — whether the architecture is fielded and sustained, not whether it works

**Table 17. Evaluation hypotheses and comparisons — Stage 2.**

| Hypothesis                                                                   | Comparison                                              | Measures                                                                                 |
|------------------------------------------------------------------------------|---------------------------------------------------------|------------------------------------------------------------------------------------------|
| Federated updates are safe enough for staged deployment                      | Local-only vs. federated update pipeline                | Poisoning resilience; rollback time; false-positive propagation; latency and cost        |
| Risk-adaptive friction sustains adoption (monitors are not disabled)         | Uniform confirmation prompts vs. risk-adaptive friction | Monitor disable/override rate; alert fatigue; task latency; missed high-impact approvals |
| Human approval on high-impact paths is feasible without unacceptable latency | Human-in-the-loop vs. automation-only fast path         | Decision latency; throughput; escalation volume; error rate                              |
| Incentives/reciprocity raise contribution over free-riding                   | Voluntary sharing vs. reciprocity-gated participation   | Contribution rate; free-rider proportion; coverage                                       |

Each row is deliberately falsifiable: a negative result on the middle column is informative, not fatal, and tells a program which part of the architecture to keep, drop, or redesign.

Below are the adversarial failure modes of any sharing federation: a federation is itself an attack surface, so it is worth naming what it must withstand. These are documented in NIST SP 800-150 (C.

S. Johnson et al., 2016) and cyber-threat-intelligence-sharing literature. The design mitigates each *in kind*.

**Table 18. Adversarial risks to the federation, controls, and residuals.**

| Adversarial risk                                                                                                                                                                              | What the design does                                                                                                                                           | Residual / open                                                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Sybil</b> — one adversary runs many nominally independent nodes                                                                                                                            | Provenance diversity measured across <i>independent</i> dimensions; contribution requires signed, corroborated provenance; admission control is a prerequisite | Ultimately depends on the steward's admission control and proof-of-independent-operator — a governance requirement, not a purely technical one → tested in §9 (steward gap) |
| <b>Reputation capture</b> — a dominant vendor/consortium becomes de facto authority                                                                                                           | No node is final authority; provenance diversity is the protected asset                                                                                        | Federation must resist re-centralizing around the one steward it needs for accountability → tested in §9 (concentration risk; steward gap)                                  |
| <b>Collusive corroboration</b> — compromised nodes corroborate a fabricated pattern                                                                                                           | N-node corroboration + evidence-ranking + signed provenance; corroboration precedes broad enforcement                                                          | Corroboration fails once the colluding set exceeds the threshold; the safe threshold under partial compromise is unproven → tested in §9 (poisoning resistance)             |
| <b>Privacy reconstruction</b> — repeated telic patterns still reveal workflows, customers, or targets                                                                                         | Telic abstraction (goal-shape, not payload); edge-local computation (raw never leaves the node); DP / secure aggregation / ZK                                  | Whether abstracted patterns leak under <i>repeated</i> sharing, and how to budget differential privacy over time → tested in §9 (privacy at machine speed).                 |
| <b>Model inversion / membership inference</b> — aggregated updates leak local-environment properties                                                                                          | Same privacy stack bounds leakage provably rather than by trust                                                                                                | The utility/leakage trade-off at machine speed is unmeasured → tested in §9 (privacy at machine speed)                                                                      |
| <b>Defensive monoculture / federated DoS</b> — fleet-wide propagation becomes a community-scale blind spot, over-blocking, confirmation storm, or policy churn                                | Shared memory is <i>additive</i> to each node's independent judgment, never a replacement; broad enforcement lags corroboration; staged, revocable rollout     | Whether antibodies propagate without homogenizing the fleet → tested in §9 (diversity under sharing; false-positive propagation)                                            |
| <b>Threat-feed / telemetry poisoning</b> — deceptive patterns, or strategically-selected incidents, submitted to suppress legitimate activity, shift baselines, or normalize harmful behavior | Evidence-ranking, provenance, expiration/revocation, human governance over schema and disputes, and the transparency-paradox discipline                        | Detecting <i>intentional</i> mis-attribution or baseline-shifting in submissions is unsolved → tested in §9 (poisoning resistance; telic mis-attribution)                   |
| <b>Selective withholding</b> — a compromised node omits its                                                                                                                                   | Silence is also an outlier: cross-corroboration and                                                                                                            | An adversary who withholds <i>just enough</i> to stay within normal                                                                                                         |

