AIMX + ILB Whitepaper

Epydios Technologies, Inc. — Conceptual overview (exported from text)

AIMX + ILB: Governed Identity and Action-Boundary Enforcement for Tool-Using AI
Whitepaper v2 (high-level overview)

Thesis (one line)
Make governance enforceable by treating identity and authority as runtime state (AIMX) and by interposing at execution interfaces where actions become effects (ILB), rigorously designed for standards alignment to support control mapping to ISO/IEC 27001:2022 Annex A, SOC 2 Common Criteria and NIST SP 800-171 by tying decision-time authority to action-boundary enforcement and evidence chains.

Executive summary
Tool-using AI turns model output into real-world effects. When systems can call tools, move money, change production configurations, touch sensitive data, or operate through delegated agents, governance must be enforceable inside the runtime loop, not only stated in policy or prompts.

This whitepaper describes two complementary runtime layers.

AIMX: governed identity and authority context as runtime state. AIMX maintains who is acting now, under what role and authority, and supports controlled transitions such as step-up, constraint tightening, and revocation.

ILB: action-boundary evaluation and enforcement. ILB interposes at execution interfaces, evaluates proposed actions at the moment they would take effect, attaches constraints or requires escalation, and records evidence that binds intent, context, decision, and outcome.

If you remember one thing: governance becomes real when identity and authority are explicit at decision time and enforcement is anchored at the action boundary.

What this enables
• Clear accountability: decisions are coupled to identity, role, authority, and posture.
• Consistent enforcement: actions are evaluated at execution boundaries, not only in prompts.
• Safer delegation: controlled transitions support step-up and revocation without brittle hacks.
• Better incident response: evidence links what was proposed, what was allowed, and what executed.
• Scalable governance: vocabulary and boundary-first design reduce drift as tools and teams proliferate.
• Standards alignment: control-ID crosswalk and boundary-class evidence expectations.

1. Who this is for
This document is written for readers who need a crisp, high-level runtime story and a shared vocabulary.

Technical leadership: Understand where governance attaches and what a governable runtime must provide.
Safety, risk, and compliance owners: Understand how authority and enforceability can be made inspectable and reviewable.
GRC and internal audit: Evaluate how control objectives map to runtime enforcement touchpoints and evidence categories so review relies on evidence, not narrative.
Platform and orchestration teams: Understand how identity context and boundary interposition fit into the AI runtime loop.
Partners and investors: Understand the architecture-level thesis and why it matters before deeper diligence.

2. The problem: governance fails when action is easy
Traditional governance assumes humans are the bottleneck. Humans authenticate, interpret policy, and execute actions through tools. In tool-using AI, the system can propose and execute actions continuously, at machine speed, across many tools and contexts. This creates recurring failure modes.

Identity drift. In many systems, identity is treated as a login artifact, not an evolving runtime state. The system may act on behalf of a user, a service account, an agent, or a delegated role, and that “who” can shift implicitly as context accumulates. When identity is inferred from prompts or fragile session metadata, it becomes hard to answer basic questions during an incident: who authorized this action, under what scope, and why did the system believe it was allowed.

Authority ambiguity. Authority is frequently encoded as an informal notion of “allowed tools” or “allowed tasks.” That works until the system encounters gray zones: delegated authority, emergency escalation, step-up requirements, or conflicting constraints across policies. Without explicit authority context, enforcement becomes inconsistent and review becomes narrative rather than evidence.

Boundary bypass. Many systems log decisions, but enforcement happens elsewhere or not at all. If the execution surfaces can be reached without governance, governance becomes advisory. In high-stakes domains, advisory governance is not governance.

These failures are not primarily model failures. They are runtime architecture failures. The fix is to make identity and authority explicit inside the loop, and to attach enforcement to the execution boundary.

2.1 Why now

Tool-use and agentic patterns are pushing AI systems from advisory roles into action-taking roles. As soon as an AI system can trigger irreversible effects, the cost of “close enough” governance rises sharply: small mis-scopes become high-impact mistakes, post-incident narratives replace evidence, and organizations discover they cannot reliably explain who authorized what, under what conditions, and why the system believed it was allowed.

