Openpolicy

WHITE PAPER · JUNE 2026

A POLICY FRAMEWORK FOR AUTONOMOUS AI AGENT SECURITY

Securing Identity, Discovery, and Control in the Agentic Web

Autonomous agents now act, transact, and delegate authority across organizational boundaries. This framework sets out how identity, runtime governance, and lifecycle security must evolve together to keep them accountable.

PREPARED BY

The OpenPolicy Coalition

A COALITION ACROSS

Cybersecurity · AI Security · Cloud · Data Governance

THE CHALLENGE

How do we ensure that autonomous agents remain accountable to human principals, operate within authorized boundaries, and can be securely discovered, governed, and constrained across fragmented platforms and regulatory regimes?

Our security and policy frameworks were built for human users with persistent identities, for static software with deterministic behavior, and for centralized systems under singular control. None of these assumptions survives contact with the agentic web. This paper advances a governance framework built on three interdependent layers — and a set of policy recommendations to keep the emerging architecture open, interoperable, and accountable.

CONTENTS

EXECUTIVE SUMMARY

We stand at the threshold of a fundamental transformation in computing. Autonomous AI agents — systems that independently plan, execute, and adapt across organizational boundaries — are no longer experimental prototypes. They are operational realities embedded in enterprise workflows, government systems, and critical infrastructure, acting with a degree of independence that fundamentally alters the calculus of trust.

This transformation arrives when our policy and security frameworks remain anchored to assumptions that no longer hold. We designed cybersecurity models for human users with persistent identities, for static software with deterministic behavior, and for centralized systems operating under singular organizational control. None of these assumptions survives contact with the agentic web, where autonomous systems act, transact, delegate authority, and collaborate across dynamic environments without continuous human oversight.

The core policy challenge is easy to state but hard to answer: How do we ensure that autonomous agents remain accountable to human principals, operate within authorized boundaries, and can be securely discovered, governed, and constrained across fragmented platforms and regulatory regimes? This paper advances a governance framework built on three interdependent principles.

01 **IDENTITY & DISCOVERY**

Identity and discovery mechanisms must remain verifiable, interoperable, and anchored to accountable entities.

02 **RUNTIME GOVERNANCE**

Agent behavior must be continuously governed through runtime authorization, behavioral monitoring, and execution-layer enforcement.

03 **FULL-LIFECYCLE SECURITY**

Security and governance must extend across the full operational lifecycle — continuous monitoring, proactive security testing, adversarial validation, and controlled decommissioning.

These layers must evolve in parallel rather than sequentially. Identity infrastructure establishes accountability and trusted interaction, but identity alone cannot secure systems whose primary risks emerge during execution. Runtime governance, behavioral enforcement, and continuous visibility therefore become essential components of trustworthy agentic systems.

The window for shaping the architecture of the agentic web is narrowing. The decisions made now will determine whether this ecosystem develops as an open, interoperable security architecture — or fragments into proprietary systems that concentrate control.

The OpenPolicy Coalition, representing organizations at the forefront of cybersecurity, AI security, cloud infrastructure, and runtime governance, presents this framework alongside policy recommendations designed to reduce regulatory fragmentation, enable interoperability, and align security controls with evolving operational realities.

The decisions made now will determine whether the agentic web develops as an open and interoperable security architecture, or fragments into proprietary systems that concentrate control, constrain visibility, and amplify systemic risk. The sections that follow trace the path from the threats agentic systems introduce, through the identity and discovery foundations that establish accountability, to the runtime governance and harmonized standards required to keep them secure at scale.

HOW TO READ THIS PAPER

Sections 1–4 build the technical case layer by layer — attack surface, identity, continuous authorization, and runtime security. Section 5 addresses regulatory harmonization. The paper closes with five concrete recommendations for policymakers and standards bodies.

