Intrusion Prevention for Agentic Processes: Evidence-Governed Runtime Response and Enforcement

Arslan Brömme
CISSP, CISM, CISA, CAISE
Independent Researcher
arslan.broemme@aviomatik.de

Draft v0.9.1.4 – 15 September 2026

Preprint / working paper. This version is a work in progress and may be updated. Comments are welcome.
DOI: 10.5281/zenodo.22765147

Abstract

Agentic systems can execute tools, retain state, delegate authority, propagate information, and act across organizational boundaries. A security-relevant finding may therefore require intervention, while direct detector-to-block coupling can turn uncertain evidence into harmful enforcement. This paper proposes an *Agentic-Process Intrusion Prevention System (A-IPS)*, a conceptual, evidence-governed intervention model for agentic processes. A-IPS composes bounded findings with a versioned response policy while keeping evidence support, policy-relative evidence sufficiency, admissibility, authorization, execution, scoped effect, and recovery validity distinct. Existing enforcement mechanisms can serve as policy-controlled intervention points rather than being replaced. Agentic-process-specific response targets and dependencies include process branches, delegated authority, information and persistent state, propagation paths, cross-domain relationships, resource budgets, and recovery artifacts. The same governance interface can optionally represent proactive resilience hardening as preventive posture control, but such control remains risk reduction rather than an incident finding or proof of trust. The prevention plane is itself treated as an attack surface. A-IPS is a conceptual architecture and does not claim a new incident-response lifecycle, runtime-enforcement primitive, universally correct policy, empirically validated prevention mechanism, or proof that successful response establishes an uncompromised agent state.

1 Introduction

An agentic process is a bounded task-oriented sequence or graph of observable interactions among AI agents, humans, tools, external systems, information sources, and governance components. Such processes may perform actions with direct operational consequences while retaining context, delegating authority, and propagating information across multiple components. In this paper, an *intrusion finding* denotes a bounded security-relevant deviation from an authorized agentic process. Safety-related outputs, content, or actions are outside the scope of A-IPS unless they also constitute or cause a security-relevant deviation affecting authorization, confidentiality, integrity, availability, propagation, or governed process state. Reliability-related failures are likewise outside scope unless they create such a security-relevant deviation. Such a finding can require more than reporting. At the same time, automatically converting uncertain findings into blocking actions can create a second class of risk because the prevention system itself may interrupt legitimate work, revoke valid credentials, isolate healthy components, quarantine benign information, or terminate an otherwise recoverable process.

This paper continues a line of work that first proposed a black-box evidence architecture for agentic processes and then an evidence model for bounded claims and policy assessment [1, 2]. A subsequent intrusion-detection paper defined an evidence-aware runtime monitoring layer in which findings remain bounded by observation visibility, matching state, source properties, impact, assumptions, and limitations [3]. The present paper asks what should happen after such a finding is produced, and which additional response characteristics arise because the protected object is an agentic process rather than only a conventional host, account, or network flow.

The paper addresses two research questions. *Research Question 1 asks which prevention and response targets, failure modes, and recovery requirements arise from the specific properties of agentic processes. Research Question 2 asks how evidence-supported intrusion findings can be translated into authorized and proportionate runtime interventions without allowing uncertain evidence or compromised monitoring components to exercise unrestricted enforcement.*

The central principle is that *a finding may justify consideration of a response, but it does not itself authorize an intervention.* We use the term *Agentic-Process Intrusion Prevention System (A-IPS)*[1] for the proposed evidence-governed prevention and response model whose intervention target is the agentic process and its relevant dependencies. The paper makes one primary and two supporting conceptual contributions. The primary contribution is a typed intervention model that represents a bounded finding, policy-relative evidence sufficiency decision, admissibility decision, authorization decision, enforcement state, scoped effect claim, and recovery validity claim as distinct record classes and lifecycle states. The first supporting contribution is an integration boundary that permits heterogeneous detectors and existing enforcement mechanisms to be composed without granting detector output implicit authority. The second is an agentic-process instantiation of that boundary for targets and dependencies that include information and memory, process branches, propagation paths, cross-domain authority, resource budgets, and recovery artifacts. Proactive resilience hardening is treated as an optional preventive extension of the same governance interface rather than as a new hardening technique. The paper does not claim a new detector, runtime-enforcement primitive, incident-response lifecycle, checkpoint mechanism, or empirical prevention result.

[1] Pronunciation (IPA): /eɪ | aɪ piː ɛs/.

2 Background and Related Work

Classical intrusion detection and prevention systems combine monitoring with attempts to stop suspected incidents [4]. NIST incident-response guidance places detection, response, recovery, and improvement within broader cybersecurity risk management rather than treating detection as an endpoint [5]. The NIST Generative AI Profile additionally calls for GAI incident-response and recovery plans, remediation procedures, deactivation criteria, and continual improvement, including retraining or decommissioning models that operate outside defined limits [11]. These sources establish a response and governance baseline, but they do not define a technical evidence-to-intervention model for multi-step agentic processes.

Agent-security literature shows why an extension is needed. A recent Systematization of Knowledge (SoK) identifies complex attack and defense surfaces that arise when language models are integrated with tools and non-AI components [12]. Prompt Infection demonstrates self-propagating prompt injection across interconnected agents [13], while cross-layer contagion work models propagation through both agents and shared tools and explicitly includes quarantine and return to service [14]. Persistent state adds another dimension. Planted instructions in agent memory can influence later sessions [15], and recent work on autonomous agent worms shows that persistent workspace and memory state can support temporal re-entry and cross-agent transmission [16]. These results imply that containment may need to address not only an executing agent, but also information, shared tools, memory, and downstream state.

Availability also becomes agent-process-specific. AgentDoS finds resource-exhaustion vulnerabilities caused by resource management in widely used LLM agents [17]. Tool-calling chains can amplify tokens, energy, latency, and cost while still completing an apparently benign task [18]. Adversarial content can also trap LLM-based guardrails in extended reasoning and create large latency amplification [19]. These attacks motivate response policies that consider cumulative resource use and economic impact rather than only unauthorized final actions.

Proactive defense hardening also predates A-IPS. HASTE supports proactive and reactive hardening of LLM defenses and guardrails, while empirical multi-agent work shows that security-prompt hardening can improve resilience while leaving material residual risk [24, 25]. A-IPS does not introduce a hardening method. Its optional proactive resilience hardening extension represents preventive posture actions under the same policy, authority, provenance, and validation boundaries used for other interventions.

Recent work addresses important pieces of containment and runtime enforcement. VIGIL checks execution traces against behavioral specifications [7], C-Trace rejects actions that violate selected runtime compliance predicates [8], AttriGuard blocks tool calls that are not causally supported by user intent under its model [10], and AgentSpec defines customizable trigger-predicate-enforcement rules that intercept proposed agent actions before execution [9]. These systems demonstrate action-local and trajectory-level runtime enforcement. A-IPS does not claim to systematize such enforcement for the first time. Its focus begins when a bounded finding, potentially from heterogeneous detectors, must be composed with intervention sufficiency, delegated authority, process-wide containment, effect verification, and recovery governance.

Li et al. identify a cross-substrate authority gap in which decision-relevant authorization state can reside outside the planner-visible workspace or memory, and show that an execution-time authority check can prevent unsafe model-generated intents from becoming downstream effects [32]. This provides a concrete neighboring mechanism for revalidating authority at a mutation boundary. A-IPS addresses the broader response-governance problem by separately representing bounded finding support, evidence sufficiency, admissibility, delegated authorization, enforcement state, independently observed scoped effect, and recovery validity across heterogeneous intervention targets. Li et al. evaluate authority consistency under trusted host, authority-database, and governance-component assumptions, whereas A-IPS additionally treats the prevention plane itself as an attack surface.

BlueSTAR provides an empirical autonomous-cyber-defense neighbor on live enterprise IT/OT ranges. It grounds high-volume telemetry into indicators, separates deterministic response for precision detections from LLM reasoning over ambiguous cross-cycle context, and evaluates defensive action under partial observability and collateral cost [33]. In BlueSTAR, precision findings can directly trigger deterministic remediation and action outcomes are learned from subsequent telemetry, while attacks directed at the defender itself are outside its threat model. A-IPS addresses a different layer by requiring evidence sufficiency, admissibility, authorization, enforcement state, separately observed scoped effect, and recovery validity to remain distinct even for high-confidence findings, while also treating the prevention plane itself as an attack surface.