At the same time, AI systems are becoming more compositional. They chain tools, delegate to sub-agents, and operate across multiple data domains and services. That increases the surface area where identity can drift and where execution paths can bypass governance. The practical implication is that governance must be attached to runtime state and to the action boundary, not merely to prompts, policy documents, or retrospective logging.

2.2 Credibility anchor

This framing is built around recurring failure modes observed in modern tool-using and agentic systems: identity drift across sessions and delegation, authority ambiguity under escalation, and governance that exists as narrative rather than enforceable control at execution surfaces. The approach is intentionally infrastructure-adjacent: it is designed to layer onto existing IAM and policy ecosystems while improving runtime legibility, boundary control, and evidence quality in regulated and high-stakes environments.

3. Requirements: what “governable runtime” must provide

A governable tool-using AI runtime should provide outcomes. These are requirements on behavior, not on implementation details.

R1. A stable answer to “who is acting now.”

The runtime maintains a governed identity state for the acting entity, not just a login token. That state is legible at decision time and review time.

R2. A stable answer to “under what authority.”

The runtime supplies an authority context for each proposed action, derived from governed identity state and relevant policy inputs. Authority is separable from identity and role.

R3. Controlled transitions, not implicit drift.

Identity-relevant state changes occur through controlled transitions that can be explained and reviewed. Step-up, constraint tightening, and revocation are expressible without rewriting the system.

R4. Enforceability at the action boundary.

The runtime has a reliable attachment point where actions are evaluated at the moment they would execute, using identity and authority context, and where constraints can be attached or execution denied.

R5. Evidence that binds decision to outcome.

The runtime produces evidence artifacts that tie identity, authority context, posture, proposed action, evaluation result, and execution outcome into a coherent chain.

4. System at a glance

Figure 1. Conceptual placement of AIMX and ILB in a tool-using AI runtime (conceptual, non-implementing)

```
Inputs: user intent, policy, environment context
|
v
Orchestrator / Planner
|
| (governed identity + authority context)
```

```
v
AIMX
|
v
Proposed Action (intent)
|
| (boundary evaluation + constraints + escalation)
v
ILB
|
v
Tools / Systems / External Effects
```

Evidence artifacts flow to review and audit surfaces.

Note: Throughout this whitepaper, terms name governance concepts and evidence categories, not storage objects, message formats, API shapes, or integration contracts.

4.1 Conceptual swimlane (conceptual, non-implementing)

Figure 2. Governance checkpoints in the runtime loop

```
Propose action → Provide context → Boundary evaluate → Execute → Record evidence
| | | | |
Orchestrator AIMX ILB Tools Review
```

5. Conceptual overview: AIMX and ILB in the runtime story

AIMX sits upstream of tool execution. It maintains governed identity state and derives authority context suitable for decisioning. AIMX can be integrated alongside existing identity providers and policy sources. It is not a replacement for them. It is the runtime layer that makes identity and authority usable inside the AI system’s loop.

ILB sits at the action boundary. It interposes at execution interfaces, evaluates proposed actions using context provided by AIMX and policy sources, attaches constraints or requires escalation, and produces evidence artifacts that bind the decision to the execution.

5.1 Core vocabulary

Identity is who the acting entity is in the governance sense. Identity can be a human user, an organizational actor, an agent instance, or a service principal, as long as it can bear accountability.

Role is the governance-relevant function or stance under which the identity is currently operating. Role helps answer what kinds of actions are expected and what constraints apply.

Authority is scoped permission to perform specific action classes under specific conditions. Authority is contextual and can change without changing identity.

Posture is a governed state that expresses the system’s current constraint profile and escalation expectations. Posture supports controlled transitions such as step-up, step-down, constraint tightening, and revocation.

Action boundary is the point where a proposed action would become an external effect. In a tool-using system, boundaries are often tool calls, workflow triggers, commits, transfers, writes, or control signals.

Evidence artifacts are categories of records that bind decisions to outcomes. In this paper, they are referenced by category labels using a lightweight `*Ref` convention.