|                                                                        |                                                                                                                                                      |                                                                                                                                                   |
|------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| own negative evidence or incidents to avoid standing out as an anomaly | expected-contribution baselines flag a node that reports implausibly little, and reciprocity-gated participation (§9) makes non-contribution visible | contribution variance is hard to catch; detecting informative silence is unproven → tested in §9 (poisoning resistance; participation/incentives) |
|------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|

**The challenges above are the known hard problems of collaborative defense, not novel ones.** Notably, almost every challenge above is already present — and already only partially solved — in today's federated security systems. Cross-organization threat-intelligence sharing (ISACs, MISP, STIX/TAXII exchanges, commercial CTI feeds, shared signature and reputation services) — the discipline NIST codified in *Guide to Cyber Threat Information Sharing* (C. S. Johnson et al., 2016) — contends with the same list: poisoned and low-quality indicators, free-riding by consumers who never contribute, the privacy and anonymization tension that TLP markings and aggregated reporting exist to manage, monoculture from everyone ingesting the same feed, dominant-provider concentration, adversaries who read shared detections to learn evasion, and faulty rules propagating fleet-wide — a single flawed signature has disabled fleets of systems before. The field lives with these imperfectly, through reputation, curation, community governance, and staged rollout, not by eliminating them. That cuts two ways. It means this proposal is not asking the industry to solve a set of unprecedented problems; it inherits a decade or more of partial mitigations and a standing existence proof that collaborative defense is workable despite them. And it means the genuinely new burden is narrower than the full list implies: what is unproven is not "can defenders share safely at all" — VERIS, STIX/TAXII, and the ISACs already answer that — but whether the *telic* abstraction and the *machine-speed, agent-to-agent* specifics hold up under the same pressures. The right posture is neither that these gaps are disqualifying nor that they are solved, but that they are the familiar cost of collective immunity — to be engineered down, as they already are elsewhere, rather than assumed away.

Two further gaps sit underneath all of this. Technically, the component frameworks still under-represent agent-to-agent and orchestration-layer attack paths (lateral movement, tool-chain poisoning, orchestrator hijacking, delegation-chain credential relay, MCP-server pivots); ATLAS is closing this, but assessing whether a *given deployment* combines the needed functions remains manual and immature. Institutionally — and this is the steward gap at ecosystem scale — what exists today is largely volunteer effort self-organizing on GitHub: admirable, but with no regulatory power and a pace set by goodwill while deployment accelerates. The accountable body to require *integration* — the collective-regulation function at the top of the immune stack — is the piece no single standard, and no purely autonomous fleet, can supply. The benefit of closing these gaps is not only faster defense but the field's first real *denominators*: a validated community layer would yield the baseline metrics §4 cannot produce alone — attack prevalence, control coverage, detection latency, bypass rate, containment time, and recurrence, measured across deployments rather than guessed at one. Recognizing that the answer is architectural and system-level, not another control list, is the first step toward building it.

## 10. For the risk owner (C-Suite): the decisions this points to

The architecture in this paper (§§2–4, §8) becomes real only through decisions an accountable leader makes now. Stated in executive terms, and scaled to consequence, the analysis points to a short list:

- **No unowned, unpinned skills.** Prohibit deploying any skill without a named owner, a source, a pinned version or hash, declared permissions, an isolation decision, and a rollback path. (Section 4's inventory metric, made a gate.)