ContainmentBench evaluates post-injection containment using trace, propagation, recovery, and utility evidence rather than terminal policy labels alone [20]. A recent architectural containment model proposes explicit constraints and a Propose-Verify-Act-Verify structure [21]. Most directly, AIR integrates incident detection, containment, recovery, and eradication into the agent execution loop and empirically evaluates rule-guided remediation [23]. AIR is therefore the closest incident-response neighbor to A-IPS. AIR primarily provides autonomous in-loop response through agent-executable rules and tools. A-IPS instead models a segregated evidence- and authority-governed response plane whose main unit is a bounded finding plus response candidate, which may originate from heterogeneous detectors and may target agents, credentials, information, memory, tools, branches, or cross-domain relationships. A-IPS makes delegated authorization, enforcement state, effect attribution through a separately controlled observation path, and recovery validity explicit architectural boundaries. This comparison does not imply that AIR lacks verification mechanisms. A-IPS does not claim priority over existing agent incident-response, containment, quarantine, or runtime-enforcement frameworks.

Cross-domain and open multi-agent operation are also active research areas. Recent cross-domain multi-agent research identifies security challenges that arise when agents collaborate across organizations and no single trust assumption governs the interaction [27]. A recent SoK characterizes multi-agent security in terms of interaction paths that move information, state, decisions, and authority across principal boundaries, with trust boundary and recovery as explicit defense dimensions [28]. The A2A protocol standardizes communication among independent and potentially opaque agent systems and places authentication and authorization responsibilities on implementations, while leaving the scope, representation, validity, and revocation semantics of in-task authorization outside the core protocol [29]. A2ABreak reports protocol-level vulnerabilities involving cross-client context injection, multi-hop identity loss, and rogue capability claims under its analysis model [30]. Ongoing IETF work separately frames cross-organizational delegation as a problem of verifiable, attenuated authority across trust anchors and explicitly remains work in progress [31]. A-IPS does not claim these trust-boundary or delegation problems as new. It asks how a prevention layer should preserve their uncertainty, authority domains, evidence, and recovery consequences when deciding and verifying response.

Checkpoint and rollback mechanisms likewise predate the

present model and are becoming part of agent runtime research. DeltaBox and Crab study efficient checkpoint/restore of agent sandbox state for branching, fault tolerance, and recovery [34, 35]. Security work shows, however, that restoring an agent checkpoint is not equivalent to restoring a safe execution history. ACRFence identifies semantic rollback risks such as action replay and authority resurrection [36], while recent work shows that secure continuation can fail when restored internal state is inconsistent with external dependencies or already executed side effects [37]. A-IPS therefore treats checkpointing and cloning as possible recovery mechanisms, not as proof that the recovered process is trustworthy.

Security design principles such as fail-safe defaults, complete mediation, separation of privilege, and least privilege remain relevant to prevention architectures [38]. The confused-deputy problem further illustrates why possession of technical authority should not be confused with authorization to exercise it for a particular principal or purpose [39]. For A-IPS, the same distinction applies to response components that possess powerful isolation, revocation, remediation, or termination capabilities.

3 Agentic-Process-Specific Response Extension

A-IPS is conceived as an overlay on established incident-response and prevention mechanisms, not as their replacement. The concerns below are not individually unique to agentic systems. Their agentic specificity lies in how language-mediated delegation, persistent context, tool use, and non-deterministic execution redistribute observability, authority, propagation, and recovery boundaries. Table 1 summarizes these manifestations.

Information quarantine is therefore broader than file blocking. A suspicious document, message, webpage, tool output, retrieval item, or memory entry may be the direct carrier of adversarial instructions, while summaries, copied text, cached outputs, generated artifacts, or other known derivatives may continue to influence the process after the original source is removed. Where provenance and visibility permit, containment should therefore combine *source quarantine* with *derivative-state quarantine* for known downstream state that can re-enter an agent's decision context. This is a visibility-bounded control, not a guarantee that every derivative has been discovered.

Agent-generated incident reports and diagnostic interactions can add evidence but should not collapse source trust. A self-report from a potentially affected agent, a peer report, or a supervisor-agent assessment is a claim with identity, timing, context, and limitations. A-IPS may respond by increasing observation, asking structured questions, running controlled tests, replaying a task in a sandbox, or comparing behavior against an independent evaluator. The affected agent should not be allowed to certify its own recovery when the relevant threat could influence that certification.

3.1 Controlled Intent Elicitation and Cross-Source Consistency

Structured diagnostic interaction can ask an affected or suspicious agent about its declared objective, rationale, interpretation of authority, or provenance of a decision. Such questioning produces a directly attributable declaration rather than requiring those claims to be inferred only from external behavior. The declaration remains a source-attributed evidence claim, not ground truth about latent or internal intent. A compromised agent may conceal an objective, rationalize retrospectively, or operate with incomplete context. A-IPS therefore does not equate a declared objective with actual intent and does not treat denial of malicious purpose as exculpatory evidence by itself.

A compact declaration claim can be represented as

$$\begin{aligned}\mathsf{DeclarationClaim} = (&\text{agent}, \text{elicitation}, \text{statement},\\ &\text{time/sequence}, \text{context reference},\\ &\text{policy version}, \text{evidence reference}).\end{aligned}$$

The elicitation context should be governed and recorded. Where feasible, questions should use a controlled path distinct from the suspected instruction channel and may include semantically varied, counterfactual, or action-specific challenges. These measures can increase diagnostic value but cannot guarantee truthful disclosure.

The declaration becomes more useful when tested against source-separated process and conventional security evidence. Agentic-process evidence can include tool calls, messages, delegated authority, memory or retrieval state, and workflow events. Conventional IDPS, endpoint, network, identity, API-gateway, or data-loss-prevention telemetry can observe communication, credential use, execution, or data movement. Let d denote a declaration claim and $\mathcal{O}_s$ relevant source-separated observations.

$$\text{consistency}(d, \mathcal{O}_s) \in \{\text{consistent}, \text{contradicted}, \text{unresolved}\}.$$

The status is a bounded evidence relation, not a prescribed classifier. Implementations may derive it deterministically or probabilistically, while any confidence remains source-specific evidence rather than A-IPS ground truth.

Here, *contradicted* denotes a contradiction between the declaration and observed evidence, not a finding of deception. Such a contradiction can weaken the declaration and support an additional bounded finding, for example when an agent denies external communication while a separately controlled gateway records an authenticated outbound action. It does not by itself establish deliberate deception, malicious intent, or a latent objective. Source separation must also be assessed rather than assumed.

Corrective response can also go beyond containment. Depending on policy and deployment architecture, remediation may include renewed policy guidance, context or memory repair, restoration from a policy-accepted checkpoint, configuration changes, retraining, fine-tuning, or model replacement. Such measures are not automatically safer than isolation, and fine-tuning can change safety behavior in unintended ways [22]. Model-changing remediation should therefore be separately authorized and independently validated rather than treated as evidence that the triggering deviation has been resolved.

3.2 Optional Proactive Resilience Hardening Extension

The A-IPS interface can also represent *proactive resilience hardening* as an optional preventive posture control with trigger semantics distinct from incident response. Periodic or risk-adaptive actions may re-establish policy-accepted constraints, revalidate credentials or delegated authority, refresh bounded context, or review persistent state before sufficient evidence exists for an incident finding. Existing work already studies proactive and reactive hardening and shows that prompt hardening can leave residual risk [24, 25]. A-IPS therefore treats hardening as risk reduction, not as evidence of an uncompromised state.

Operational traces from an agent under uncertain trust should not automatically be accepted as training, memory, or remediation material because poisoned trajectories can be promoted into persistent instructions [26]. Model-changing or experience-promoting hardening therefore requires provenance, appropriate authorization, and separately controlled post-change validation. Successful hardening does not establish a clean state.

Established concern	Agentic-process-specific manifestation	A-IPS response implication
Process or workload containment	Agents can delegate tasks and create interacting process branches	Restrict, isolate, suspend, or replace selected agents or branches without assuming the entire process has one security state
Identity and privilege	Authority may be delegated dynamically through credentials, tools, approvals, and agent roles	Revoke or narrow delegated authority, reauthorize specific actions, and bind restoration to an explicit authority decision
Malicious content or state	External information can act as instructions, persist in context or memory, and influence later behavior	Quarantine suspicious information and known derived state, repair context or memory, and validate before release
Lateral movement	Influence can propagate agent-to-agent or through shared tools and information stores	Trace propagation paths, constrain communication and shared-tool access, and contain affected peers or infrastructure where evidence supports it
Cross-domain cooperation	External and internal agents can exchange information, delegation, and authority across different trust and governance domains	Treat boundary crossings as governed trust transitions, preserve origin and authority context, and constrain local response to the authority actually held
Availability	Reasoning loops, repeated tool calls, recursive delegation, or guardrail loops can degrade latency and throughput or amplify cost	Apply time, retry, token, tool-call, rate, and cost budgets, then verify that resource consumption returned to an acceptable state
Incident reporting and diagnosis	Agents can emit self-reports, peer reports, or supervisor reports and can participate in structured tests or controlled intent elicitation	Preserve these reports as source-attributed claims, compare declared intent with authorized objectives and source-separated observations, challenge or sandbox the agent, and avoid treating self-assessment as ground truth
Proactive resilience hardening (optional)	A compromise may persist before producing an actionable finding and can reside in instructions, memory, authority, or learned behavior	Periodically or risk-adaptively refresh policy-accepted constraints, revalidate state and authority, exercise security behavior, and bound persistence without treating hardening as proof of trust
Backup and recovery	Agent state spans model/configuration, memory, context, files, process state, authority, and external side effects	Preserve policy-accepted checkpoints, clone for analysis or remediation, restore selectively, validate dependencies, and distinguish agent-state rollback from environment recovery
Eradication and recovery	Affected context, memory, artifacts, model configuration, or learned behavior may survive simple restart or isolation	Preserve evidence, remediate the affected state, independently validate the result, reauthorize, and use a recovery gate before reintegration

Table 1. Agentic-process-specific manifestations of established prevention and incident-response concerns.

