WHITEPAPER · BEST PRACTICES

AI Agent Governance Best Practices

An operational playbook for rolling out AI agent and non-human identity governance — sequencing, checklists, and common failure modes to avoid.

Contents

01 Executive Summary

This whitepaper is a practical, operational playbook for teams rolling out governance over AI agents and non-human identities (NHIs). It distills the Discover → Govern → Enforce → Remediate model into concrete practices, checklists, and sequencing guidance for security, IAM, platform, and compliance teams.

The central recommendation is sequencing: build visibility and ownership data before turning on enforcement, and turn on enforcement in monitor mode before making it authoritative. Most governance rollouts fail not because the policy model is wrong, but because enforcement is switched on before the underlying ownership and risk data is trustworthy.

02 Best Practice: Start With Discovery, Not Policy

The most common mistake in AI agent governance programs is writing policy before knowing what exists.

- **Inventory before you enforce.** Run discovery across every connector you have (source control, container runtime, process listings, secret stores) before writing a single policy rule.
- **Treat discovery as continuous, not a one-time project.** Agents are created at code-deploy speed; a quarterly inventory will always be stale. Schedule continuous or event-triggered re-scans.
- **Make discovery idempotent.** Deduplicate by a stable content or identity hash so repeated scans don't create duplicate inventory records.
- **Expect surprises.** Most organizations running a first discovery scan find agents they did not know existed — orphaned scripts, forgotten RPA bots, agents built by a team that has since disbanded. Budget time to investigate these rather than immediately auto-remediating them.

> **CHECKLIST — DISCOVERY READINESS**
>
> ☐ Connectors configured for every system that can spawn an agent (repos, containers, processes, secret stores)
> ☐ Discovery scan runs on a schedule, not just on demand
> ☐ Deduplication verified against a stable identity hash
> ☐ A named owner reviews first-scan results before any policy is enabled

03 Best Practice: Make Ownership Non-Negotiable

An agent without an accountable human owner is the single highest-risk pattern in most environments — and the easiest to fix once it's visible.

- **Infer ownership from multiple signals** — commit history, CODEOWNERS files, container labels, secret-store path conventions — rather than relying on a single declared field, since declared ownership for machine identities is frequently missing or wrong.
- **Set explicit confidence thresholds** and route anything below your organization's comfort level to human review rather than silently accepting a low-confidence guess.
- **Never hide orphaned agents.** An agent with no confident owner should be surfaced prominently, not defaulted to a generic team or suppressed from view.
- **Assign a remediation SLA for orphans.** Set a target (e.g. 5 business days) for a human to either claim ownership of an orphaned agent or approve its decommissioning.

> **COMMON FAILURE MODE**
>
> Teams sometimes respond to a large number of orphaned agents by lowering the confidence threshold until most agents auto-assign. This defeats the purpose — a wrong owner is often worse than a visible gap, because it creates false confidence that accountability exists.

04 Best Practice: Roll Out Enforcement in Stages

Policy enforcement should never be switched from off to fully authoritative in one step.

STAGE	WHAT HAPPENS	EXIT CRITERIA
Monitor mode	Policy evaluates every request and logs the decision it would have made, without blocking anything	Low, explainable false-positive rate on deny/needs-approval decisions over a representative period
Soft enforcement	Needs-approval decisions are enforced; deny decisions still only logged	Approval queue volume is manageable and approvers trust the escalations they're seeing
Full enforcement	Allow/needs-approval/deny are all authoritative	Ownership and risk data have been stable and trusted through the prior two stages

Skipping straight to full enforcement is the second most common cause of governance rollout failure — it produces a wave of blocked legitimate automation before the risk model has been tuned to the organization's actual environment, which erodes trust in the system and invites teams to route around it.

05 Best Practice: Keep Risk Scoring Explainable

- **Prefer a transparent, additive risk model** over a black-box classifier, even if it means somewhat coarser scoring. Every stakeholder — the agent's owner, the approver, and eventually an auditor — needs to be able to see why a score landed where it did.
- **Document your escalation bands** (the risk ranges that trigger needs-approval versus deny) and revisit them periodically as your agent population and threat model evolve.
- **Version your policy alongside your application code.** Policy-as-code, reviewed through the same pull-request process as everything else, prevents authorization logic from drifting out of sync with the rest of the system.
- **If you run more than one policy evaluation path** (for example, a full policy-engine deployment and a lightweight local fallback for resilience), keep them behind shared tests so the two paths cannot silently produce different decisions.