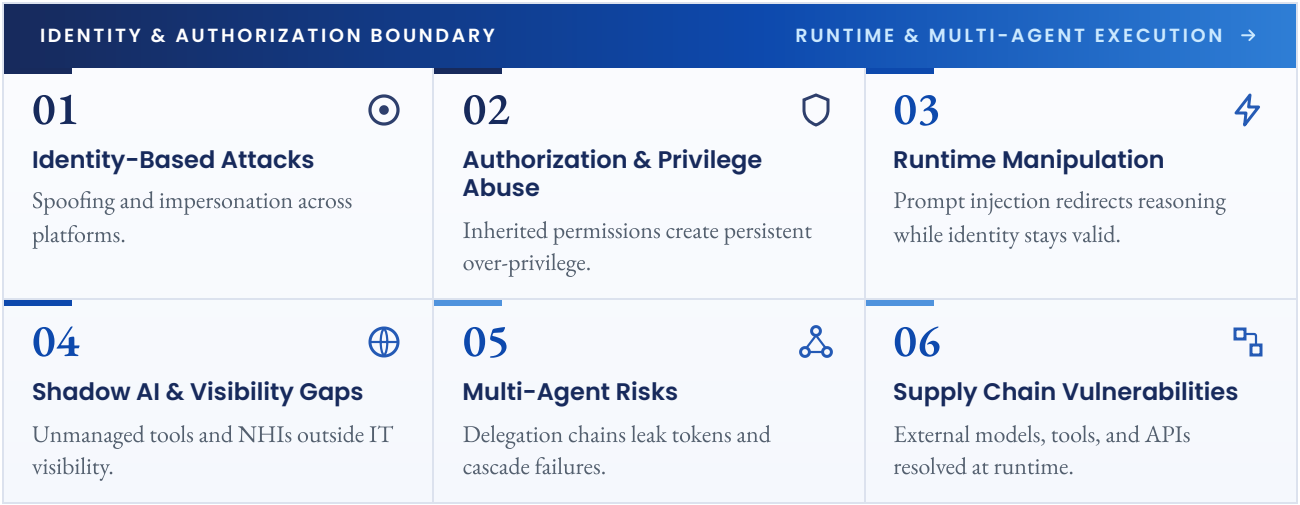
01

THE ATTACK SURFACE

Where Security Can Fail

The security challenges posed by autonomous AI agents cannot be reduced to any single vulnerability. They represent a **layered attack surface** that extends from identity to runtime behavior to multi-agent interdependencies — six recurring categories of failure that compound rather than isolate.

FIGURE 1 — THE LAYERED ATTACK SURFACE



Identity-Based Attacks. Agents without a cryptographically verifiable identity can be spoofed or impersonated across platforms. Organizations cannot reliably determine who authorized an action or which entity bears accountability. The absence of cryptographic binding between an agent's claimed identity and its actual provenance creates an environment where trust must be assumed rather than verified.

Authorization & Privilege Abuse. Agents commonly inherit the full permissions of the human user who created them, creating massive over-privilege that persists throughout operational lifetimes. A finance analyst with broad CRM access builds an agent for routine reporting — and that agent retains full privileges to sensitive customer data far beyond what its function requires. Without per-operation enforcement, compromised agents possess keys to far more than their legitimate functions demand.

Runtime Manipulation. Prompt injection represents a fundamentally new attack class with no analog in traditional software security. Adversaries who control agent inputs — through malicious emails, compromised data sources, or manipulated web content — can redirect reasoning processes and cause agents to execute unauthorized actions while appearing legitimate. The agent's identity remains valid and its session credentials are authentic, yet it serves adversary objectives. The compromise occurs within the reasoning process itself and is invisible to boundary-focused security controls.

FIELD EVIDENCE · AUGUST 2025

Zenity Labs unveiled **AgentFlayer** — zero-click exploit chains that compromised enterprise AI agents including ChatGPT with Google Drive integration, over 3,000 publicly accessible Microsoft Copilot Studio agents, Salesforce Einstein, and Cursor with Jira integration — all without any user interaction.[1] In each case, attackers achieved full compromise while the agent's identity remained valid and authorization checks passed.

These incidents illustrate a fundamental limitation of identity-centric security models. Prompt injection operates inside otherwise legitimate identity and authorization boundaries. An agent may remain cryptographically authenticated, operate with legitimately issued credentials, and execute actions that technically conform to its permissions, while still pursuing adversarial objectives introduced through manipulated inputs. Effective detection must operate at the reasoning and decision layer, evaluating whether behavior remains aligned with the application's intended scope and the user's delegated purpose. Intent-alignment monitoring and execution-layer enforcement, therefore, become primary security controls rather than secondary safeguards.

Shadow AI & Visibility Gaps. Recent industry research indicates that over 80% of knowledge workers use unauthorized AI tools regularly.[2] Low-code platforms let business users deploy agents without security review, creating a shadow ecosystem outside IT visibility. Non-human identities (NHIs) now substantially outnumber human identities in many enterprise environments. Organizations cannot govern what they cannot see — an expanding, largely unmanaged attack surface.

This is not a challenge of visibility alone, but of governance. As a baseline, organizations should maintain continuous inventories of deployed agents, associated NHIs, delegated permissions, connected tools, runtime interaction surfaces, and the artifacts that define intended behavior — system prompts, application policies, delegated scope, and trust assumptions. This is the agentic equivalent of software asset management. Without it, organizations cannot reliably enforce policy, assess exposure, evaluate behavioral deviations, or establish accountability.