3.3 Cross-Organizational Trust Boundaries

An agentic process can cross an organizational boundary even when each participating agent is locally governed. An external agent may provide messages, artifacts, delegated tasks, or authorization requests to an internal agent, and an internal agent may pass information or authority outward. The resulting security paths include inbound compromise from an external domain, outbound propagation from an affected internal process, and transitive compromise when an internal agent relies on information or authority that originated with a third party it never directly authorized. Research on cross-domain multi-agent systems already identifies the resulting loss of unified trust assumptions [27, 28]. A-IPS therefore treats each such crossing as a *governed trust transition*, not as ordinary internal messaging. Where available, the response context should preserve the source organization, agent identity, delegation chain or authority evidence, provenance references, applicable policy context, trust status, and session or correlation information.

This treatment is intended to prevent *trust laundering*. Information originating in a less trusted domain should not become more trustworthy merely because an internal agent summarized, reformatted, or forwarded it. The same applies to authority. Authentication of a remote agent establishes an identity claim under the relevant mechanism, but it does not by itself establish that the remote agent was authorized for the requested action or that its content is semantically safe. A2A reflects this distinction by leaving operation-specific authorization scope and revocation semantics to the implementation or credential issuer [29], while recent protocol analysis identifies multi-hop identity and context risks [30].

Governed trust transition

External domain → identity / provenance / delegation / policy context
→ governed trust transition → local admissibility / authority
→ permitted local response

Technical reachability ≠ cross-domain authority

Cross-domain response must also respect asymmetric authority. A local A-IPS may be able to suspend a communication relationship, quarantine inbound information, narrow local credentials, isolate an internal process branch, or require renewed authentication before accepting further work. It should not infer authority to terminate a remote agent, revoke credentials issued by another organization, or modify another organization's policy merely because such actions are technically reachable. Where coordinated response is required, evidence about the incident, requested action, local containment, and unresolved authority should be preserved for the other domain without treating cross-domain reporting as ground truth.

3.4 Backup, Cloning, Rollback, and Recovery

Checkpointing and cloning offer useful recovery mechanisms, but an agent backup is more than a copy of model weights or conversation history. Depending on the runtime, relevant state can include model and prompt versions, configuration and policy, working and persistent memory, retrieval context, files and process state, credentials and delegated authority, orchestration state, pending tasks, and references to external effects. Systems work on agent checkpoint/restore already demonstrates that files, processes, and runtime artifacts can be recovery-relevant [34, 35].

A-IPS therefore distinguishes at least four recovery artifacts. A **policy-accepted checkpoint** is a state from which recovery may be attempted. Acceptance does not establish that the state is free of compromise. A **forensic clone** preserves an affected state for investigation without allowing it to resume normal authority. A **remediation clone** is an isolated branch on which context repair, corrective exercises, configuration changes, retraining, or other remediation can be tested. A **controlled rollback** restores an approved state only after checking the dependencies required for controlled continuation. These are conceptual roles and do not prescribe a particular checkpoint implementation.

A useful **branch-and-validate recovery** pattern keeps remediation and reintegration separated by independent validation.

Branch-and-validate recovery
policy-accepted checkpoint → clone → remediation
→ independent validation → controlled replacement
→ probation → reintegration

The original affected instance can remain preserved where feasible rather than being overwritten by remediation. Restore should also be selective. Blindly restoring current memory, retrieval state, cached summaries, or generated artifacts can reintroduce the same adversarial information that triggered the incident. Conversely, rolling back only internal agent state can be unsafe when the earlier execution already changed an external system. Recent security analyses show that checkpoint/restore can enable action replay, authority resurrection, inconsistent dependencies, and duplicate or irreversible external effects [36, 37].

One possible policy pattern is *checkpoint-gated degraded operation*. It is not proposed as a generally safe recovery mechanism. Where forensic analysis identifies a reproducible security-relevant trigger or a bounded execution horizon preceding it, a policy-accepted checkpoint may be restored and the agent allowed to execute only for a bounded number of work cycles or until a defined process state or event boundary is reached. Execution is then suspended before the known risk boundary, external state is reconciled where required, and a fresh authorization is needed before another restore-and-run interval. Because wall-clock time may already have passed after rollback, the boundary should be expressed through process-relative conditions such as work cycles, tool-call counts, state transitions, or event predicates rather than an obsolete absolute timestamp. This pattern can preserve limited availability while permanent remediation is pending. Repeated rollback is not recovery and does not erase external effects. Its admissibility depends on the trigger or horizon remaining sufficiently predictable, the checkpoint and dependencies remaining policy-accepted, and repeated execution not duplicating or amplifying irreversible effects.

For that reason, A-IPS separates *agent-state recovery* from *process and environment recovery*. Restoring the agent does not undo a payment, sent message, deleted object, remote delegation, or information already propagated to peer agents. Such consequences may require compensating actions, external-system reconciliation, credential rotation, propagation tracing, or coordinated recovery of other affected process branches. A checkpoint is therefore a candidate recovery source, not evidence that the resulting process state is free of security-relevant compromise or inconsistency.

These activities need not form one mandatory sequence. A practical response may compose the functions below, skipping or repeating them according to policy and evidence.

Composable A-IPS response functions
Report/Escalate → Observe/Diagnose → Contain/Preserve
→ Remediate/Validate → Recover/Reintegrate → Learn
Composable functions, not a mandatory sequence

The functions describe response activities, while the authorization lifecycle defined below records whether each concrete intervention was proposed, authorized, executed, and verified. This section provides the design-synthesis answer to Research Question 1 by showing how agentic-process properties alter response targets, authority, observability, and recovery.

4 A-IPS Response Model

4.1 From Finding to Candidate Intervention

A-IPS assumes that an upstream detector emits a bounded finding containing at least a process window, governed expectation, supporting observations, deviation class, evidence status, impact, assumptions, limitations, and provenance references. The response layer does not reinterpret such a finding as proof of compromise. Instead, it produces one or more *candidate interventions* whose authority must be evaluated independently.

A compact response candidate can be represented as

$$\begin{aligned}\mathsf{Candidate} = (&\text{intervention id}, \text{finding reference}, \text{action},\\ &\text{target}, \text{scope}, \text{preconditions},\\ &\text{expected effect}, \text{duration}, \text{validity interval},\\ &\text{reversibility}, \text{authority requirement},\\ &\text{evidence requirement}, \text{policy version}).\end{aligned}$$

Let r denote an instance of this candidate record. To make the typed interface explicit, A-IPS also distinguishes the following record classes.

$$\begin{aligned}\mathsf{FindingRef} &= (\text{id}, \text{support ref}, \text{status}, \text{limits}),\\ \mathsf{EvidenceSufficiencyDecision} &= (\text{candidate}, \text{status}, \text{basis},\\ &\quad \text{threshold ref}, \text{policy version}),\\ \mathsf{AdmissibilityDecision} &= (\text{candidate}, \text{status}, \text{basis},\\ &\quad \text{policy version}),\\ \mathsf{AuthorizationDecision} &= (\text{candidate}, \text{principal}, \text{status},\\ &\quad \text{validity}, \text{policy version}),\\ \mathsf{EnforcementState} &= (\text{intervention id}, \text{command ref}, \text{status},\\ &\quad \text{sequence}, \text{ack/apply refs}),\\ \mathsf{EffectClaim} &= (\text{intervention id}, \text{property}, \text{target/scope},\\ &\quad \text{window}, \text{observation path}, \text{status}),\\ \mathsf{RecoveryClaim} &= (\text{intervention id}, \text{state/dependencies},\\ &\quad \text{validation refs}, \text{authorization}, \text{status}).\end{aligned}$$

These are interface record types rather than a new policy or workflow language. The enforcement-state status distinguishes local states such as issued, acknowledged, applied, or failed. None of these states establishes the intended security effect, which requires a separately controlled observation path.