06 Best Practice: Reserve Automation for Reversible Actions

The line between "safe to automate" and "requires a human" should be drawn at reversibility and blast radius, not at technical feasibility.

Good candidates for full automation

Rotating non-production secrets, issuing a workload identity to a newly-owned agent, opening a tracking issue, sending a notification — all reversible, low-impact, and easy to roll back.

Should require human approval

Production credential rotation, stopping or terminating a running container, revoking a secret, isolating an agent's network access — all hard to reverse quickly or with organization-wide blast radius.

Design a tiered approval chain

Not every gated action needs the same approver. A practical tiering pattern:

TIER	APPROVER	TYPICAL ACTIONS
T1	Security engineer	Production credential rotation, container termination
T2	SOC lead	Secret revocation, network isolation
T3	Platform admin	Organization-wide or cross-team impact

Set a timeout and an escalation path for approvals — an unattended approval queue is functionally equivalent to no enforcement at all.

07 Best Practice: Treat Audit Evidence as a Build Requirement

- **Write the audit record before the action executes**, not after — including for automatic allows. If the action fails after the audit record is written, that is recoverable; a missing audit record for an action that already happened is not.
- **Make the audit store append-only** at the architecture level (no update or delete path), not just by policy.
- **Hash-chain audit records** so tampering with history is detectable, and periodically run an independent chain-verification check rather than assuming integrity.
- **Map audit events to the compliance frameworks you actually report against** (SOC 2, ISO 27001, NIST AI RMF, EU AI Act, PCI-DSS, or others relevant to your industry) so evidence retrieval for an audit is a query, not a project.

CHECKLIST — AUDIT READINESS

☐ Every governance decision (allow, needs-approval, deny, remediation) writes a record before acting
☐ Audit store enforces append-only at the data layer
☐ Hash chain is verified on a schedule, not only when an incident is suspected
☐ Audit events are tagged to the frameworks your compliance team reports against

08 Best Practice: Secure the Secrets, Not Just the Policy

- **Never persist raw secret values** in the governance platform's own database — only path references into a dedicated secret manager.
- **Rotate secrets server-side** as part of remediation, and never return raw rotated values through an API response.
- **Treat connector and model output as untrusted input.** Anything read from a source-control system, container registry, or produced by an LLM should be validated and sanitized before it can influence a database query, a system call, or a secret-manager call.

09 Best Practice: Be Honest About Maturity

Many governance programs — and many platforms, including early deployments of this one — start with local or mock implementations of sensitive integrations (identity issuance, secret storage, cryptography) to prove the governance model cheaply before committing to production infrastructure.

- **Label reference/mock implementations clearly**, internally and to any auditor, so no one mistakes a pilot for a production-secure deployment.
- **Require a security review before promoting** authentication, cryptography, identity, or remediation code paths to touch real credentials or regulated data.
- **Design for configuration-only promotion** from local to production providers wherever possible, so hardening the deployment doesn't require rewriting the governance logic itself.

10 A Suggested Rollout Timeline

PHASE	DURATION (TYPICAL)	FOCUS
1. Discovery & ownership	2–4 weeks	Connect discovery sources, tune ownership inference, clear the orphan backlog
2. Monitor-mode enforcement	2–4 weeks	Run policy in log-only mode; tune risk scoring and escalation bands against real traffic
3. Soft enforcement	2–4 weeks	Enforce needs-approval decisions; build trust in the HITL queue and approver workflow
4. Full enforcement	Ongoing	Enforce allow/needs-approval/deny; expand connector coverage; begin remediation automation for low-risk findings

> Timelines will vary significantly with the size and complexity of the agent population; the sequencing matters more than hitting a specific calendar target.

11 Summary Checklist

- ☐ Discovery runs continuously across every system that can spawn an agent
- ☐ Ownership is inferred from multiple signals with an explicit confidence threshold, and orphans are never hidden
- ☐ Enforcement rolls out through monitor → soft → full stages, not switched on all at once
- ☐ Risk scoring is explainable and policy is version-controlled with the application
- ☐ Automation is reserved for reversible, low-blast-radius actions; everything else is tiered to a human approver
- ☐ Audit records are written before actions execute, stored append-only, and hash-chained
- ☐ Secrets are never duplicated into the governance platform; only references are stored
- ☐ Reference/mock implementations are clearly labeled and reviewed before production promotion

> **NEXT**
>
> See the companion *AI Identity Governance Framework* whitepaper for the underlying model, and *Enterprise AI Security Architecture* for the technical implementation reference.