ONYX Control AI, Create the Impossible

Herding ~~Cats~~ Agents

A Five-Layer Architecture for Governing AI at Scale

Tim Youngblood
Chief Strategy Officer
ONYX

Shahzad Akhai
Principal AI Security Consultant
AHEAD

Christian Kon
Director of Security Strategy, GRC, and Resilience Delivery
AHEAD

Executive Summary

Enterprise AI agent adoption is accelerating faster than the control architectures designed to govern it. Agents now authenticate to downstream systems, invoke tools across environments, access sensitive data, and influence production outcomes. Yet most governance remains built for humans or bounded applications creating a “runtime trust gap.”

Organizations struggle to answer basic risk questions with confidence:

- What agents are operating in our environment?
- What data and systems can the agents access or influence?
- When an agent makes a decision, what is our audit trail?
- If an agent behaves outside policy, what is our containment posture?

AHEAD and Onyx Security’s reference architecture for securing and governing AI agents at enterprise scale is summarized in five layers:

Together, these layers move organizations from fragmented point controls to an accountable operating model. Most of the underlying capabilities already exist in large enterprises. The real issue is whether those tools are organized around the actual execution path of modern agents. The organizations that close this gap first will treat AI agent security as an architectural discipline, not a collection of isolated controls.

Why This Matters Now

In 2023, enterprise AI security discussions centered on chat interfaces, model endpoints, and content filtering. In 2026, agents operate across far more consequential surfaces. They use credentials, call tools, retrieve and modify data, trigger actions, and increasingly act inside workflows that carry operational, regulatory, or customer impact.

AHEAD finds that most security and governance programs are designed for human activity: endpoint controls, IAM tuned for employees, and SIEM pipelines optimized for user-initiated events. The result is a structural problem — controls exist, but not always at the point where agent intent becomes action.

Defining the Runtime Trust Gap

The runtime trust gap is the difference between what companies believe their programs control vs the reality of what agents are capable of. Onyx has first-hand experience with clients seeing this gap show up in the following ways:

- ✕ An agent is approved, but its downstream tool calls are not governed.
- ✕ Logs show a human or service account, but the enterprise cannot reliably attribute which agent took which action.
- ✕ Discovery captures sanctioned deployments while shadow or embedded agents remain invisible.
- ✕ Monitoring records inputs and outputs but not the decision path that led to a sensitive action.
- ✕ Incident response assumes external compromise when the more realistic failure mode is an over-permitted or misdirected agent acting within authority.

The Five-Layer Architecture

LAYER 1

Identity and Access

Every agent authenticates throughout the enterprise using human or non-human identities. Securing them is a foundational discipline of IAM, extended across the supporting identity layers: privileged access management, secrets management, workload identity, and just-in-time authorization for processes operating without a human in the loop.

The core architectural objectives include

- Each agent has its own identity, separate from the initiating user.
- Access is scoped to task, session, or tool need rather than broad privilege.
- Credential issuance, use, and revocation are bounded and attributable.
- Delegation chains preserve both the acting identity and the originating principal.

> Without this foundation, every downstream control becomes harder to trust and AHEAD's AI Security and Identity and Access Management teams can help.

LAYER 2

Discovery and Inventory

Discovery and continuous inventory of AI agents including all sanctioned deployments, shadow agents, embedded vendor agents, coding agents, orchestration hosts, and the server infrastructure is necessary. A defensible inventory answers the questions:

- How many AI agents are operating in our environment, and what can they reach?
- What credentials do they use?
- What data, tools, or systems can they reach?
- Which business process or owner is responsible?

> Onyx's live, query-able inventory is refreshed near-real-time and directly connects to the identity layer eliminating the need for manually assembled inventories that are obsolete the moment they are published.

LAYER 3

Runtime Control

A runtime control plane must sit between agent reasoning and operation evaluating an agent's intended action before the tool call executes. This may include agent-aware gateways, function-call interceptors, policy engines, just-in-time authorization services, and runtime guardrails.