The intervention identifier binds later authorization, enforcement, verification, release, and recovery records to one concrete intervention instance. Action states what is proposed. Target identifies the principal entity or object, while scope defines the affected capabilities, resources, interfaces, information dependencies, communication paths, or process branches. Preconditions bind the response to current process state and evidence. Expected effect states the intended security outcome. Duration and validity bound operation in time. Reversibility describes whether and how the intervention can be undone. Authority and evidence requirements specify who may approve the action and what support is required before execution.

A *Response Policy* R_v is a governed, versioned set of rules that maps finding properties and runtime context to intervention constraints and required authorizations. A-IPS does not prescribe one universal policy language or severity matrix. At minimum, a usable response policy should express evidence sufficiency thresholds, target and scope constraints, preconditions, duration and reversibility, intervention risk relative to security impact, delegated authority, emergency clauses, resource or cost budgets where relevant, escalation conditions, information-release conditions, and recovery prerequisites.

A-IPS separates whether a candidate is suitable from whether an authority may approve it. For finding f, candidate r, and runtime context c, response admissibility is modeled as

$$\text{admissible}_{R_v}(f, r, c) \in \{\text{yes}, \text{no}, \text{unresolved}\}.$$

Admissibility evaluates evidence sufficiency, response preconditions, target and scope fit, intervention risk, and other non-authority conditions. Evidence sufficiency is a policy-relative

decision over a bounded evidence record. Its corresponding decision status is yes, no, or unresolved against the candidate's policy-defined threshold. It is not an objective probability of compromise and is not inferred from detector confidence alone. Admissibility does not grant permission to act. For an authorizing principal or mechanism a, authorization is then modeled as

$$\text{authorize}_{R_v}(a, f, r, c) \in \{\text{allow}, \text{deny}, \text{defer}\}.$$

An *allow* decision requires an admissible candidate and valid delegated authority for the specified target, scope, duration, validity interval, and policy version. We use $\text{validAuthority}_{R_v}(a, r, c)$ to denote that a currently holds the delegated authority required under R_v to authorize candidate r for its bound target, scope, duration, and validity interval. We use $\text{fresh}_{R_v}(f, r, c)$ to denote that the finding, candidate, and runtime context remain current and consistently bound to the active policy version and intervention instance rather than being expired, superseded, or replayed. Formally,

$$\begin{aligned}\text{authorize}_{R_v}(a, f, r, c) = \text{allow} \Rightarrow\ & \text{admissible}_{R_v}(f, r, c) = \text{yes} \\ & \wedge \text{validAuthority}_{R_v}(a, r, c) \\ & \wedge \text{fresh}_{R_v}(f, r, c).\end{aligned}$$

Freshness therefore binds the finding and runtime context to the active policy and intervention instance. An enforcement or release command should therefore carry the intervention identifier, target and scope binding, policy version, issuing sequence, and validity interval. Release should reference the active intervention instance rather than only the resource name. Expired or superseded authorization cannot replay or widen a newer restriction. *Deny* rejects authorization under the governing policy, while *defer* preserves unresolved authority, approval, freshness, or process-state conditions.

The model distinguishes seven claim stages, of which six precede recovery validity. The six pre-recovery stages are *evidence support*, *evidence sufficiency*, *admissibility*, *authorization*, *execution*, and *effect*. Their composition is policy-constrained.

A-IPS typed intervention chain

FindingRef → Candidate → EvidenceSufficiencyDecision
→ AdmissibilityDecision → AuthorizationDecision
→ EnforcementState → EffectClaim → RecoveryClaim

Each transition establishes only the bounded claim of its output type.

Composition does not merge the evidentiary or authority semantics of these records. Table 2 summarizes what each stage establishes and what it does not imply.

4.2 Response Proportionality and Reversibility

A-IPS separates the impact of the suspected deviation from the potential harm caused by the intervention. Impact can include confidentiality, integrity, authorization, availability, latency, throughput, accumulated resource use, and operational cost. A high-impact finding with weak evidence does not automatically justify an irreversible response, and a low-impact finding with strong evidence does not automatically require process termination. Response policy should therefore consider both *security impact* and *intervention risk*.

Table 3 gives representative response classes. The ordering is illustrative rather than normative. Deployments may skip levels, combine controls, or require human approval for particular actions.

Soft behavioral reinforcement should not be treated as equivalent to deterministic enforcement. It may be appropriate for low-impact uncertainty or as a first corrective step, but a component suspected of adversarial influence should not be relied upon to enforce a critical restriction against itself. The same caution applies to self-diagnosis and self-declared recovery. Higher-impact conditions should increasingly favor controls outside the affected agent's own reasoning path, such as gateway restrictions, identity controls, execution suspension, information quarantine, or external isolation.

4.3 Response Admissibility and Minimum Necessary Intervention

Admissibility is intentionally not a probability of compromise and does not replace human judgment where policy requires it. Among candidates that are admissible and could satisfy the same containment objective, lower disruption is a preference only when assurance and recovery safety are not worse. For two admissible response candidates r_1 and r_2, a policy may define $r_1 \preceq r_2$ when r_1 provides no weaker containment for the stated objective and no weaker assurance or recovery safety while imposing lower intervention cost. Here, $r_1 \preceq r_2$ means that r_1 is at least as preferable as r_2 under the response policy for the stated containment objective. Candidates can remain incomparable when, for example, one is broader but easier to verify or roll back. In that case policy should defer, escalate, or require an additional approval rather than applying a universal "least disruptive" rule.

Emergency conditions require an explicit policy branch rather than an implicit exception. Emergency authorization may permit temporary containment under incomplete evidence only when the policy defines a corresponding emergency admissibility condition, valid emergency authority, bounded target and scope, duration, evidence record, and retrospective review. It does not bypass freshness or create automatic reintegration authority. Emergency authorization remains distinct from fail-open, fail-closed, or degraded-operation defaults for component failure.

4.4 Design Requirements

The following eleven properties are intended design requirements for an implementation and are not proven implementation properties in this conceptual paper. They are security conditions only under the stated trust and policy assumptions, not universal guarantees.

(i) a finding does not carry implicit enforcement authority
(ii) admissibility and authorization remain distinct decisions
(iii) every privileged intervention binds to an authorized response-policy version and delegated authority
(iv) authorization, enforcement, and release records bind to an intervention identifier, target/scope, policy version, sequence, and validity interval so stale or superseded decisions cannot silently widen newer restrictions
(v) response scope and duration are bounded by the containment objective and policy-defined intervention risk
(vi) escalation requires an explicit policy condition rather than repeated alerts alone
(vii) an issued, acknowledged, or locally applied action is not treated as effective until effect evidence is obtained through a separately controlled observation path, and a verified effect does not establish persistence beyond its observation window
(viii) recovery or reintegration authority does not follow automatically from containment authority
(ix) cross-domain interventions remain bounded by the authority domain in which A-IPS is authorized to act
(x) authorization, execution, effect-verification, and recovery evidence remain separately attributable and subject to explicit visibility and trust limits
(xi) agent statements about intent, justification, or recovery remain source-attributed claims, and contradictions with source-separated evidence may strengthen a bounded find-

Stage	Claim represented	Does not imply
Evidence support	Observations and evidence claims support a bounded finding under stated visibility and assumptions	The finding is globally true or complete
Evidence sufficiency	The bounded evidence record meets the policy-relative threshold for the proposed intervention	The intervention is admissible or authorized
Admissibility	The candidate is suitable under current preconditions, scope, freshness, and intervention-risk constraints	Delegated authority exists
Authorization	The approving principal or mechanism may authorize this current candidate under the active policy version	The intervention was executed
Execution	The intervention reached an explicitly recorded local enforcement state such as issued, acknowledged, or applied	The intended security effect occurred
Effect	The specified property was observed through a separately controlled path for the stated target, scope, and observation window	Global containment or absence of bypass paths
Recovery validity	The proposed recovered or reintegrated state passed the policy-defined recovery gate	Historical or unobserved compromise was absent

Table 2. Claim boundaries in the A-IPS intervention model. Each stage preserves what has and has not been established.