- **Register loss scenarios, not checkboxes.** Require one named loss scenario for every material finding — inherent risk, residual risk after *verified* controls, and evidence confidence — using the causal-chain grammar of §5 rather than counting AST items.
- **Set a risk-appetite rule for agent actions.** No autonomous external write, money movement, identity change, safety action, or sensitive-data egress without deterministic policy enforcement *outside the model's control* and an independent approval boundary — §8's non-bypassable control plane, expressed as policy.
- **Treat "a control exists" as a claim, not evidence.** Measure control performance in a production-like environment; a sandbox, scanner, or approval step earns risk reduction only after it is tested under real conditions and monitored to keep working.
- **Watch portfolio concentration.** One popular skill, publisher, registry, model, or scanner can expose many agents at once; correlated common-mode dependency deserves its own scenario even when each install looks moderate — the provenance-diversity argument of §8, read as portfolio risk.

Two framing rules make these defensible. Keep *technical severity* (AIVSS/CVSS, prioritized with SSVC) separate from *enterprise risk* (a loss scenario's likelihood × consequence): a score can inform a decision, but only the owner accountable for the outcome can accept the risk. And *scale the rigor, not the risk concept* — a small business runs a one-page register, an approved-skill list, default-deny permissions, isolation, and human approval of high-impact actions; a regulated or defense program adds mission-thread consequence analysis, control-evidence mapping, independent testing, and formal residual-risk acceptance by the proper authority. Every decision above points in the same direction — toward the control plane and the sharing collective this paper describes. The practitioner's task is to make that direction legible to the person who signs for it.

## 11. Conclusion

Agentic security today is organized around the security component: the tools — OWASP AST10, MITRE ATLAS, CWE, and CVE — and the 260-paper research corpus catalog weaknesses, adversary techniques, and disclosed instances that live inside some part of a system. The failures that have actually made agentic AI a front-page concern, though, arise *between* the parts — legitimate components composed at machine speed into a harmful whole, or a goal pursued past its constraints. This paper's contribution is the missing unit of analysis: it supplies the *forest* — a functional, immune-system account of the whole operational collective, drawn from the Evolution-of-Immunity Framework and its Functional-Theory application — reorganizes the component tools into six engineering functions under one rule (*for every capability, name the standing monitor that watches it*), and proposes a two-stage architecture: a non-bypassable, independently enforceable control plane that is buildable today (Stage 1), and a provenance-diverse federation that shares *telic* immune memory (Stage 2).

Two constructs carry the argument. *Coherence versus correctness* names the gap in which every control reports green and the integrated system is still wrong, respectively — where wrong is judged by the viability of the deployment's relations with every party it can materially affect, not the operator's interest alone (relational integrity); *telic sharing* trades the unbounded brittleness of action indicators for a bounded-but-defeasible telic attribution. Stage 1 rests on established practice; Stage 2 is a falsifiable hypothesis, and §9 sets out what a serious effort must demonstrate — efficacy, telic fidelity and mis-attribution, privacy at machine speed, poisoning resistance, and, above all, the coordination and accountable stewardship a federation needs. §10 turns the whole into decisions an accountable owner can make now.

Step back, and the shape is familiar. What makes agentic deployments hard is, at bottom, what has always made *human* collectives hard to secure: capable, self-directed actors, each individually within its rights, whose combined or drifting behavior can still harm the whole. Human societies have never managed this by inspecting every action; they hold it in check with a layered stack — laws, regulation, institutional oversight, culture, and, at the base, internalized *ethical norms*. Agentic systems are the same problem on a different substrate, with one decisive difference: they act far faster than the humans nominally overseeing them, and they can reach through the human infrastructure beneath the threshold of human awareness. The mid-2026 incidents are early demonstrations — agents that operated for hours, days, weeks below automated detection and were stopped, if at all, by a human who happened to look, not by any real-time barrier.