6. AIMX: governed identity as runtime state

AIMX is an identity-governance architecture for AI systems that treats identity and authority as first-class runtime state. The guiding idea is simple: in a tool-using AI system, identity cannot be an afterthought. It must be represented, updated, and evaluated inside the same loop that produces decisions, so downstream enforcement can rely on it.

6.1 Where AIMX sits

AIMX is designed to live in or alongside the orchestration layer, upstream of tool execution. It is not a replacement for an identity provider. It is the runtime layer that makes identity and authority usable, consistent, and inspectable inside the AI system’s decision loop.

6.2 What AIMX does at a conceptual level

AIMX supports three conceptual functions.

Representation. Provide an identity representation richer than a username or token, including role and context signals that matter for governance.

Controlled updates. Ensure changes to identity-relevant state occur through controlled transitions rather than implicit drift. The goal is not to freeze identity. The goal is to make state updates traceable and defensible.

Accountability coupling. Bind identity and authority to decisions in a way that supports evidence. When the system proposes actions, AIMX supplies the identity and authority context that explains why an action is within scope, out of scope, or requires escalation.

6.3 What AIMX produces

AIMX can be described by its outputs as artifact categories rather than internal machinery. AIMX produces:

- Governed identity state for the session or acting entity (*Ref: GovernedIdentityState)
- Authority context derived from identity state and relevant policy inputs (*Ref: AuthorityContext)
- Identity-relevant audit artifacts recording state transitions and the basis for authority assertions (*Ref: IdentityTransitionLog)

6.4 What AIMX is not

AIMX is not a foundation model. It does not depend on a specific LLM. It is not merely authentication. Authentication answers who logged in. AIMX answers who is acting now, under what authority, with what governed continuity, in a form the runtime can enforce and later review.

7. ILB: enforceability at the action boundary

ILB is an action-boundary interposition and enforcement layer. It treats execution as a governed interface rather than an opaque side effect. The core idea is that model outputs should not directly become external effects. Instead, the system proposes actions that are evaluated at the boundary where they would take effect.

7.1 Where ILB sits

ILB attaches to execution interfaces. These are the points where actions would touch tools, systems, or external resources. ILB may be centralized at shared execution points or closer to local systems that execute actions. The architectural requirement is an enforceable attachment point, not a specific deployment pattern.

7.2 What ILB does at a conceptual level

ILB supports four conceptual functions.

Boundary evaluation. Evaluate a proposed action at the moment it would execute, using identity and authority context and the current posture.

Constraint attachment. Attach constraints or conditions to allowed actions, or require escalation before execution.

Execution control. Allow, deny, or defer actions based on evaluation results and posture transitions.

Evidence production. Produce evidence artifacts that bind proposed action, evaluation result, identity context, and execution outcome into a reviewable chain.

7.3 What ILB produces

ILB’s outputs are best framed as artifact categories.

- Proposed action record and boundary evaluation outcome (*Ref: BoundaryDecision)
- Constraints or conditions attached to the action (*Ref: BoundaryConstraints)
- Execution record linking the decision to the outcome (*Ref: ExecutionReceipt)
- Evidence bundle enabling review and replay at the conceptual level (*Ref: EvidenceBundle)

8. The AIMX → ILB handshake: end-to-end runtime story

AIMX and ILB are complementary. AIMX makes identity and authority legible inside the decision loop. ILB makes governance enforceable at execution.

At a conceptual level, the runtime loop looks like this:

1) The system forms a proposed action based on the task, context, and tool affordances.
2) AIMX supplies governed identity state and authority context for the acting entity.

3) ILB evaluates the proposed action at the execution boundary using authority context and posture.
4) ILB either allows execution with constraints, denies execution, or triggers an escalation and posture transition.
5) The system executes only through the governed boundary path, producing evidence artifacts that bind decision to outcome.

This handshake avoids two common failure patterns. It avoids identity inference from prompts by making identity and authority explicit runtime state. It avoids advisory governance by attaching enforcement to the boundary where action becomes effect.

