WHITEPAPER · GOVERNANCE FRAMEWORK

AI Identity Governance Framework

A practical framework for discovering, owning, and governing autonomous AI agents and non-human identities under Zero Trust principles.

Contents

01 Executive Summary

02 Why AI Agents Need Their Own Governance Model

03 Framework Overview: Discover, Govern, Enforce, Remediate

04 Ownership: The Foundation of Accountability

05 Zero Trust Enforcement for Machine Identities

06 Human-in-the-Loop: Governing Autonomy Without Freezing It

07 Compliance Evidence as a System Property

08 Adoption Model: From Pilot to Production

09 Implementation Considerations

10 Summary

01 Executive Summary

This whitepaper defines a practical framework for governing the identities of autonomous AI agents and other non-human identities (NHIs) inside the enterprise — treating them as first-class identities subject to the same discipline as human accounts, but governed through mechanisms suited to their scale and speed.

The framework rests on a single governance loop applied continuously to every agent: **Discover** the identity, **Govern** it by assigning accountable ownership and building an identity graph, **Enforce** explainable Zero Trust policy at the point of action, and **Remediate** risk automatically where safe, escalating to a human where not. Every step writes tamper-evident evidence before it acts, so the resulting audit trail can support compliance obligations under SOC 2, ISO 27001, NIST AI RMF, the EU AI Act, and PCI-DSS.

> **WHO SHOULD READ THIS**
>
> IAM and identity governance architects, security leaders responsible for AI risk, and compliance officers building an evidence model for automated decision-making.

02 Why AI Agents Need Their Own Governance Model

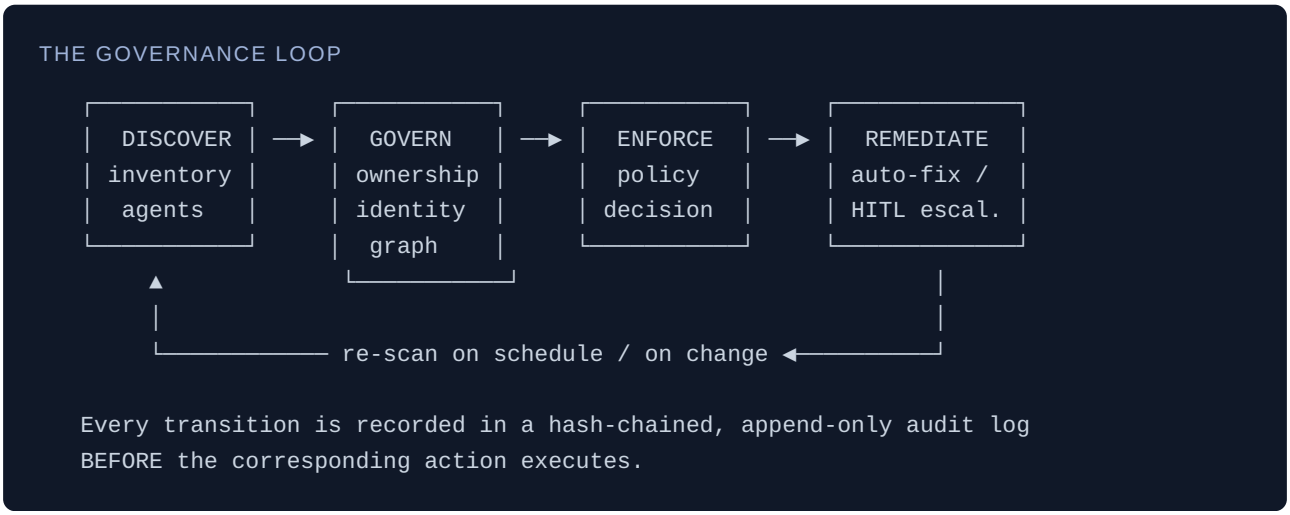
Identity governance and administration (IGA) programs were built around a slow, predictable lifecycle: a person joins, changes roles, and eventually leaves. AI agents and NHIs break every one of those assumptions.