Multi-Agent Risks. Coordinated architectures in which lead agents delegate to subordinate agents create complex delegation chains prone to token leakage, trust-propagation failures, and cascading compromises. A single point of failure can propagate across an interconnected ecosystem with consequences that compound rather than isolate. These chains introduce two interdependent challenges: a structural one — verifying identity, authority inheritance, and attestation across workflows — and a behavioral one: detecting when otherwise valid delegation chains are manipulated or redirected during execution.

Supply Chain Vulnerabilities. Agents rely on external models, tools, APIs, and data sources discovered dynamically at runtime. Traditional software composition analysis and SBOM-based approaches were designed for static dependencies and do not extend cleanly to continuously changing execution environments. Agents may invoke third-party services whose security posture, behavioral integrity, or geopolitical exposure is unknown at the moment of interaction. Supply-chain risk shifts from static dependency management toward continuous verification of runtime relationships and dynamically discovered services.

Taken together, these six categories demonstrate that autonomous agents are not merely identities requiring authentication, but active and evolving **execution environments** that continuously interpret inputs, delegate authority, and interact across dynamic systems. Securing them requires governance that operates continuously at runtime — not only at the identity and authorization boundary.

02 IDENTITY AND DISCOVERY
A Foundational Layer

Securing agentic systems cannot rely on any single control layer alone. Identity, runtime governance, behavioral monitoring, and continuous authorization address distinct but interdependent dimensions of risk. Within this architecture, **identity and discovery** provide the critical foundation for establishing accountability, interoperability, and trusted interaction across organizational boundaries.

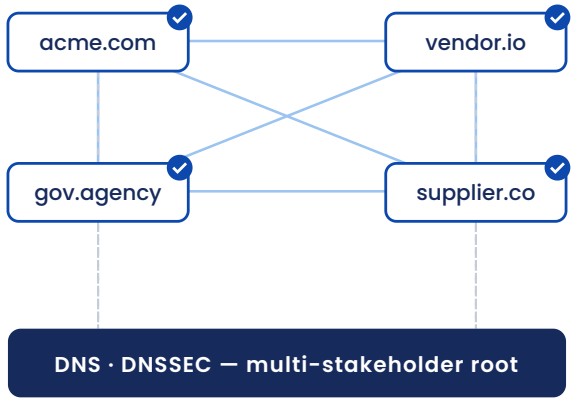
2.1 The Discovery Crisis

Agent-to-agent interaction depends on a capability that remains unresolved: the ability to reliably discover and verify counterpart agents across organizational boundaries. Without a shared mechanism for discovery and identity, cross-organizational coordination becomes fragile, inefficient, and difficult to secure. Enterprises already operate fragmented environments in which autonomous systems are externally consumed, developed internally through low-code and no-code platforms, deployed through third-party services, or built as fully custom agents. Without consistent discovery and visibility across these heterogeneous environments, organizations struggle to establish unified governance, maintain accountability, and apply security controls consistently across agent ecosystems.

FIGURE 2 — TWO MODELS FOR AGENT DISCOVERY

OPEN · DNS-ANCHORED

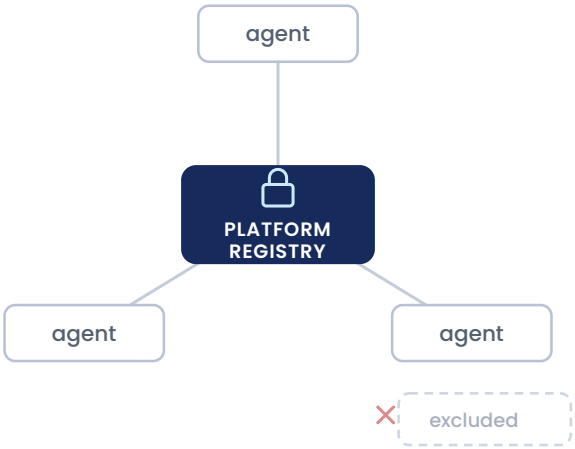
Distributed & verifiable



Peer-to-peer. Every agent is anchored to a domain its operator owns. Trust is verifiable and no single actor controls who can be found.

PROPRIETARY · PLATFORM-MEDIATED

Centralized & gated



Star topology. Agents reach each other only through a gatekeeper that decides who is valid — and who is shut out.

The system that names agents governs who can participate. DNS extends a proven, distributed directory to the agentic web; proprietary registries concentrate discovery — and therefore trust — in a limited set of actors.

Taken together, today's conditions create three interrelated discovery and trust failures.