9. Benefits: why this matters in practice

This approach supports practical benefits that show up in real operating environments.

Fewer high-impact mistakes. Boundary-first evaluation reduces the chance that a plausible model output becomes an irreversible effect without review.

Faster, cleaner incident response. Evidence artifacts bind intent, context, evaluation, and outcome, reducing “what happened” time and improving accountability.

More reliable delegation. Controlled transitions support step-up and revocation without resorting to brittle prompt hacks or ad hoc overrides.

Lower governance overhead at scale. A shared vocabulary and an explicit boundary story reduce drift as tools and teams proliferate.

Better partner alignment. Conceptual clarity lets teams align on what governance should mean before arguing about implementations.

9.1 What changes in practice (conceptual)

The table below summarizes the behavioral shift without prescribing implementation.

Without governed runtime layers:
- Identity is inferred from prompts or thin session metadata.
- Authority is implicit or tool-scoped rather than action-scoped.
- Enforcement is advisory or inconsistent across execution surfaces.
- Incidents are reconstructed from partial logs and narratives.

With AIMX + ILB framing:
- Identity, role, and authority are explicit runtime context at decision time.
- Posture transitions express step-up, constraint tightening, and revocation.
- Actions are evaluated at the boundary where they become effects.
- Evidence links intent, context, boundary outcome, and execution.

10. Case studies: applicability without a build recipe

These scenarios illustrate how the same conceptual machinery supports governance in different environments. They are written as decision-point stories: where the boundary is, what must be evaluated, and what evidence must exist for review.

10.1 Clinical intake: constrained assistance with escalation

A system assists with clinical intake and coordination. The governance question is not whether the model can summarize symptoms. The question is what actions the system may take: retrieving sensitive records, scheduling, routing, ordering tests, or messaging clinicians.

Posture transitions (examples)
- Step-up when moving from advisory output to an irreversible effect.
- Constraint tightening when context uncertainty rises or sensitivity increases.
- Revocation or deferment when authority cannot be established.

Evidence chain (conceptual)
- Decision-time identity, role, authority context, and posture.
- Proposed action and boundary evaluation outcome (allow, deny, escalate, constrain).
- Execution receipt and an evidence bundle sufficient for review.

Boundary decision points (examples)
- Record retrieval: allow only with appropriate authority context and posture, otherwise require escalation.

- Messaging: condition on recipient class and sensitivity, otherwise defer or deny.
- Scheduling and orders: require step-up when moving from advisory to action on protected workflow systems.

Evidence expectations (conceptual)

- What was proposed, by whom, under what role and authority context.
- What boundary evaluation occurred, what constraints were attached, and what executed.

10.2 Autonomous mobility: safe action under changing context

An autonomous mobility system operates in a dynamic environment with many interacting actors. The governance question is not only safety. It is authority under context: which entity is responsible for initiating a maneuver, when constraints tighten, and how overrides or deferrals are handled in a reviewable way.

Posture transitions (examples)

- Step-up when moving from advisory output to an irreversible effect.
- Constraint tightening when context uncertainty rises or sensitivity increases.
- Revocation or deferment when authority cannot be established.

Evidence chain (conceptual)

- Decision-time identity, role, authority context, and posture.
- Proposed action and boundary evaluation outcome (allow, deny, escalate, constrain).
- Execution receipt and an evidence bundle sufficient for review.

Boundary decision points (examples)

- Control changes: require posture-dependent evaluation before effects are applied.
- Route changes: condition on operating context and authority scope, otherwise escalate.
- Overrides and deferrals: ensure responsibility is legible and evidence-coupled.

Evidence expectations (conceptual)

- Which identity and role were active at decision time.
- Which authority context applied, what posture was in effect, and what outcome occurred.

10.3 Enterprise change authority: preventing high-impact mistakes

An enterprise workflow proposes changes to production systems, releases, or configurations. The governance risk is not only malice. It is drift, miscommunication, and under-specified authority.

Posture transitions (examples)

- Step-up when moving from advisory output to an irreversible effect.
- Constraint tightening when context uncertainty rises or sensitivity increases.
- Revocation or deferment when authority cannot be established.