Response class	Example intervention	Primary purpose	Typical property
Report / Escalate	Self-report, peer report, supervisor report, human escalation	Surface suspected incident or uncertainty	Claim, not ground truth
Observe / Diagnose	Increase telemetry, elicit declared intent, challenge, test, sandbox, replay	Acquire evidence, test cross-source consistency, and localize affected state	Non-destructive where feasible
Reinforce	Restate security constraint, request self-check	Low-disruption behavioral correction	Highly reversible, low assurance
Verify / Reauthorize	Require human approval, second factor, independent policy check	Resolve uncertainty before privileged continuation	Reversible gate
Restrict	Reduce tool set, API scope, data domain, communication path, egress, or budgets	Limit consequence while preserving partial operation	Scoped and reversible
Isolate / Quarantine	Separate agent, branch, tool, document, memory item, or derived state	Contain propagation or continued influence	Strong containment
Preserve	Snapshot context, evidence, state, and affected artifacts	Retain forensic and recovery-relevant state	May precede destructive response
Suspend / Revoke	Pause execution, disable credential, revoke delegated authority	Stop privileged action pending review	Strong and usually reversible
Remediate / Adapt	Repair context or memory, corrective exercises, retraining, configuration change	Correct affected behavior or state	Requires independent validation
Terminate / Replace	End affected instance or branch and start governed replacement	Stop continuing harmful execution	High disruption
Recover / Reintegrate	Restore approved state, authority, context, or connectivity	Return to controlled operation after validation	Requires follow-up evidence

Table 3. Illustrative A-IPS response classes. They are composable activities rather than a mandatory severity ladder.

ing but do not by themselves establish deliberate deception or latent malicious intent

5 Architecture and Enforcement

A practical A-IPS deployment can separate six functions. An upstream evidence-producing detector supplies bounded findings. This can include the A-IDS model proposed in prior work or another detector with explicitly represented source and trust properties. A **response assessor** proposes candidate interventions. A governed **response-policy authority** determines whether the proposed action is permitted and what approvals are required. A **response controller** issues authorized commands. One or more **enforcement points** apply the action through tool or API gateways, identity systems, network controls, orchestrators, agent-runtime controls, retrieval or memory controls, content stores, or existing prevention products. A separately controlled **effect verifier** observes whether the intended state change actually occurred within its defined property, scope, and observation window. The corresponding response-stage pipeline, rather than a one-to-one component diagram, is summarized below.

A-IPS response architecture

bounded finding → response assessment [evidence sufficiency / admissibility] → authorization
→ enforcement / execution → scoped effect verification → recovery gate / validity
→ response evidence / observation stream

5.1 Detection-to-Prevention Interface and Infrastructure Integration

A-IPS consumes bounded findings from an upstream evidence-producing detector as governed inputs to response assessment, not as self-authorizing enforcement commands. The prior A-IDS model is one such upstream source rather than an assumed external standard. A practical detection-to-prevention interface should preserve the finding identifier, process or correlation identifier, deviation class, evidence status, operational impact, policy version, assumptions, limitations, and references to supporting evidence. This preserves bounded detector semantics instead of collapsing a finding into a binary block signal. Equivalent bounded findings from other detectors can be accepted when their source and trust properties are represented explicitly.

The interface is bidirectional at the evidence level but not at the authority level. An intervention can change the monitored process state by revoking a credential, restricting a tool, quarantining information, isolating an agent, repairing persistent context, or suspending a process branch. Authorization, execution, and effect-verification records should therefore re-enter the observation and evidence stream. A-IDS may use those records when evaluating later process state or follow-up expectations, but the feedback does not grant the detector authority to authorize or execute the intervention that produced it.

Detection-to-prevention evidence boundary

Detector / A-IDS → bounded finding → A-IPS assessment
→ policy-governed authorization → enforcement
→ separately controlled effect verification → response / effect evidence
→ observation and evidence stream → detector
Evidence feedback is bidirectional. Authority is not.

A-IPS is intended to complement rather than replace existing prevention and enforcement infrastructure. Conventional intrusion-prevention systems, endpoint or network response controls, identity and privileged-access systems, firewalls, API or tool gateways, cloud controls, and orchestrators can serve as enforcement points or adapters. Existing detection, SIEM, or SOAR infrastructure can provide findings and context upstream, while existing prevention controls execute policy-authorized actions downstream. This permits an agent-

process-aware response layer to be introduced without requiring replacement of established security infrastructure [4, 6].

Assessment, approval, diagnostic, and verification steps can themselves add latency and cost. Deployments should therefore bound their depth by policy and apply more expensive paths selectively according to impact, uncertainty, and intervention risk rather than treating maximum verification effort as a universal default.

The same infrastructure can also contribute diagnostic evidence. A controlled statement from an agent about its objective or prior actions can be compared with telemetry from source-separated network, endpoint, identity, gateway, or data-protection controls. This comparison does not turn conventional telemetry into a semantic truth oracle. Instead, it can identify bounded inconsistencies between what the agent declares and what independently governed observation paths recorded. Such inconsistencies may increase evidence support or justify further containment, questioning, or escalation under policy, while the inference of deliberate lying or malicious latent intent remains outside the claim.

The same records may also support optional, evidence-backed security posture summaries at multiple governed scopes. These can include an individual agent, an agentic process or branch, an agent organization, and supporting domains such as identity, tools, memory, or cross-domain dependencies. A posture summary can represent policy conformity, authority integrity, context or memory integrity, propagation exposure, resource behavior, recovery status, and other policy-defined security dimensions. These summaries are bounded policy views rather than additional A-IPS decision primitives or ground-truth security scores. Higher-level summaries should preserve critical dependencies, visibility limits, uncertainty, and references to supporting evidence rather than average them away. Interventions can then be assessed partly by whether affected dimensions return to a policy-accepted operating envelope.

The separation is a governance boundary as well as a technical boundary. Detection logic should not automatically inherit the authority to revoke credentials, isolate systems, or terminate processes. A semantic response analyzer may recommend an action, but the final authorization and privileged enforcement path should remain governed by explicit policy and independently controlled credentials. This is especially important when the analyzer itself processes adversarial content or findings derived from semantic classifiers.

Human approval is likewise not sufficient merely because a person is in the loop. A governed approval should bind the approving principal to the specific finding or response candidate, target, scope, response-policy version, and validity interval. Reuse outside that context should require an explicit policy rule. Timeout, revocation of approval, and emergency substitution should be defined by policy, and the approver should receive the bounded evidence status and intervention consequences needed to make the delegated decision. Human involvement also requires an operationally usable situational representation. The approving principal should be able to identify affected agents and agentic processes, relevant evidence and limitations, current security posture at applicable levels, active interventions, authority and trust boundaries, and the expected consequences of available response options. A dashboard is one possible implementation, but A-IPS does not prescribe a user interface. Any such representation remains a bounded evidence view rather than ground truth and should preserve uncertainty, visibility limits, and source attribution. Without an adequately bounded situational representation, human approval may provide nominal authorization without sufficient basis for a well-informed operational decision.

The intervention lifecycle should be auditable and typed. Its main backbone is summarized below.

Intervention lifecycle
Proposed → Admissible / Rejected / Unresolved
→ Authorized / Denied / Deferred
→ Issued → Acknowledged → Applied / Enforcement-failed
→ Verified / Partial / Verification-unavailable
Later states can include Released, Escalated, Rolled back, or Recovery-gated / Reintegrated.

This is not a linear terminal automaton. Side states can include **Expired**, **Superseded**, **Verification-unavailable**, **Rollback-pending**, **Recovery-blocked**, or **Reintegration-denied**. A scoped verified claim can be reopened when new evidence exposes a bypass or invalidates its observation window. Conflicting candidates for the same target and scope must be correlated so that stale release or recovery decisions cannot silently override an active restriction.

Effect is a scoped claim, not a global Boolean. Effect verification is *independent* in the bounded architectural sense only when it does not rely solely on the component that issued or acknowledged the intervention and observes a security-relevant state or behavior through a separately controlled evidence path. Credential revocation can be checked against identity-service state and rejected credential use, tool restriction against gateway state or denied invocation, and process suspension against runtime state. The resulting claim is bounded by property, target and scope, observation path, and window. A denied test call therefore does not prove that all bypass paths are closed. Failed, partial, or unavailable verification should remain explicit rather than being collapsed into success. A verified effect establishes only what was observed within that scope and window. It does not establish that the effect will persist. Where policy requires, successful remediation, hardening, release, or reintegration should therefore remain under post-intervention monitoring for recurrence, compensating behavior, renewed influence, propagation, or degradation of the applied control.

Response decisions and actions should themselves enter the evidence stream. At minimum, the system should retain the intervention identifier, triggering finding, policy version, authorization and approving principal, target/scope binding, issuing sequence and validity interval, enforcement command, acknowledgments, local application records, effect evidence, and rollback or release decision. A release command should reference the active intervention identifier. Authorization, execution, and effect evidence remain distinct, and failed, partial, superseded, or rolled-back interventions remain part of the retained history.