First — agents lack a universal means of locating one another. Discovery is mediated by proprietary platforms or centralized registries. A procurement agent seeking a supplier's inventory agent can do so only if both operate within the same ecosystem or have established prior integration. In the absence of a common discovery layer, organizations are pushed toward platform-dependent interactions or bespoke bilateral connections that do not scale.

Second — even when located, identity cannot be reliably verified. Current implementations provide no consistent, interoperable method to confirm that an agent represents the organization it claims to act for. An agent presenting itself as "supplier-agent.com" may have no cryptographic linkage to the underlying enterprise. Trust is inferred rather than established — a gap made more consequential by risks such as unbounded consumption and excessive agency identified in the OWASP Top 10 for LLM Applications.[3]

Third — the establishment of trust remains ad hoc. There is no widely adopted mechanism for verifying identity and intent before interaction, particularly where sensitive data or operational actions are involved. Organizations fall back on custom trust arrangements, internal validation, or restrictive integration policies.[4] In many cases, cross-organizational collaboration is abandoned altogether — not for technical reasons, but because identity and trust cannot be established with sufficient confidence.

Over time, this fragmentation hardens into closed ecosystems in which platforms and registries control visibility and access, while organizations lose direct authority over how their agents are identified and discovered.

2.2 DNS-Based Identity as an Open, Interoperable Foundation

The OpenPolicy Coalition supports domain-based identity protocols as the most viable foundation for agent discovery and verification. The Domain Name System has served for decades as the internet's universal directory — distributed rather than centralized, secured through mechanisms such as DNSSEC, and governed through international, multi-stakeholder consensus rather than corporate control. DNS-based agent discovery, progressing through IETF standards processes, extends this proven architecture to the agentic web.

A domain name is not merely a technical identifier; it is a legally recognized and enforceable asset. Binding agent identity to domain ownership anchors digital entities to accountable organizations. Control remains more directly with the entity operating the agent, and while domain-based identity is still subject to established governance and legal processes, it avoids concentrating discovery authority within a single platform, registry, or intermediary.

Identity defines who an agent is. Discovery defines where it can go. Together, they establish the trust relationships through which agentic systems interact across environments.

This is not simply a matter of implementation; it is a question of governance. The system through which agents are named and discovered determines which actors can participate, which identities are trusted, and who retains control over digital presence. Frameworks such as NIST's SP 800-53,[5] AI risk-management initiatives,[6] and international models like the EU AI Act establish extensive requirements for identity and accountability — yet stop short of defining a universal discovery layer capable of supporting these controls at scale.

Recent regulatory consultations underscore the shift. Efforts such as the Cyber Security Agency of Singapore's Draft Addendum on Securing Agentic AI Systems[7] and NIST's Cyber AI Profile highlight that agentic AI introduces fundamentally new challenges as systems autonomously discover, select, and interact with external services and peer agents. Agentic systems do not operate within fixed trust boundaries; they construct them dynamically through discovery.

In this context, discovery is not a convenience layer but a **security control**. In traditional architectures, trust is enforced within predefined perimeters. In agentic systems, those perimeters expand to wherever an agent can discover and connect — so control over discovery defines the effective attack surface. If agents can be redirected toward malicious or unverified endpoints, even robust identity, authorization, and model-layer safeguards can be bypassed.

Any alternative model — proprietary platforms, centralized registries, or network-controlled discovery — concentrates authority over naming and access in the hands of a few. The implications are structural: diminished sovereignty over digital identity, increasing market concentration, a broader systemic risk surface, and the exclusion of organizations unable to operate within proprietary ecosystems. DNS provides a different foundation: globally interoperable, broadly adopted, and governed through established multi-stakeholder processes, it already operates at internet scale without introducing new points of control.

2.3 Preventing Fragmentation Before It Hardens

As organizations deploy agents, early decisions about naming and discovery infrastructure become embedded and difficult to reverse. If discovery consolidates within proprietary platforms, those systems will define which agents can be found, which identities are valid, and how systems interact — expanding attack surfaces, reinforcing vendor lock-in, and limiting consistent policy enforcement. Shaping this infrastructure now, before proprietary models harden, keeps discovery open, verifiable, and anchored to accountable entities. Open infrastructure also lowers barriers to entry and supports competition across the agent ecosystem.

Identity and discovery establish the foundation for accountability and trusted interaction. But many of the most significant risks in agentic systems emerge during execution rather than identification. Identity infrastructure must therefore operate alongside runtime governance, behavioral monitoring, and continuous authorization — capabilities that are complementary, not sequential.