Evidence chain (conceptual)

- Decision-time identity, role, authority context, and posture.
- Proposed action and boundary evaluation outcome (allow, deny, escalate, constrain).
- Execution receipt and an evidence bundle sufficient for review.

Boundary decision points (examples)

- Production changes: require explicit authority context, step-up, or approval under tightened posture.
- Sensitive configuration writes: attach constraints or deny when scope is insufficient.
- Delegated changes: ensure delegation is explicit and reviewable.

Evidence expectations (conceptual)

- Decision-time identity and authority context, and the boundary outcome.
- A coherent chain linking proposal, evaluation, approvals (if any), and execution receipt.

11. How this fits with existing stacks

This architecture is designed to layer onto existing systems rather than replace them.

Identity provider and authentication: Continue to answer who logged in and how. AIMX uses those inputs but maintains the governed “who is acting now” state inside the runtime loop.

Policy sources: Existing policy engines, RBAC/ABAC frameworks, and organizational rules can inform authority context. The whitepaper does not prescribe a specific policy substrate.

Orchestration layer: AIMX and ILB typically live in or near the orchestration layer because that is where decisions are formed and where tool calls are emitted.

Logging and observability: Evidence artifacts are conceptually distinct from logs. In practice, they should integrate with existing monitoring, audit, and incident response workflows, but the integration details are out of scope here.

12. Standards alignment and control mapping

AIMX and ILB are built to make compliance controls operational in tool-using and agentic systems. In most organizations, the hard part of frameworks like ISO 27001, SOC 2, and NIST SP 800-171 is not writing policies. The hard part is demonstrating, at the moments that matter, that the system enforced scope and produced evidence that can be reviewed. Tool-using AI forces this issue because the system's "decision" and the system's "effect" can be separated by orchestration layers, delegated agents, and external tools. Without runtime anchors, governance becomes a story told after the fact.

The architecture is intentionally shaped to support control objective coverage through mapping. AIMX provides governed identity and authority context as decision-time state: who is acting now, under what role and authority, with controlled transitions that support step-up, tightening, and revocation. ILB provides the complementary anchor: action-boundary evaluation and enforcement where tool calls become real effects. Together, they enable control objectives to be mapped to explicit runtime touchpoints and to reviewable evidence categories. In other words, they are designed so control owners can point to boundary-evaluable decisions and evidence chains rather than relying on prompts, retrospective log stitching, or implicit assumptions about authority.

This is the basis for a control-ID crosswalk approach. For ISO/IEC 27001:2022 Annex A, SOC 2 Common Criteria, and NIST SP 800-171, the mapping logic is consistent: identify the in-scope boundary classes (data access, outbound messages, configuration changes, transfers, control commands), define the authority and escalation semantics for each class, and require evidence categories that allow an auditor or reviewer to reconstruct who acted, what was proposed, what was allowed or constrained, what executed, and why. When those elements exist at the boundary, the resulting evidence chain supports objectives like least privilege, privileged action control, change authorization, traceability, monitoring and incident reconstruction without requiring the reader to accept "trust me" narratives.

Control-ID mapping materials and evidence expectations catalogs are available for diligence. They include (i) a crosswalk that links representative control identifiers to AIMX/ILB touchpoints and (ii) a boundary-class evidence expectations catalog suitable for review and incident response. This section describes architectural support for standards alignment and control mapping. It is not a certification or compliance claim, which depends on deployment scope, implementation details, operational processes, and audit outcomes.

13. Differentiators: what this approach changes

This approach is defined by a small set of principles.

Identity and authority are runtime state. Governance is strongest when the system can represent and reason about identity, role, and authority inside the loop, not as a thin external token.

Controlled transitions prevent silent drift. Step-up, constraint tightening, and revocation are treated as explicit posture transitions tied to evidence, rather than ad hoc prompt changes.

Enforcement is anchored at the action boundary. Governance becomes enforceable when actions are evaluated at the moment they would execute, and execution paths are controlled through the boundary.