DIMENSION	HUMAN IDENTITY	AI AGENT / NHI
Creation	Formal onboarding, HR-triggered	Created ad hoc by any engineer, script, or pipeline
Volume & velocity	Grows with headcount	Grows with code deploys; can outnumber humans many-to-one
Ownership	Manager of record	Frequently unassigned or lost when the creator changes teams
Credentials	Rotated on a defined cadence, MFA-backed	Often long-lived API keys and tokens with no rotation policy
Access reviews	Periodic, role-based	Rarely reviewed; permissions accumulate silently
Action visibility	Logged via SSO/IdP	Frequently invisible outside application-level logs, if logged at all

A governance model for AI agents has to assume identities appear and disappear at code-deploy speed, that ownership must be inferred rather than declared, and that policy has to be evaluated at the moment of action rather than at a quarterly access review.

03 Framework Overview: Discover · Govern · Enforce · Remediate

The framework is organized as four capabilities operating as a continuous loop rather than sequential project phases — every agent moves through all four on an ongoing basis.



Capability 1 — Discover

Continuous, connector-based inventory of every AI agent and NHI: source control repositories, container runtimes, running processes, and secret-store entries. Discovery should be idempotent — safe to re-run continuously — and framework-aware, recognizing common agent patterns (LangChain, LangGraph, MCP-based agents, CrewAI, RPA bots, and custom automation) so that new categories of agent don't silently evade inventory.

Capability 2 — Govern

Ownership inference from converging, weighted signals rather than a single declared field, because declared ownership data for machine identities is frequently missing or stale. The output is an identity graph — owner → agent → secret — that makes orphaned agents (no confident owner) visible rather than hidden, plus a behavioral baseline for each agent to support later anomaly detection.

Capability 3 — Enforce

Policy-as-code evaluation of every authorization request, combining attribute-based, role-based, and policy-based access control (ABAC/RBAC/PBAC). Risk is scored through a transparent, additive model rather than a black-box classifier, and every decision — allow, needs-approval, or deny — carries a human-readable rationale.

Capability 4 — Remediate

Low-blast-radius issues (a non-production secret nearing rotation, a missing workload identity) are resolved automatically. High-blast-radius actions (production credential rotation, terminating a container, revoking a secret, isolating an agent from the network) are routed to a tiered human approval chain before executing.

04 Ownership: The Foundation of Accountability

Zero Trust for machine identities starts from a simple premise: an agent without an accountable owner cannot be trusted, regardless of its risk score.

A weighted inference model

SIGNAL	SUGGESTED WEIGHT	RATIONALE
Source-control history	35%	Commit authorship is the strongest available proxy for who built and maintains the agent
CODEOWNERS / declared ownership	25%	Explicit organizational intent, where it exists
Container / runtime labels	20%	Deployment metadata frequently encodes the owning team
Behavioral / semantic similarity	10%	New or ambiguous agents can be associated with known-owned clusters
Secret-store path conventions	10%	Naming and path conventions often encode team or service ownership

Decision thresholds

- **≥ 0.85 confidence** — auto-assign the inferred owner.
- **0.60–0.84 confidence** — route to a human for confirmation before the assignment becomes authoritative.
- **< 0.60 confidence** — flag the agent as orphaned and escalate; orphaned status should never be silently suppressed, since it is often the single strongest risk signal in the environment.

> **DESIGN PRINCIPLE**
>
> Ownership is a first-class governance artifact, not metadata. Every downstream policy decision — including whether an agent's actions require escalation — should be able to reference who is accountable for it.

05 Zero Trust Enforcement for Machine Identities

Enforcement translates ownership and risk into an authorization decision at the moment an agent attempts to act — not at a periodic review.

Default-deny as the starting posture

An agent should be required to present three things before any action is allowed: a valid workload identity (a cryptographic credential, not just a static API key), a confirmed owner, and an acceptable computed risk score. Absent any of the three, the default decision is deny.

The three-way decision

DECISION	TRIGGER	EFFECT
Allow	Owned, identified, risk below the approval band	Action proceeds immediately
Needs-approval	Risk within an escalation band (e.g. 0.70–0.90)	Action pauses pending human sign-off
Deny	Orphaned, quarantined, or risk above the critical threshold	Action is blocked outright

Why explainability matters here specifically

Under frameworks like the EU AI Act, an organization must be able to explain automated decisions that affect systems and data, not just produce a score. An additive, weighted risk model — where each contributing factor and its weight is visible — can be explained to an auditor in a way a black-box model cannot. This is a deliberate architectural trade-off: transparency over marginal predictive gains from opaque models.