This addresses the most consequential failure mode in modern deployments: not that agents generate bad answers, but that they accidentally take a harmful action because no effective control sits between decision and execution.

A mature runtime control plane can

- Inspect tool calls before execution
- Evaluate them against policy in real time
- Allow, narrow, steer, or block based on risk and context
- Record the policy decision with agent attribution
- Establish behavioral baselines by agent role and detect drift

> When this is absent, downstream layers become reactive rather than preventive. AHEAD AI Security can help configure Onyx's Guardian Agents automating this entire layer.

LAYER 4

Telemetry and Detection

Traditional logging is not enough as detection teams need the decision trail: what the agent decided, what it attempted, what policy evaluation occurred, what data or systems were touched, and which identity performed the action. This layer includes SIEM, XDR, security data lakes, non-human entity analytics, reasoning-trace capture, and anomaly detection tuned to agent behavior rather than only human behavior.

The outcome is audit-grade telemetry, not application noise. Without the upstream runtime layer, telemetry often arrives too late when correct architecture could have prevented it.

> Onyx has direct connections to major observability platforms making this layer plug-and-play for most organizations.

LAYER 5

Governance and Response

This is where the enterprise produces the artifacts that boards, regulators, auditors, and executives expect including policy mapping, incident definitions, tested response playbooks, risk metrics, and governance reporting. Most large enterprises already have GRC platforms, incident response processes, and board reporting structures. What is often missing is the underlying technical evidence required to make those mechanisms meaningful for agentic systems.

A functioning governance and response layer can answer

- How many agents are in production, and what is their risk classification?
- What policy actions were taken against them this quarter?
- What incidents were detected, contained, or escalated?
- Which agent classes have tested incident response playbooks?
- Where does residual risk remain concentrated?

> In this layer, AHEAD's Strategy, Governance, and AI Security teams help shift the conversation from abstract statements about AI risk to evidence of control operation, exposure trends, and response readiness.

A Practical 90-Day Path Forward

AHEAD recommends for most enterprises not to build out a separate AI security program, but to refresh current policy and controls for AI applicability. Using engineering first principles, AI security relies on the same core tenets that are already present in most organizations. A practical 90-day plan focuses on three steps:

1. **Establish a real inventory for the highest-risk agents.**

   Start with coding agents, finance-related agents, HR-data agents, customer-facing agents, and any agent operating with elevated or standing access. Build a current inventory tied to runtime, credential, ownership, and business impact.

2. **Add a runtime decision point for sensitive tool calls.**

   Focus first on repository write operations, production deployment actions, secrets access, database modification, and external API calls with write scope. The objective is to create a pre-execution decision point where none existed before.

3. **Define and test one agent-specific response playbook.**

   Choose a realistic scenario (e.g., a destructive coding-agent action, an HR-data agent exceeding purpose, or an agent using credentials outside approved scope). Document the containment path, escalation path, and evidence requirements. Then test it.

This creates the foundation of governance rather than a one-time briefing exercise.

Conclusion: From Principles to Architecture

Enterprise AI agent security is no longer a narrow model-security problem. It is a control architecture problem and the organizations that move fastest will not be the ones that simply deploy more automation. They will be the ones that create a runtime control and governance architecture capable of making that automation trustworthy.

> AHEAD and Onyx's architecture gives organizations a way to identify agents, constrain them, observe them, and explain their behavior to the people who own the risk.
>
> **That is what mature ai agent governance will increasingly require.**

AHEAD

Combining cloud-native capabilities in software and data engineering with an unparalleled track record of modernizing infrastructure, we're uniquely positioned to help accelerate the promise of digital transformation.

Visit us at ahead.com

ONYX

Onyx is the Secure AI Control Plane for the Agentic Era. Discover every AI agent and model across the enterprise, govern their actions with natural-language policy, and steer agent behavior at execution time—before actions take effect. Trusted by Fortune 500 enterprises across energy, financial services, and healthcare.

Learn more at onyx.security