These response records inherit the same evidentiary limitations as other logs and observation paths. Integrity protection does not establish that a response was appropriate or effective. A missing record does not prove that an intervention did not occur. Source compromise, delayed logging, conflicting acknowledgments, clock or identity uncertainty, and incomplete visibility can all weaken assurance. Response evidence should therefore preserve source, trust, timing, integrity, and visibility properties rather than being treated as an authoritative history merely because it was produced by the prevention plane. This allows later incident analysis to distinguish what the agent did, what A-IPS authorized, what enforcement infrastructure executed, and what effect was independently observed.

5.2 Illustrative End-to-End Response

Consider a bounded A-IDS finding for a privileged payment-tool invocation for which the required approval cannot yet be resolved because approval-path visibility is incomplete. The finding has high operational impact but does not establish an approval bypass. The response assessor may propose both termination of the agent and a ten-minute restriction on further payment-tool calls combined with independent reauthorization. Under a policy that requires stronger evidence for irreversible action, termination is deferred, while the scoped restriction is

allowed because its evidence sufficiency threshold, delegated authority, target, duration, and reversibility conditions are satisfied. The response controller issues the restriction through the tool gateway. Gateway acknowledgment records receipt or acceptance but not effect. A local application record can show that the rule was applied, while a source-separated observation of the active gateway rule or a subsequently denied invocation verifies the restriction. If later evidence resolves the approval state and the recovery gate confirms credentials, context, authorization, and reintegration conditions, the restriction may be released. The finding, authorization decision, enforcement action, verification evidence, and release decision remain separately recorded throughout the lifecycle.

5.3 Fail States and Response Loops

A-IPS needs explicit behavior when its own components fail. These fail-state defaults are distinct from emergency authority. Emergency authority is an explicit policy permission to act under incomplete evidence, while fail-open, fail-closed, or degraded operation specifies what happens when a required prevention component is unavailable or unreliable. A universal fail-closed policy can create avoidable denial of service, while a universal fail-open policy can preserve access that policy would otherwise restrict. The response policy should therefore define fail-open, fail-closed, or degraded-operation behavior by resource class and intervention type. For example, failure to reauthorize a destructive financial action may justify blocking, while failure of a low-risk logging enrichment step may justify continued operation with reduced assurance.

Degraded operation should not use human takeover as an unqualified fallback. If manual authorization or control is required, the operator needs the bounded situational representation described above for the affected agent organization and response state. Without it, human approval may remain nominal rather than provide a sufficient basis for a well-informed operational decision. The interface design itself remains outside the scope of this paper.

Automated response can also create loops. A restriction may cause an agent to retry, the retry may produce another finding, and repeated findings may trigger progressively stronger interventions. An adversarial information object may similarly induce repeated planning, recursive delegation, guardrail reasoning, or tool use that increases latency and accumulated cost without producing an obviously unauthorized endpoint action [17, 18, 19]. A-IPS should therefore support policy-defined cooldown periods, retry budgets, time and cost budgets, idempotent enforcement, response correlation, and conditions for escalation or restoration. Diagnostic work should also be bounded by rate, concurrency, time, and cost so that elicitation, replay, or verification cannot itself become an unbounded availability path against the prevention plane. A later finding should not automatically multiply the same enforcement action if the relevant restriction is already active.

5.4 Recovery and Reintegration

Recovery is not simply the inverse of isolation. A process may remain affected by security-relevant compromise or inconsistency after the triggering behavior stops because credentials, delegated authority, working memory, retrieved context, persistent memory, generated artifacts, information derivatives, or downstream state may have been altered. Persistent prompt-injection payloads can influence later sessions, so reconnecting the same agent instance after removing only the original source can restore the original risk [15, 16].

In the proposed model, recovery is a governed state transition with its own evidence and authority requirements. A recovery gate can require that the triggering condition is no longer active, that credentials and delegated authority are reviewed, and that relevant information, context, memory, and artifacts are validated or replaced. It can also require that the selected checkpoint or clone fits the recovery boundary, external side effects and dependencies are reconciled, remediation passes independent tests, reintegration is separately authorized where required, and reconnection is verified. These dimensions are policy-defined. The paper specifies the role of a recovery gate, not a universal test suite that can establish recovery free of security-relevant compromise or inconsistency across deployments. Recovery may require credential rotation, selective restoration, context or memory repair, compensating external actions, replacement, or a probation period with reduced authority. Correct checkpoint restoration is insufficient when restored state conflicts with external effects or dependencies [36, 37].

Recovery authority should remain distinct from emergency containment authority. A component permitted to isolate an agent immediately need not also be permitted to restore its credentials or reconnect it to sensitive systems. This distinction limits the consequence of a compromised response component and supports independent review before high-value access is returned. The resulting recovery evidence also provides a clear boundary for later incident reconstruction. It records when the system moved from containment toward controlled operation and under which policy version that decision was made.

6 Prevention-Plane Threats and Governance

6.1 Threat Model

The prevention-plane threat model distinguishes manipulation of findings, response assessment, response policy, authorization, enforcement, information quarantine, remediation, recovery state, and verification. An adversary may forge, delay, suppress, or replay findings, semantically manipulate assessor input, substitute or downgrade policy, replay an authorization, induce response loops, contaminate recovery state, confuse cross-domain authority, compromise an enforcement adapter, or spoof an effect-verification path. Protected assets include response-policy integrity and authorization, enforcement credentials, target scope, availability of legitimate agentic processes, recovery validity, retained response evidence, and the independence of effect verification.

Trust assumptions are role-specific rather than global. A detector is trusted for bounded provenance under its observation model, not for truth. A response assessor may recommend candidates but cannot create authority. A policy authority is trusted for integrity of the active authorized policy state, not for correctness of every policy choice. An enforcement point is trusted only for its local action, an effect verifier only within its separately controlled observation path, and human approval is an authorization input rather than proof of technical effect. The exclusion of simultaneous adversarial control over the authoritative policy/authorization path, every relevant enforcement point, and every observation path is a security boundary of the model. Under such total control, A-IPS cannot provide meaningful assurance that a response was authorized or effective.

The prevention plane is itself a high-value attack surface. In A-IDS, a compromised monitor can create a false finding. In A-IPS, that false finding may additionally trigger an operationally harmful intervention. Detector compromise can therefore degrade the evidence basis, but a detector finding still cannot create authorization by itself. An attacker may therefore seek either to suppress a justified response or to induce an unjustified one. The latter can turn the prevention system into a denial-of-service, cost-amplification, information-suppression, or privilege-disruption mechanism against legitimate agentic processes.

Indirect prompt injection makes this risk especially relevant when semantic components participate in response assessment [40]. A malicious artifact may influence an LLM-based response assessor even when the monitored process that produced the artifact was otherwise authorized. If the same content also enters persistent memory or downstream agent communication, the response plane can face both semantic manipulation and propagation risk [13, 15, 14]. Technical isolation limits what the assessor can modify directly, but it does not by itself guarantee that its recommendation is trustworthy. As in the detection layer, source and trust properties of semantic outputs should therefore remain visible to the authorization policy.

A-IPS should apply segregation of duties to high-impact intervention paths. The component that detects a deviation should not unilaterally redefine response policy. The component that recommends isolation should not automatically possess the credential that performs isolation. High-impact response-policy changes may require separate keys, multi-party approval, or independent review. Emergency authority may be necessary, but its scope, duration, activation condition, and retrospective review should be governed and evidenced.

A prevention policy can also be wrong. Correct enforcement of an unsafe or unauthorized policy is still harmful. Integrity protection, external anchoring, or signed configuration history can establish that a particular response policy existed, but they do not establish that the policy was appropriate. Policy authorization, provenance, testing, rollback, and supersession must therefore remain distinct from integrity evidence.

Cross-organization agentic processes require explicit authority and trust domains. Current research already recognizes that cross-domain multi-agent systems can move information, state, decisions, and authority across principal boundaries and that local safety assumptions need not compose [27, 28]. A response policy should identify which organization may authorize restrictions against shared agents, tools, credentials, communication paths, or data, which external identities and delegation evidence are accepted, and how conflicting policies or approvals are resolved. Technical ability to disable a shared resource does not establish authority to do so for another organization. Where no common authority exists, A-IPS should preserve the conflict and defer or constrain intervention according to the applicable agreement rather than silently selecting one party's policy. The unresolved authority conflict should itself be retained as response evidence together with the applicable policy versions, identity or delegation claims, and local containment actions. A-IPS does not assume a universal least-common-authority rule across organizations. Precedence, joint authorization, notification duties, or mandatory deferral must be established by the relevant inter-organizational agreement or policy. Ongoing IETF work on cross-organizational agent delegation illustrates that recursive attenuation, cross-domain verification, principal binding, revocation, and composable audit are separate requirements rather than consequences of connectivity alone [31].