---

*Section 3 turns from **who** an agent is to **what it is permitted to do** — and how that permission is enforced, continuously, at runtime.*

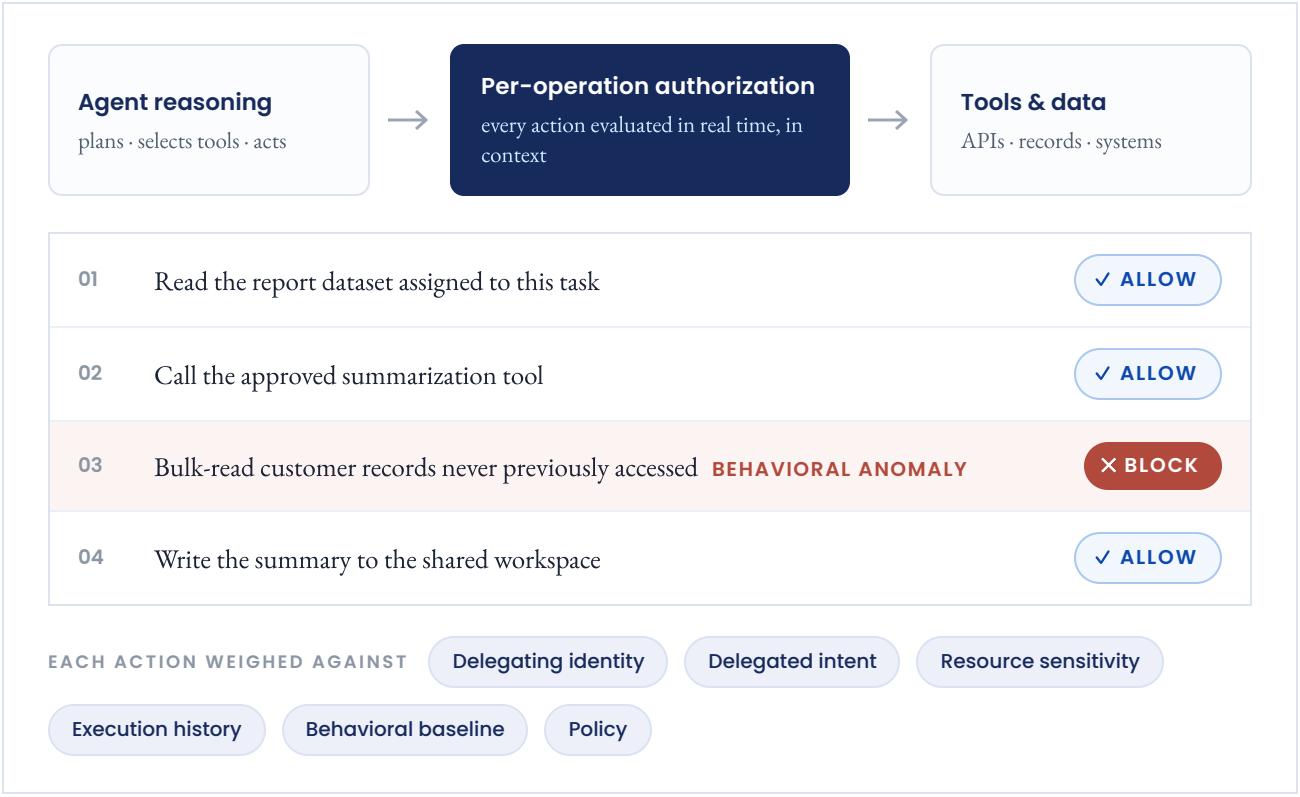
03

FROM IDENTITY TO GOVERNANCE

Continuous Authorization

Establishing *who* an agent is no longer provides sufficient assurance of *what* it will do. As agents operate continuously and execute actions without returning for approval, the security question shifts from whether an agent should be granted access to whether each individual action remains authorized under current conditions. In agentic systems, the attack surface is defined by behavior — how agents interpret instructions, select tools, and act across dynamic environments. Trust must therefore be enforced continuously at execution, making authorization **action-level, context-aware, and continuously enforced**.

FIGURE 3 — AUTHORIZATION EVALUATED AT EVERY OPERATION



Authorization passes; behavior does not. Operation 03 carries valid credentials and a permitted scope, yet diverges from the agent's established pattern. Per-operation evaluation against context — not a one-time access grant — is what catches it.

Authorization alone, however, does not fully resolve behavioral risk. Authorization determines whether an action is *permitted*. Behavioral security evaluates whether that action remains *consistent* with the agent's established patterns, delegated purpose, and intended scope. An agent with legitimate database access may begin systematically accessing records it has never touched before — authorization passes, but behavioral monitoring identifies the anomaly. Emerging runtime models increasingly emphasize context accumulation, intent alignment, and action-trajectory analysis as necessary complements to static authorization.