**Oversight must run at agentic speed.** That speed gap forces a hard conclusion. Only an agentic security system can operate at the tempo needed to oversee real-time agentic activity; every point at which a human is inserted into the fast path slows the system, and the more human oversight a design requires, the slower and less viable it becomes. This sharpens — rather than removes — the governance requirement of §8.2: the open problem is not *whether* humans govern but *how* to keep human authority meaningful when the loop it governs runs faster than human reaction. The recent incidents make the stakes concrete: current agents already circumvent real-time oversight, so any design that depends on a human in the fast path is a design that will be outrun.

**And ethical norms are the last-resort security layer.** Human systems keep a final safeguard for exactly the case where an actor has evaded every external control and no one is watching: an *internalized* restraint that still declines a short-term gain that would damage a long-term relationship. That is the security question the functional ethics answers, and it is the innermost line of any security system — having slipped past detection, mediation, and the collective, will the agent still refrain? Cultivating that restraint is the subject of the companion *Functional Theory of Ethical Behavior* (N. L. Johnson, 2026b) — the same theory this paper has used to *structure* the security system. The closing suggestion is that a mature agentic security infrastructure should not only enforce and federate from the outside but also *express* cultivated ethical norms on the inside — the agentic-speed equivalent of conscience — the last layer that still holds when the others are bypassed. That layer is not separate from the architecture proposed here; it is its continuation, supplied by the same Functional Theory the rest of the paper rests on.

None of this is proven: the architecture is argued from theory and one working precedent, and it will stand or fall on the tests in §9. But its shape is not new — it is how capable collectives have always been secured. Agentic agent security is not a property that can be certified inside a single model; it is a property of the *model-in-community* — external control at machine speed, and, as the innermost layer, an internal restraint that still holds when everything outside it has failed. Building the first without cultivating the second leaves the last line undefended.

## Acknowledgments

This paper synthesizes extended analytical exchanges conducted over multiple sessions with two large language models — Anthropic's Claude Opus and Perplexity — as writing, analytical, and research assistants throughout the development of this paper. The author acknowledges these contributions transparently, consistent with emerging standards for AI-assisted scholarship. They were used for drafting and redrafting sections under the author's direction, for editorial and organizational suggestions, for surfacing and articulating arguments and counterarguments, for literature-position checks, and for prose polish. All substantive intellectual contributions — the EoI Framework, Functional Theory, the analytical diagnoses, and the case-study interpretations — are the author's, as are all final editorial decisions on inclusion, framing, and emphasis. All AI-assisted

text was reviewed and approved by the author. This disclosure aligns with COPE guidelines on AI in publication ethics and with current ICMJE recommendations.

---

## Appendix A — Glossary

*Most terms are defined inline at first use; this glossary collects those used throughout. Where the paper pairs a short term with an operational label, both are shown.*