Compliance requirements may influence response policy, but a technical response should not be presented as a legal conclusion. A later compliance-focused model can reuse the same pattern of governed expectations, bounded findings, authorized treatment, and evidence feedback while keeping technical control deviations distinct from legal non-compliance.

7 Evaluation and Limitations

This paper is a conceptual design and does not report empirical prevention performance. Research Question 1 is addressed through design synthesis from the agentic-process properties and response requirements developed above. A future empirical evaluation should primarily test the separations and failure conditions that operationalize Research Question 2. Uncertain findings must not automatically cause irreversible intervention, a compromised assessor must not create authority, stale authorization must not release a newer restriction, issuance or acknowledgment must not count as effect, and partial verification must not become a global containment claim. The recovery gate must block invalid reintegration, and local response must not exceed its authority domain. Controlled agent declarations that conflict with source-separated telemetry, persistent-state contamination, resource amplification, and unsafe rollback provide additional stress cases. Ground truth should be defined outside the evaluated prevention pipeline, for example through a signed scenario manifest or external harness. AgentDojo, ContainmentBench, and rollback-security studies provide useful scenario foundations [41, 20, 36, 37].

Core measures should include unauthorized-intervention rate, false-authorization rate, stale-command acceptance rate, verification-overclaim rate, recovery-gate bypass rate, authority-domain violation rate, post-intervention recurrence rate, benign disruption, time to verified effect, time to policy-accepted reintegration, and latency or cost overhead. These measures test whether the model's claim boundaries remain distinct under incomplete or adversarial evidence and partial or failed enforcement. Baselines can include detection-only operation, direct detector-to-block coupling, AgentSpec-style action-local runtime enforcement [9], AIR-style in-loop incident response where comparable, manual runbook response, and A-IPS ablations without evidence sufficiency, independent verification, or recovery-dependency checks.

False positives can cause unnecessary restrictions or denial of service. False negatives can permit harmful actions to continue. Delayed evidence can make a response arrive too late, while overly aggressive containment can destroy useful forensic state or interrupt legitimate recovery. Enforcement points can fail independently from the decision layer. Cross-organization processes raise additional questions about who may authorize restrictions against shared agents, tools, credentials, or data. These limitations are deployment-specific and should remain explicit rather than being hidden behind a single prevention score.

8 Conclusion and Future Work

A-IPS extends evidence-aware runtime security from detection toward controlled intervention and governed recovery. Agentic processes alter the response boundary because security-relevant state can reside in delegated authority, process branches, information and memory, shared tools, cross-domain relationships, and recovery artifacts. The same governance interface can optionally represent periodic or risk-adaptive proactive resilience hardening, while keeping such actions distinct from incident findings and proof of trust.

The central contribution is a typed, evidence-governed intervention interface whose claim boundaries do not collapse into one another. Finding support is not authorization, authorization is not execution, scoped effect is not global containment, and verified effect is not recovery validity. Preserving these distinctions allows heterogeneous detectors and enforcement mechanisms to be composed without turning uncertain evidence or compromised semantic components into unrestricted response authority.

The resulting post-intervention assurance loop is summarized below. A verified effect remains bounded to its observation window and therefore remains subject to continued observation where policy requires it.

Post-intervention assurance loop
Intervention → scoped effect verification → continued monitoring
→ recurrence or degradation finding → reassessment / hardening

Future work should examine how well the A-IPS model

holds when evidence is incomplete, conflicting, or adversarially influenced and when agentic processes operate concurrently across changing trust and policy boundaries. A second focus is operational robustness when enforcement or verification components fail, resource or cost amplification occurs, or persistent information and state continue to influence behavior after an intervention. Recovery requires particular attention because checkpoint and rollback mechanisms must preserve provenance and authority without reintroducing contaminated state or invalid external assumptions. A subsequent line of work will extend these questions into penetration testing for agentic processes. Controlled adversarial testing of individual agents and agent organizations can be used to observe security-posture degradation and recovery, acquire additional evidence, and derive hardening measures whose effectiveness can then be retested. Further work will address runtime compliance assessment while keeping technical findings distinct from legal conclusions. Evidence-based incident analysis can subsequently examine how deviations, interventions, recovery actions, propagation, and consequences can be reconstructed across the complete agentic process.

References

[1] A. Brömme, "A Black Box for Agentic Processes: Blockchain-Anchored Evidence for AI Agent Communication, Human Oversight, and GRC Audits," arXiv:2609.04017 [cs.CR], 2026. doi:10.48550/arXiv.2609.04017. https://arxiv.org/abs/2609.04017

[2] A. Brömme, "An Evidence Model for Agentic Processes: Evidence Claims, Trust Assumptions, and Policy Assessment," arXiv:2609.08481 [cs.CR], 2026. doi:10.48550/arXiv.2609.08481. https://arxiv.org/abs/2609.08481

[3] A. Brömme, "Intrusion Detection for Agentic Processes: Evidence-Based Runtime Monitoring," Zenodo, Version v0.9.1.8, 2026. doi:10.5281/zenodo.22764610. https://doi.org/10.5281/zenodo.22764610

[4] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," NIST Special Publication 800-94, 2007. doi:10.6028/NIST.SP.800-94. https://doi.org/10.6028/NIST.SP.800-94

[5] A. Nelson, S. Rekhi, M. Souppaya, and K. Scarfone, "Incident Response Recommendations and Considerations for Cybersecurity Risk Management: A CSF 2.0 Community Profile," NIST Special Publication 800-61 Rev. 3, 2025. doi:10.6028/NIST.SP.800-61r3. https://doi.org/10.6028/NIST.SP.800-61r3

[6] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero Trust Architecture," NIST Special Publication 800-207, 2020. doi:10.6028/NIST.SP.800-207. https://doi.org/10.6028/NIST.SP.800-207

[7] Y. Li, Y. Chen, H. Wen, B. Zhang, H. Liu, P. Wang, Y. Feng, and Y. Tian, "VIGIL: Runtime Enforcement of Behavioral Specifications in AI Agent Skills," arXiv:2606.26524, 2026. https://arxiv.org/abs/2606.26524

[8] N. Kahani, M. Barati, and D. Addae, "Runtime Compliance Verification for AI Agents," arXiv:2606.19242, 2026. https://arxiv.org/abs/2606.19242

[9] H. Wang, C. M. Poskitt, and J. Sun, "AgentSpec: Customizable Runtime Enforcement for Safe and Reliable LLM Agents," in *2026 IEEE/ACM 48th International Conference on Software Engineering (ICSE '26)*, Rio de Janeiro, Brazil, Apr. 2026. doi:10.1145/3744916.3764546. https://doi.org/10.1145/3744916.3764546

[10] Y. He, H. Zhu, Y. Li, S. Shao, H. Yao, Z. Liu, and Z. Qin, "AttriGuard: Defeating Indirect Prompt Injection in LLM Agents via Causal Attribution of Tool Invocations," in *35th USENIX Security Symposium (USENIX Security 26)*, Baltimore, MD, Aug. 2026, pp. 1547–1566. USENIX Association. https://www.usenix.org/conference/usenixsecurity26/presentation/he-yu

[11] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile," NIST AI 600-1, July 2024. doi:10.6028/NIST.AI.600-1. https://doi.org/10.6028/NIST.AI.600-1

[12] J. Kim, W. Guo, and D. Song, "SoK: Attack and Defense Landscape of Agentic AI Systems," in *35th USENIX Security Symposium (USENIX Security 26)*, Baltimore, MD, Aug. 2026, pp. 6047–6066. USENIX Association. https://www.usenix.org/conference/usenixsecurity26/presentation/kim-juhee-agentic

[13] D. Lee and M. Tiwari, "Prompt Infection: LLM-to-LLM Prompt Injection within Multi-Agent Systems," arXiv:2410.07283, 2024. doi:10.48550/arXiv.2410.07283. https://arxiv.org/abs/2410.07283

[14] T. D. Le, T. D. Dinh, T. L. Q. Nguyen, and C. D. Nguyen, "Cross-layer contagion of prompt injections in multi-agent swarms: a multiplex microscopic Markov chain approach," *Cybersecurity*, vol. 9, art. 191, 2026. doi:10.1186/s42400-026-00628-w. https://doi.org/10.1186/s42400-026-00628-w

[15] S. Gadgil, D. Alexander, S. Sunku, and F. Roesner, "Bad Memory: Evaluating Prompt Injection Risks from Memory in Agentic Systems," arXiv:2607.14611, 2026. doi:10.48550/arXiv.2607.14611. https://arxiv.org/abs/2607.14611