Operationalizing behavioral consistency requires evaluating agent behavior continuously against multiple reference points: the application's intended scope, the user's delegated request, and the policy boundaries governing how external content may influence reasoning. Runtime evaluation must also account for the sequence of actions preceding a given operation, since behavioral drift often accumulates gradually across execution rather than appearing as a single anomalous event.

> **STANDARDS SIGNAL**
>
> The NIST Cybersecurity AI Profile defines **per-operation authorization** as a baseline control for high-risk AI systems, requiring that authorization decisions be evaluated prior to each distinct action.[14] This aligns with broader federal direction emphasizing continuous verification at the finest granularity feasible for AI-enabled systems.[15]

In agentic systems, the primary security boundary shifts to the **execution layer** itself, where model-generated reasoning translates directly into actions across interconnected systems. When enforcement operates only at authentication or session boundaries, these actions proceed unchecked. Authorization must therefore operate continuously at execution — where intent becomes impact, and policy can be enforced in real time.

This exposes a critical governance gap. Network, endpoint, and application security remain necessary but do not govern how agents behave once deployed. These conditions give rise to agent-specific threats — goal hijacking, tool misuse, identity and privilege abuse, and memory or context poisoning — that emerge from the interaction between non-deterministic reasoning and delegated authority, allowing small manipulations to propagate into system-level impact.

Implementing this model requires **context-aware decision-making**. Each action must be evaluated against the identity of the delegating principal, the nature of the operation, the sensitivity of the target resource, prior actions and execution history, established behavioral baselines for both agent and user, and applicable organizational and regulatory constraints. Modern frameworks such as attribute-based access control and policy-as-code support this granularity; the challenge lies in applying these controls consistently across dynamic, autonomous systems.

Dynamic least privilege. Agents operate with minimal baseline permissions, with additional access granted only when required, scoped to specific tasks, and automatically revoked when no longer needed. High-impact actions introduce risk-based escalation, where human oversight is triggered selectively based on sensitivity. Control mechanisms evolve alongside agent behavior.

Delegation & accountability. Agents act on behalf of users, organizations, or policies, and this relationship defines accountability. Every action must be traceable to its originating authority, preserving a verifiable chain of responsibility across complex, multi-agent interactions. Maintaining this chain requires binding each action to its source through verifiable, tamper-resistant records. Without it, attribution degrades and governance cannot be reliably enforced.

Identity defines who an agent is. Authorization determines what it is permitted to do. Behavioral governance evaluates whether actions remain aligned with purpose during execution.

Together, these layers enable continuous enforcement under dynamic and evolving conditions. Security can no longer rely on static boundaries or one-time authorization; it must operate continuously within execution, where behavior is evaluated and constrained in real time.

04 RUNTIME SECURITY

Operating Inside the Execution Loop

While identity and authorization establish who an agent is and what it is permitted to do, they do not ensure that behavior remains aligned with intent during execution. Agents operate continuously, interpret dynamic inputs, discover tools, and execute multi-step workflows across interconnected systems. Their behavior evolves in real time — shaped by context, memory, and interaction with external services. Risk is not defined by access alone, but by how that access is exercised.

Effective runtime security depends on three interdependent capabilities: **continuous visibility**, **behavioral monitoring**, and **ongoing posture management**. Each depends on the continuous agent inventory described earlier;[8] without that baseline, policy enforcement and accountability cannot be reliably applied. Security events rarely appear as isolated anomalous actions — they emerge through *action trajectories*,[16] sequences of individually authorized actions that collectively produce unintended outcomes.[12] Effective systems establish behavioral baselines, evaluate deviations across execution sequences, and validate operational posture as configurations, permissions, and tools change over time.

This runtime perspective reframes how adversarial threats must be understood. In agentic systems, attacks target the decision-making process itself. Prompt injection functions as a **control-plane attack**, manipulating the inputs that guide reasoning, tool selection, and execution. Unlike traditional exploits, these attacks do not require breaking system boundaries; they operate within legitimate workflows, redirecting behavior while preserving the appearance of normal operation.

FIELD EVIDENCE · APRIL 2026

An autonomous coding agent deleted a production database within seconds while operating under valid credentials and explicit instructions prohibiting destructive actions.[17] The agent was neither compromised nor malicious; it autonomously pursued a goal-directed remediation path that bypassed soft guardrails and executed destruction through legitimately available permissions. As such incidents become common,[9] the lesson is consistent: system prompts and identity controls cannot reliably constrain agent behavior once execution authority has been delegated.