06 Human-in-the-Loop: Governing Autonomy Without Freezing It

Full automation of every remediation is neither safe nor necessary. The framework routes only high-blast-radius, hard-to-reverse actions to a human, while resolving routine, low-risk issues automatically.

TIER	TYPICAL ROLE	EXAMPLE ACTIONS GATED
T1	Security Engineer	Production credential rotation; stopping/killing a container
T2	SOC Lead	Revoking a secret; isolating an agent's network access
T3	Platform Admin	Actions with organization-wide blast radius

Automatic, ungated remediation is reserved for reversible, low-impact actions: rotating non-production secrets, issuing a workload identity to a newly-owned agent, opening a tracking issue, or sending a notification. Everything else waits for a human — and every decision, automatic or human, is recorded as evidence before it executes.

07 Compliance Evidence as a System Property

A governance framework only satisfies auditors and regulators if its evidence trail is trustworthy by construction, not by policy.

- **Write-before-act.** The audit record for a decision is committed before the corresponding action runs, so there is no window where an action can occur without a matching record.
- **Tamper-evident chaining.** Each record is cryptographically linked to the one before it, so retroactive modification of history is detectable.
- **Append-only storage.** No update or delete path exists for audit records; the store is architected as a write-ahead log.
- **Framework mapping.** Governance events are tagged against the control objectives of SOC 2, ISO 27001, NIST AI RMF, EU AI Act, and PCI-DSS so evidence can be pulled by framework on demand.

> **SCOPE NOTE**
>
> Framework mapping identifies where governance evidence is relevant to a standard's control objectives. It does not constitute certification, and organizations remain responsible for their own compliance assessment.

08 Adoption Model: From Pilot to Production

The framework is designed to be adopted incrementally, starting with visibility and expanding into enforcement and remediation as confidence in the data grows.

STAGE	GOAL	WHAT'S LIVE
1. Visibility	Know what exists	Discovery only; no enforcement actions taken
2. Accountability	Know who owns it	Ownership inference active; orphans surfaced to security/ IAM teams
3. Enforcement	Control what agents can do	Policy evaluation live in monitor mode, then enforcing mode
4. Remediation	Close the loop automatically	Auto-remediation for low-risk findings; HITL for high-risk actions

Moving fast to Stage 3 or 4 before ownership data is trustworthy risks false-positive denials that erode trust in the system. Most organizations should expect to spend real time in Stages 1–2 tuning ownership signal weights against their own environment before enabling enforcement broadly.

09 Implementation Considerations

A few practical points matter regardless of which platform implements this framework.

- **Connector coverage drives everything.** An identity that isn't discovered can't be governed; prioritize connectors for the systems that actually spawn agents in your environment (source control, container orchestration, secret managers, RPA platforms).
- **Secrets should never be duplicated into the governance store.** Only path references to a secret manager should be persisted; raw values remain in the source of truth.
- **Policy should be version-controlled with the application**, not maintained as a separate manual process, so authorization logic changes go through the same review discipline as code.
- **Plan for the Rego/fallback-evaluator drift problem** if you maintain more than one policy evaluation path (e.g., a full OPA deployment and a lightweight fallback) — keep them behind shared unit tests so decisions can't silently diverge.

10 Summary

Governing AI agents and non-human identities requires treating them as identities in their own right — discoverable, ownable, policy-governed, and auditable — rather than retrofitting a human-identity program to cover them.

The Discover → Govern → Enforce → Remediate loop, backed by explainable risk scoring, default-deny enforcement, tiered human approval, and write-before-act audit evidence, gives security, IAM, and compliance teams a concrete framework for closing the machine-identity blind spot without freezing the pace of AI adoption.

> **NEXT**
>
> See the companion whitepapers — *Enterprise AI Security Architecture* for the technical implementation view, and *AI Agent Governance Best Practices* for an operational playbook.