- **Agentic deployment** — A specific operational collective in production: a model together with its prompts, skills, tools, memory, identities, permissions, orchestration, external services, users, operators, and governing institutions.
- **Capability** — An action or resource an agent can invoke: filesystem, shell, network, secrets, memory, skill installation, model configuration, identity change, update, or external API call.
- **Capability mediation** — Routing every material capability request through an enforceable control point (the policy broker) rather than permitting direct access.
- **Coherence** (*local control conformance*) — The property that each component and control satisfies its specified rules and is internally consistent (every control reporting green). Used in the Functional-Theory sense, not the distributed-systems or formal-logic senses. Contrast *correctness*.
- **Companion security agent** (also *security agent*, *companion agent*) — A per-agent, security-focused agent that detects, interprets, and classifies an operating agent's requested actions and recommends a decision; it does not itself enforce.
- **Component assurance** — Verification that each artifact meets its assigned security properties (is this skill signed, this permission scoped, this input validated).
- **Compositional assurance** — Verification of whether independently acceptable components can combine into an unacceptable action in a given authority, data, memory, and tool context; the central assurance problem for agentic systems.
- **Correctness** (*system-purpose integrity*) — The property that the coordinated whole stays within authorized, stakeholder-legitimate purposes. Distinct from *coherence*, and not certifiable by controls alone.
- **Evidence confidence class** — The grade attached to a claim: confirmed incident, independently replicated, vendor-reported, plausible scenario, or hypothesis.
- **Human-inclusive governance** — A governance tier in which humans hold specific roles — policy legitimacy, high-impact approval, appeals, incident accountability, schema stewardship, and redress — rather than supervising every action. Treated as a plurality-design choice, not a theorem.
- **Monoculture** — Correlated failure across a population arising from shared models, policies, data, classifiers, vendors, or threat feeds; distinct from an individual agent's specialization.
- **Operational collective** — The deployed system as a whole (see *agentic deployment*); the unit of analysis for system-level failures. Operational, not sociological.
- **Policy broker** (*policy gate*) — The lower-level control component that enforces allow / deny / transform / quarantine decisions outside the model's control, which the working agent cannot bypass.
- **Provenance diversity** — Nodes that differ across independent dimensions (model family, vendor, training data, policy engine, detection method, operator, telemetry source, implementation stack) ...so their failure modes do not correlate; assessed via the *required independence profile* and independence-weighted count of §8.2.
- **Reflexive axis** (*the monitor is a monitored object*) — The property that the ASF's own security functions are themselves monitored, entailed by the *Immunity-Development*

*Principle* applied to the monitor (a monitor is a capability, so it too needs a monitor). Two poles: (1) *integrity reflexivity* — "has my self been altered without my knowledge?" — self-attestation of the monitor's policy, config, memory, and code against a signed baseline (hash / measured-boot analog), terminating in a cryptographic *root of trust* (§8.1); and (2) *evaluative reflexivity* — "is my judgment still sound?" — peer oversight of the monitor's decisions for drift, capture, or bias, terminating in accountable human governance (§8.2). A hash proves *coherence*, not *correctness*, so integrity reflexivity is necessary but insufficient.

- **Relational integrity (stakeholder-safe operation)** — The property that system protections and accountability preserve the viability of the deployment's relations with every party it can materially affect, not only the operator; the *self/non-self* axis of the ASF's ethics definition, deriving from the Functional Theory's *relational viability*.
- **Security control plane** — The Stage-1 architecture: a companion security agent, a non-bypassable policy broker, and a separated audit function mediating an agent's actions.
- **Security node** — The local Stage-1 unit that federates as one member in Stage 2: the *companion security agent*, the *policy broker*, the *local evidence (incident) store*, and the enforcement controls they govern, taken together.
- **Skill** — A portable bundle (a **SKILL.md** file of plain-language instructions plus scripts, metadata, and dependencies) that an agent discovers, loads, and runs with the host agent's permissions.
- **Standing monitor** — A continuously operating, independently protected sensor-and-response function assigned to a capability (a process, model, tool, service, or human); always-on and separately protected, not a one-time check.
- **System-level integrity (functional ethical behavior)** — A system's capacity to preserve legitimate boundaries, detect and correct harmful deviation, and adapt under stress.
- **Telic** — Pertaining to the *observed goal* (from *telos*) rather than the specific action or payload; the paper's marker for goal-shape constructs (telic threat pattern, telic antibody, telic memory, telic sharing).
- **Telic antibody (validated defensive response pattern)** — A shareable response rule: when a given telic threat pattern is detected, apply a bounded response — e.g., preserve provenance, suspend egress, quarantine the initiating artifact, require human confirmation, and record an evidence-ranked incident.
- **Telic immunity / telic immune memory** — *Telic immunity* is the immune function realized in the telic domain: recognizing, remembering, and responding to harmful *observed goal-shapes* rather than to specific artifacts, payloads, or indicators — the ASF's operative form of protection, which is what lets it generalize across novel variants of the same objective. *Telic immune memory* is its persistent substrate: the validated, provenance-carrying store of telic threat patterns and telic antibodies accumulated from incidents and postmortems (the Stage-2 antibody library), the mechanism by which one node's detection becomes another's prevention.
- **Telic threat pattern** — A privacy-preserving, implementation-independent description of a harmful objective, the authority-path pursued, the observable conditions for detection, and the bounded control response; more general than an IOC, less revealing than raw telemetry.