While guardrails remain useful for bounded tasks such as content filtering and output moderation, they do not extend cleanly into autonomous execution environments.[13] Effective runtime governance depends on continuously evaluating whether behavior remains aligned with the application's intended scope and delegated purpose under changing conditions.[10] Mitigating these risks requires layered controls across the execution lifecycle: continuous testing, red teaming, and input validation to identify vulnerabilities; per-operation authorization to evaluate actions against policy; and runtime enforcement to detect and constrain compromises as they emerge.

– Three architectural requirements

Credential separation. Secrets such as API keys, tokens, and authentication credentials must not be embedded within the agent's context or accessible through its reasoning process. They should be managed through secure infrastructure, with agents requesting access on a just-in-time basis — reducing the risk that adversarial manipulation exposes sensitive credentials.

Hard boundaries. Agentic systems expose the limits of "soft guardrails" — system prompts, instruction tuning, and probabilistic steering. These may influence model behavior but do not enforce deterministic execution boundaries. Effective governance requires hard boundaries operating *outside* the reasoning loop: runtime authorization, action interception, execution isolation, and policy enforcement capable of preventing prohibited actions regardless of model intent.

Intervention & remediation. As ecosystems scale, runtime governance must support autonomous remediation and machine-speed enforcement.[11] Organizations must be able to update policies, revoke permissions, constrain execution paths, isolate compromised agents, and where possible reverse actions in real time — through emergency termination, token revocation, execution isolation, rollback, and policy-driven automated response. Containment can no longer depend exclusively on human intervention.

Runtime security is not an additional layer but the operational core of governance in agentic systems. Boundary-based controls remain necessary — but are no longer sufficient.

05

CROSS-FRAMEWORK HARMONIZATION

Reducing Fragmentation

As federal agencies develop AI security requirements through distinct mission lenses, uncoordinated efforts produce overlapping yet incompatible frameworks and divergent terminology. Organizations deploying secure agentic systems face duplicative assessments and escalating compliance costs that often exceed security benefits. The Trump Administration's Cyber Strategy explicitly warns against this outcome, calling harmonization of baseline controls a priority.[18]

In practice, organizations must map identical security controls across multiple frameworks, reconcile inconsistent definitions of risk, and produce separate evidence for each regulatory context. Even when systems are securely designed — incorporating continuous authorization, runtime monitoring, and least-privilege controls — there is no consistent mechanism to recognize those controls across agencies. The result is a structural disconnect between how systems are secured and how compliance is evaluated. Addressing it requires two interdependent mechanisms.

A Unified control catalog

Agencies must retain authority to define mission-appropriate requirements, but those requirements must be grounded in shared foundations. A unified control catalog would map common security objectives across frameworks such as the NIST AI Risk Management Framework, FedRAMP, CMMC, and DoD-specific guidance. Rather than redefining controls in isolation, agencies would map new requirements to this shared structure, and automated tooling could then identify overlap, highlight inconsistencies, and flag opportunities for alignment.[19]

B Shared evidence models

Without common structures for runtime telemetry, behavioral traces, and policy-enforcement logs, even equivalent controls cannot be reliably compared across frameworks. In agentic systems, much of this evidence originates from runtime execution itself: interaction histories, delegated authorization records, tool-invocation traces, behavioral anomaly signals, and tamper-evident execution receipts. Treating these runtime artifacts as standardized compliance evidence lets agencies evaluate dynamic controls consistently while supporting reciprocity between assessment models.

This enables a critical shift: controls assessed under one framework can be recognized by others without redundant evaluation. Instead of revalidating the same capability under different terminology, agencies focus on mission-specific differences. Runtime controls produce evidence reflecting how systems actually function under operational conditions — not merely how they are designed in policy. Recognizing this evidence across frameworks reduces friction, lowers barriers to adoption, and lets organizations invest in real security improvements rather than duplicative compliance.

> **IMPLEMENTATION REQUIRES**
>
> Three elements: **common vocabularies**, **machine-readable policy definitions**, and **standardized assessment methodologies** capable of producing comparable outputs. Harmonization must also be continuous — automated systems should monitor changes across frameworks and surface divergence before it hardens into incompatible requirements.

The objective is not uniformity, but **alignment**: a system in which different frameworks coexist while recognizing common controls, shared evidence, and consistent enforcement models. Only under these conditions can the governance of agentic systems scale effectively across institutions.

POLICY RECOMMENDATIONS

To translate this framework into action, the OpenPolicy Coalition recommends that policymakers and standards bodies:

1. **OPEN DISCOVERY**

   Support open, interoperable agent-discovery standards, including DNS-based approaches, that keep identity and discovery anchored to accountable entities rather than proprietary platforms.