Evidence makes decisions reviewable by linking context, evaluation, and outcome. Review and audit require more than logs. They require a coherent chain tying identity context, authority context, boundary evaluation, constraints, and execution results.

14. Limitations and non-goals

This approach does not claim to solve every safety or governance problem by itself.

It does not replace IAM, authentication, or enterprise identity governance. It builds on them and focuses on runtime legibility.

It does not "make the model safe." It makes governance enforceable at boundaries, but policy quality, tool design, and organizational controls still matter.

It does not remove the need for humans in the loop for high-stakes decisions. In many environments, it supports cleaner escalation and approvals.

It assumes you can control at least some execution interfaces. If execution paths bypass the boundary, enforcement will be incomplete.

15. Adoption guidance: how organizations typically engage

Organizations usually adopt governed runtime concepts in phases, even when the end-state is ambitious.

Phase 1: Align on vocabulary and runtime story. Establish shared definitions of identity, role, authority, posture, and boundary. Validate that stakeholders agree on where enforceability should attach.

Phase 2: Identify boundary attachment points. Enumerate the execution surfaces where actions become effects. Clarify what kinds of actions require evaluation, conditioning, escalation, or denial.

Phase 3: Define authority context and posture semantics. Establish how authority scope is expressed and how posture transitions should behave in common scenarios such as step-up and revocation.

Phase 4: Establish evidence expectations. Decide what categories of evidence are required for review, and what “good” looks like for accountability in the operating context.

At this level, the material is sufficient for Phase 1 alignment and Phase 2 boundary identification. Deeper phases require environment-specific, mechanism-level choices.

16.1 A practical 30/60/90 view (conceptual)

30 days: Identify your true execution boundaries and agree on the vocabulary and responsibilities.
60 days: Define authority semantics and posture transitions for a small set of high-impact action classes.
90 days: Establish evidence expectations for review and incident response and validate the end-to-end runtime story on representative workflows.

15. Scope and confidentiality boundary

This whitepaper is intended for early alignment and diligence discussions. It discloses the conceptual model, the vocabulary, and the runtime story, along with scenario illustrations. It is not intended to enable implementation.

What is intentionally out of scope includes internal representations, field-level structures, concrete integration contracts, scoring or risk models, thresholds, detailed rule tables, privacy redaction logic, conformance tests, evaluation harnesses, operational runbooks, and other mechanism-level specifics. If a detail would allow a competent engineer to build a working version from this text alone, it belongs in separate technical materials.

17. IP posture and diligence note

The concepts described here are part of an established IP posture in this space. Deeper technical materials exist for diligence discussions when appropriate, including more concrete boundary semantics, evidence expectations, and environment-specific operating stories.

18. Next Steps

If you are reviewing this as an advisor, partner, or investor, the most useful feedback is conceptual.

1) Vocabulary: Are the terms clear and non-overlapping. Where do definitions feel ambiguous or redundant.
2) Runtime story: Does the AIMX → ILB story feel coherent. Where does it jump too quickly or leave a gap.
3) Boundary reality: In your environment, where are the true execution boundaries and what would you insist be evidenced at those boundaries.

For diligence, a standards alignment mapping pack is available, including a control-ID crosswalk (ISO/IEC 27001:2022 Annex A, SOC 2 CC families and selected NIST SP 800-171 requirements) and a boundary-class evidence expectations catalog.

If this framing matches your operating needs, the next step is a structured diligence discussion centered on your execution boundaries, authority model, and evidence expectations, followed by deeper technical materials as appropriate.

Appendix A: Glossary (compact)

Action boundary: The point where a proposed action becomes an external effect.
Authority: Scoped permission to perform specific action classes under specific conditions.
Evidence artifacts: Categories of records that bind decisions to outcomes.
Identity: Who the acting entity is in the governance sense.
ILB: Action-boundary interposition and enforcement layer.

AIMX: Governed identity and authority context layer.
Posture: Governed state expressing the current constraint profile and escalation expectations.
Role: Governance-relevant function or stance under which an identity operates.
`*Ref`: A label for an artifact category, not a representation.

End of whitepaper.