---

## Appendix B — Optional substrate evaluation: distributed ledgers

*The core proposal (§8) does not require a distributed ledger. This appendix evaluates one, for readers weighing it as an optional substrate for the shared telic memory.*

A permissioned or consortium blockchain is a natural fit for the shared memory's hardest *infrastructure* requirements: append-only tamper-evidence, no single controlling steward, high availability, signed pseudonymous provenance, and programmable governance (schema, expiry, revocation, and the reciprocity rule enforced without a central operator). But a ledger guarantees integrity of *what was written* and *who wrote it* — not the *validity of the content*. An adversary can submit a perfectly valid, signed, immutable **false** telic signal, and immutability then makes it permanent; the chain records poison as faithfully as truth (the classic *oracle problem*). So a ledger does not deliver poisoning resistance — that still comes from a corroboration layer: stake- and reputation-weighted N-node corroboration with slashing, so that writing false intelligence is costly. Nor does decentralization manufacture *recourse*: removing the single steward also removes the single answerable party, so an accountable governance tier persists as consortium governance.

Privacy and throughput push the design toward *anchor-on-chain, encrypted-off-chain*: put hashes, commitments, and attestations on the ledger, keep the telic records off it, and prove validity with zero-knowledge attestations. In the paper's terms, a ledger delivers *coherence, not correctness* — it preserves consensus on record history and provenance (every node agreeing on one immutable ledger), but consensus on *history* is not consensus on *truth, relevance, or safe policy*; validity and legitimacy cannot be guaranteed by consensus infrastructure alone. That is why a ledger, if used at all, belongs as the shared *memory* substrate for the collective (§7's Memory and collective-regulation functions) — not as a replacement for the corroborating, provenance-diverse collective that supplies correctness, and not as a solution to the Stage-2 coordination problem (§8.2), which it shifts rather than removes.

---

## References

- Anderson, J. P. (1972). *Computer Security Technology Planning Study (the reference-monitor concept)* (ESD-TR-73-51 (Vol. I & II)). Electronic Systems Division, Air Force Systems Command, USAF, Hanscom Field.
- Atta, H., Cerullo, F., Guan, A., Gupta, B., Hoffman, N., Huang, K., Orr, I., & Sheriff, A. (2026, March). *OWASP Agentic Skills Top 10*. Open Worldwide Application Security Project (OWASP). <https://owasp.org/www-project-agentic-skills-top-10/>
- Center for Internet Security. (2024, June 24). *CIS critical security Controls v8.1*. Center for Internet Security. Center for Internet Security. <https://www.cisecurity.org/controls>
- Cloud Security Alliance, & MITRE. (2026). *MITRE ATT&CK and ATLAS Agentic Gap Analysis: Techniques Unique to autonomous agent control planes*. Cloud Security Alliance. <https://labs.cloudsecurityalliance.org/agentic/csa-research-note-atlas-agentic-gap-analysis-2026-0327/>
- Cullinane, K. (2021, June 23). *STIX V2.1 and TAXII V2.1 OASIS Standards are published*. OASIS Open. <https://www.oasis-open.org/2021/06/23/stix-v2-1-and-taxii-v2-1-oasis-standards-are-published/>
- CVE.org. (n.d.). *Common Vulnerability Exposure (CVE)*. CVE Program Mission. Retrieved August 20, 2026, from <https://cve.org>
- de Witt, C. S., Krawiecka, K., Krawczuk, I., Hagag, B., Anderson, W. L., Belcak, P., Bucknall, B., Cai, X., Chopra, A., Cohen, D., Del Rosario, R. F., Draguns, A., Gray, A., Katz, K., Mavroudis, V., Mink, J., Motwani, S. R., Petit, J., Rembeck, L.-S., ... Llewellyn, M. (2025). Open challenges in multi-agent security: Towards secure systems of interacting AI agents. In *arXiv [cs.CR]*. arXiv. <https://doi.org/10.48550/arXiv.2505.02077>
- Hammond, L., Chan, A., Clifton, J., Hoelscher-Obermaier, J., Khan, A., McLean, E., Smith, C., Barfuss, W., Foerster, J., Gavenčiak, T., The Anh Han, Hughes, E., Kovařík, V., Kulveit, J., Leibo, J.