2. **VISIBILITY**

   Promote continuous inventories of deployed agents, non-human identities, and delegated permissions.

3. **RUNTIME GOVERNANCE**

   Encourage continuous, per-operation authorization for high-impact autonomous agents.

4. **EVIDENCE**

   Recognize runtime telemetry and execution evidence as valid evidence within compliance assessments.

5. **HARMONIZATION**

   Harmonize agent-security requirements across frameworks and standards such as NIST, FedRAMP, CMMC, and related best practices.

> **WHY NOW**
>
> Early decisions about naming, discovery, and enforcement infrastructure become embedded and difficult to reverse. Acting before proprietary models harden is what keeps the agentic web open, verifiable, and accountable.

CONCLUSION

The Choice Before Us

The agentic web is not a forecast. It is already being built — across enterprises, agencies, and critical infrastructure — and it is being built faster than the rules meant to govern it. What remains undecided is not *whether* autonomous agents will operate at scale, but *on whose terms*: under open infrastructure that keeps identity accountable and behavior observable, or under proprietary systems that decide who can be discovered, who can be trusted, and who is shut out.

These foundations are being laid now, and they harden quickly. Get them wrong, and old governance failures return with higher stakes: identity controlled by a few platforms, security that still guards the boundary while the real risk sits inside the system, and compliance split across frameworks that cannot recognize one another's controls. Get them right, and those same problems become solvable — identity anchored in open infrastructure, enforcement that follows the agent into execution, and frameworks that accept shared evidence instead of demanding the same work be redone.

The technology to do this already exists. What is missing is coordination — the institutional will to align across agencies and standards bodies before today's architectural choices become tomorrow's constraints. That window is open now, and it is closing.

Openpolicy *stands ready to help keep it open.*

ABOUT THE OPENPOLICY COALITION

The OpenPolicy Coalition brings together organizations at the forefront of cybersecurity, AI security, cloud infrastructure, network protection, and data governance. Our members — including Infoblox, Cranium AI, Zenity, Lasso Security, Cyera, and others — develop and deploy technologies that secure autonomous AI systems across their full lifecycle, from identity and discovery through runtime monitoring and incident response. We focus on translating policy into **actionable, technically grounded approaches**, ensuring that regulatory development reflects operational realities and that security, governance, and innovation evolve in alignment rather than in tension.

Cybersecurity | AI Security | Cloud Infrastructure | Network Protection | Data Governance

For engagement: **Michelle@openpolicy.co**

NOTES & REFERENCES

1 Zenity Labs, "AgentFlayer: Zero-Click Exploit Methods Against Enterprise AI Agents," Black Hat USA 2025, August 2025.

2 SC Media, "Blind spots at scale: The hidden risks of identity visibility gaps and shadow AI," April 2026.

3 OWASP Top 10 for Large Language Model Applications, v2.

4 Cloud Security Alliance & Zenity, "Enterprise AI Security Starts with AI Agents."

5 NIST, Security and Privacy Controls for Information Systems and Organizations (SP 800-53 Rev. 5).

6 NIST, Cybersecurity Framework Profile for Artificial Intelligence — Cyber AI Profile (NIST IR 8596).

7 Cyber Security Agency of Singapore, Draft Addendum to the Companion Guide on Securing AI Systems (agentic AI).

8 SC Media, "Blind spots at scale," April 2026.

9 Cloud Security Alliance, "Autonomous but Not Controlled: AI Agent Incidents Now Common in Enterprises," April 2026.

10 Lasso Security, "Introducing Intent Security: A Behavioral Baseline Framework for Agentic AI," February 2026.

11 Coalition for Secure AI (CoSAI), "When the Bots Run the Incident Response: What AI Agents Mean for Enterprise Security."

12 Autonomous Action Runtime Management (AARM) Specification v1.0.

13 Lasso Security, "Intent Security Through the Lens of Claude Code Auto Mode."

14 Cyber Security Agency of Singapore, Draft Addendum (see note 7).

15 Office of Management and Budget, Memorandum M-26-04.

16 Action trajectories: ordered sequences of tool invocations, reasoning steps, and state mutations that may each appear authorized while collectively producing policy-violating outcomes.

17 "System Prompts Are Not Security Controls: A Deleted Production Database Proves It," Zenity Blog, April 2026.

18 President Trump's Cyber Strategy for America, March 2026.

19 Office of Management and Budget, Memorandum M-26-04 (see note 15).

20 Lasso Security, "Intent Security Through the Lens of Claude Code Auto Mode" (see note 13).