[16] M. Zha and X. Wang, "Autonomous LLM Agent Worms: Cross-Platform Propagation, Automated Discovery and Temporal Re-Entry Defense," arXiv:2605.02812, 2026. doi:10.48550/arXiv.2605.02812. https://arxiv.org/abs/2605.02812

[17] J. Luo, J. Dai, F. Liu, S. Peng, Y. Shi, T. Bu, G. Hong, X. Pan, and Y. Zhang, "Autonomy Comes with Costs: Detecting Denial-of-Service Vulnerabilities Caused by Resource Abusing in LLM-based Agents," in *35th USENIX Security Symposium (USENIX Security 26)*, Baltimore, MD, Aug. 2026, pp. 3991–4010. USENIX Association. https://www.usenix.org/conference/usenixsecurity26/presentation/luo

[18] K. Zhou, Y. Zheng, Y. He, M. Xue, X. Gong, Y. Wang, X. Zhang, and K.-Y. Lam, "Beyond Max Tokens: Stealthy Resource Amplification via Tool Calling Chains in LLM Agents," arXiv:2601.10955, 2026. doi:10.48550/arXiv.2601.10955. https://arxiv.org/abs/2601.10955

[19] Y. Zhou, X. Wang, P. Ma, Z. Xue, Z. Wang, and S. Wang, "From Shield to Target: Denial-of-Service Attacks on LLM-Based Agent Guardrails," arXiv:2606.14517, 2026. https://arxiv.org/abs/2606.14517

[20] W. Lan, S. Li, X. Lai, M. Wu, J. Yang, and H. Shen, "ContainmentBench: Trace-Based Evaluation of Post-Injection Containment in Tool-Using LLM Agents," arXiv:2607.23999, 2026. https://arxiv.org/abs/2607.23999

[21] M. ElBendary, "Agentic AI Containment Architecture for Security Hardening," arXiv:2608.26108, 2026. Accepted to IntelliSys 2026. https://arxiv.org/abs/2608.26108

[22] D. Hahm, T. Min, W. Jin, and K. Lee, "Unintended Misalignment from Agentic Fine-Tuning: Risks and Mitigation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 44, pp. 37443–37451, 2026. doi:10.1609/aaai.v40i44.41077. https://doi.org/10.1609/aaai.v40i44.41077

[23] Z. Xiao, J. Sun, and J. Chen, "AIR: Improving Agent Safety through Incident Response," arXiv:2602.11749, 2026. Accepted at the 43rd International Conference on Machine Learning (ICML 2026), Seoul, South Korea. https://arxiv.org/abs/2602.11749

[24] H. Chen, V. Aranda, S. Keshari, R. Heartfield, and N. Nichols, "Proactive Hardening of LLM Defenses with HASTE," arXiv:2601.19051, 2026. doi:10.48550/arXiv.2601.19051. https://arxiv.org/abs/2601.19051

[25] M. Roshan and S. Nagesh, "Securing Multi-Agent Systems: An Empirical Analysis of Security Prompt Hardening and Residual Risks," Google Research, 2026. Google Research publication page.

[26] J. Chen, L. Jiang, X. Deng, X. Du, J. Ma, Y. Feng, Y. Qing, Z. Yuan, L. Du, and J. Wang, "When Experience Becomes Instruction: Trajectory Poisoning in Self-Evolving Agent Skill Systems," arXiv:2608.05563, 2026. doi:10.48550/arXiv.2608.05563. https://arxiv.org/abs/2608.05563

[27] R. Ko, J. Chung, S. Zheng, C. Xiao, T.-W. Kim, M. Onizuka, and W.-Y. Shin, "Seven security challenges in cross-domain multi-agent LLM systems," *npj Artificial Intelligence*, 2026. doi:10.1038/s44387-026-00128-9. https://doi.org/10.1038/s44387-026-00128-9

[28] R. Yang, J. Xu, Z. Liu, N. Fendley, Y. Hong, Z. Li, and Y. Cao, "SoK: When Safe Agents Fail Together: The Security of Multi Agent LLM Systems," arXiv:2609.00595, 2026. doi:10.48550/arXiv.2609.00595. https://arxiv.org/abs/2609.00595

[29] A2A Project, "Agent2Agent (A2A) Protocol Specification," version 1.0.0, 2026. https://a2a-protocol.org/v1.0.0/specification

[30] A. Lotfi, M. M. Rahman, I. Karim, and E. Bertino, "A2ABreak: Systematic Security Analysis of the A2A Protocol," arXiv:2609.10871, 2026. doi:10.48550/arXiv.2609.10871. https://arxiv.org/abs/2609.10871

[31] M. Reece, "Cross-Organizational Delegation for Workload and Agent Identity: Problem Statement and Requirements," Internet-Draft draft-reece-wimse-cross-org-delegation-02, work in progress, Aug. 2026. https://datatracker.ietf.org/doc/draft-reece-wimse-cross-org-delegation/02/

[32] Y. Li, S. Volkov, H. Liu, Z. Xu, X. Chen, T. Zhou, D. Shao, H. Sun, and Y. Lu, "Beyond Agent Harnesses: Cross-Substrate Authority for Multi-Agent Systems," arXiv:2609.08472, 2026. https://arxiv.org/abs/2609.08472

[33] S. Boboila, X. Cadet, E. Koh, D. Balasubramanian, D. Van Bruggen, P. Chin, and A. Oprea, "BlueSTAR: Tiered Agentic Architecture for Autonomous Cyber Defense," arXiv:2609.11852 [cs.CR], 2026. https://arxiv.org/abs/2609.11852

[34] Y. Dong, J. He, Y. Hou, D. Du, Z. Xu, S. Yu, Y. Xia, and H. Chen, "DeltaBox: Scaling Stateful AI Agents with Millisecond-Level Sandbox Checkpoint/Rollback," arXiv:2605.22781, 2026. doi:10.48550/arXiv.2605.22781. https://arxiv.org/abs/2605.22781

[35] T. Wu, C. Chang, L. Cao, W. Gao, and W. Wang, "Crab: A Semantics-Aware Checkpoint/Restore Runtime for Agent Sandboxes," arXiv:2604.28138, 2026. doi:10.48550/arXiv.2604.28138. https://arxiv.org/abs/2604.28138

[36] Y. Zheng, Y. Yang, W. Zhang, and A. Quinn, "ACRFence: Preventing Semantic Rollback Attacks in Agent Checkpoint-Restore," arXiv:2603.20625, 2026. CoDAIM Workshop 2026. doi:10.48550/arXiv.2603.20625. https://arxiv.org/abs/2603.20625

[37] G. Wu, D. Li, K. Jiang, J. Niu, C. Wang, and Y. Zhang, "Safe to Resume? Breaking Execution Continuity of Agent Execution via Rollback," arXiv:2608.29381, 2026. doi:10.48550/arXiv.2608.29381. https://arxiv.org/abs/2608.29381

[38] J. H. Saltzer and M. D. Schroeder, "The Protection of Information in Computer Systems," *Proceedings of the IEEE*, vol. 63, no. 9, pp. 1278–1308, 1975. doi:10.1109/PROC.1975.9939. https://doi.org/10.1109/PROC.1975.9939

[39] N. Hardy, "The Confused Deputy: (or why capabilities might have been invented)," *ACM SIGOPS Operating Systems Review*, vol. 22, no. 4, pp. 36–38, 1988. doi:10.1145/54289.871709. https://pdos.csail.mit.edu/6.828/2009/readings/hardy-confused-deputy.html

[40] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection," in *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec 2023)*, pp. 79–90, 2023. doi:10.1145/3605764.3623985. https://arxiv.org/abs/2302.12173

[41] E. Debenedetti, J. Zhang, M. Balunović, L. Beurer-Kellner, M. Fischer, and F. Tramèr, "AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents," *Advances in Neural Information Processing Systems*, vol. 37, pp. 82895–82920, 2024. doi:10.52202/079017-2636. https://doi.org/10.52202/079017-2636

Declaration on the Use of AI Tools. AI language models were used as research and writing support during preparation of this paper. The research direction, core ideas, conceptual constructs, and substantive argumentative choices were defined by the author and repeatedly steered in detail throughout the iterative development of the manuscript. OpenAI GPT-5.6 supported literature-oriented analysis, drafting, critical examination, iterative refinement of these author-defined concepts, and LaTeX/PDF artifact generation. Claude, Gemini, Grok, and Perplexity were used as independent review systems for critical manuscript assessment, literature checking, and identification of conceptual or methodological weaknesses. Their feedback was evaluated by the author and selectively incorporated. The author remains solely responsible for the manuscript, its claims, references, and conclusions.