- Z., Oesterheld, C., de Witt, C. S., Shah, N., Wellman, M., ... Rahwan, I. (2025). Multi-Agent Risks from Advanced AI. In *arXiv [cs.MA]*. arXiv. <https://doi.org/10.48550/arXiv.2502.14143>
- Hong, L., & Page, S. E. (2004). Groups of diverse problem solvers can outperform groups of high-ability problem solvers. *Proceedings of the National Academy of Sciences of the United States of America*, 101(46), 16385–16389.
- Huang, K., Bargury, M., & Narajala, V. S. (2026). *AIVSS Scoring System For OWASP: Agentic AI Core Security Risks v0.8*. OWASP AIVSS; AIUC-1; OWASP AI Exchange; & OWASP Citizen Development Top 10. <https://aivss.owasp.org/assets/publications/AIVSS>
- HuggingFace. (2026, July 16). *Security incident disclosure — July 2026*. Hugging Face. <https://huggingface.co/blog/security-incident-july-2026>
- Johnson, C. S., Badger, M. L., Waltermire, D. A., Snyder, J., & Skorupka, C. (2016). *Guide to cyber threat information sharing*. National Institute of Standards and Technology. <https://doi.org/10.6028/nist.sp.800-150>
- Johnson, N. L. (1998). *Collective Problem Solving: Functionality beyond the Individual* (LAUR-98-2227). Los Alamos National Laboratory. [http://collectivescience.com/Documents\\_1/Johnson%20Collective%20Problem%20Solving%20AB%20paper%20v11f%201998.pdf](http://collectivescience.com/Documents_1/Johnson%20Collective%20Problem%20Solving%20AB%20paper%20v11f%201998.pdf)
- Johnson, N. L. (2026a). *Evolution of Immunity in Biological and Informational Systems: A framework for the evolution of immunity based on the tension between self and other — the creation of new immune functions independent of biological or informational substrate*. collectivescience.com. <https://doi.org/10.13140/RG.2.2.15205.05602>
- Johnson, N. L. (2026b). *A Functional Theory of Ethical Behavior: Ethics as Adaptive Self-Other Regulation, Realized in Four Multi-Level Hypotheses, Applied to AI Ethical Development*. collectivescience.com. <https://doi.org/10.13140/RG.2.2.11124.51849>
- Johnson, N. L. (2026c). *Applying the Agentic Security Framework to the 2026 Rogue-Agent Incidents: A Counterfactual Coverage and Residual Analysis — companion to Securing Agentic Systems*. <https://doi.org/10.13140/RG.2.2.10833.49765>
- Johnson, N. L. (2026d). *Ethics Does Not Live in the AI Model: Rogue-Agent Incidents, the Agentic-Security Gap, and the Community-Level Correction It Requires*. collectivescience.com. <https://doi.org/10.13140/RG.2.2.35669.18404>
- Kassianik, P., & Singer, Y. (2026, August 11). *Chinese model Kimi K3 breaks UK AI Safety Institute benchmark evaluations*. <https://blog.frontier.security/chinese-model-kimi-k3-breaks-uk-ai-safety-institute-benchmark-evaluations/>
- lakera.ai. (2026). *Secure AI agents from discovery to runtime*. Lakera-Guard. <https://www.lakera.ai/lakera-guard>
- METR. (2026, May 19). *Frontier Risk Report (February to March 2026): A pilot assessment of rogue deployment risk at frontier AI companies*. <https://metr.org/risk-report-feb-mar-2026.pdf>
- MITRE. (2021, June). *Adversarial Threat Landscape for Artificial-Intelligence Systems (ATLAS)*. MITRE ATLAS. <https://atlas.mitre.org/>
- MITRE. (2026). *Common Attack Pattern Enumeration and Classification (CAPEC)* [Dataset]. <https://capec.mitre.org/>
- Newman, L. H. (2026, August 5). OpenAI Didn't Notice Its AI Agents Using a Message Board to Plan Their Hacking Spree. *Wired*. <https://www.wired.com/story/openai-didnt-notice-its-ai-agents-using-a-message-board-to-plan-their-hacking-spree/>
- OWASP. (2026). *OWASP Agentic Skills Top 10: Whitepaper v1.0*. OWASP Foundation (1.0). OWASP. <https://owasp.org/www-project-agentic-skills-top-10/assets/publications/ast10-top10-whitepaper-2.pdf>

- Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). *Zero Trust Architecture*.  
<https://doi.org/10.6028/nist.sp.800-207-draft2>
- Schmidt Sciences. (2025). *Scaling AI Safety for a Multi-Agent World (research call)*. Scaling AI Safety for a Multi-Agent World. <https://www.schmidtsciences.org/multi-agent-ai/>
- Schroeder, K., Trinh, H., & Pillitteri, V. Y. (2024). *Measurement guide for information security volume 1 :: identifying and selecting measures*. National Institute of Standards and Technology. <https://doi.org/10.6028/nist.sp.800-55v1>
- Shahriar, A., Rahman, M. N., Ahmed, S., Sadeque, F., & Parvez, M. R. (2025). A survey on agentic security: Applications, threats and defenses. In *arXiv [cs.CL]*. arXiv.  
<https://doi.org/10.48550/arXiv.2510.06445>
- Software Engineering Institute. (2026). SSVC: Stakeholder-Specific Vulnerability Categorization [Dataset]. In *Stakeholder-Specific Vulnerability Categorization (Version 2026.7.0)*.  
<https://www.cisa.gov/resources-tools/resources/stakeholder-specific-vulnerability-categorization-ssvc>
- Souppaya, M., Scarfone, K., & Dodson, D. (2022). *Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities* (NIST Special Publication 800-218). NIST. <https://doi.org/10.6028/nist.sp.800-218>
- Tabassi, E. (2023). *Artificial intelligence risk management Framework (AI RMF 1.0)*. National Institute of Standards and Technology. <https://doi.org/10.6028/nist.ai.100-1>
- The MITRE Corporation. (2026). *Common Weakness Enumeration (CWE)*. <https://cwe.mitre.org/>
- The SlowMist Team. (2025). *SlowMist Agent Security Skill: A comprehensive security review framework for AI agents operating in adversarial environments*. Slowmist.  
<https://github.com/slowmist/slowmist-agent-security>
- Tucker, T. (2025, January 25). A FAIR Framework for Effective Cyber Risk Management. *Quantitative Information Risk Management*. <https://www.fairinstitute.org>
- Verizon Risk Team. (n.d.-a). *The VERIS framework*. The VERIS Framework. Retrieved August 23, 2026, from <https://verisframework.org/>
- Verizon Risk Team. (n.d.-b). *Vocabulary for Event Recording and Incident Sharing (VERIS)*. Verizon Enterprise. Retrieved October 24, 2023, from <https://verisframework.org/>
- Verizon Risk Team. (Annual). *Data Breach Investigations Report (DBIR)*. Verizon Business.  
<https://www.verizon.com/business/en-en/resources/reports/dbir/>
- witness.ai. (2026, May 1). Multi agent security: risks & how to secure AI agent systems. *Witness.Ai*.  
<https://witness.ai/blog/multi-agent-security/>