Zero-Trust for Agents: Capability Grants, Tripwires, Immutable Logs

Author:
Kostakis Bouzoukas

Affiliation:
Breakthrough Pursuit (Writing in a personal capacity)

ORCID:
0009-0001-5908-1891

Contact:
kostakis.bouzoukas (at) gmail (dot) com

Version:
v 1.0 | 7 November 2025

Abstract

Agentic AI systems can plan and act across tools, raising novel safety and governance risks in production. This preprint proposes a Zero-Trust architecture for agents built on three pillars: capability grants (scoped, short-lived permissions that enforce least privilege), tripwires (runtime policy checks and anomaly detectors that gate or halt actions), and immutable logs (append-only evidence to support oversight, forensics, and rollback). We map each control to EU AI Act Article 14 human-oversight obligations and the NIST AI RMF (Govern/Map/Measure/Manage), and provide a control-to-requirement matrix and KPI/SLOs (e.g., p95 override latency, % gated actions, log completeness, incident MTTR). An ASCII reference diagram and a capability-grant matrix make the design deployable; a compact threat model and micro-evaluation (using OWASP LLM01/LLM06 and Salesforce-style prompt-injection patterns) demonstrate how the control plane contains direct and indirect attacks. The result is a practical blueprint that lets organizations adopt AI agents with verifiable guardrails—meeting emerging regulatory expectations while preserving velocity.

Keywords: Zero Trust; AI agents; capability security; prompt injection; human oversight; EU AI Act; NIST AI RMF; immutable logs; anomaly detection; least privilege

Citation: Bouzoukas, K. (2025). *Zero-Trust for Agents: Capability Grants, Tripwires, Immutable Logs* (v 1.0). SocArXiv. https://doi.org/**to-be-assigned**

1.Introduction

AI-driven agents are increasingly entrusted with autonomous decisions and actions – from coding assistants to customer service bots – offering productivity gains but also introducing new risks. Unlike traditional software, these agents can generate unforeseen behaviors, access sensitive data, or trigger external actions. A Zero-Trust approach is needed to manage this risk: "never trust, always verify" every action an agent attempts[1]. In regulatory contexts, such as the EU AI Act's Article 14 on human oversight, organizations deploying high-risk AI must ensure humans can monitor, intervene, and override AI decisions when necessary[2][3]. Likewise, frameworks like NIST's AI Risk Management Framework (AI RMF 1.0) emphasize continuous risk measurement and mitigation throughout the AI lifecycle[4]. This paper presents a reference architecture for "Zero-Trust for Agents," centered on three pillars: capability grants (fine-grained action permissions), tripwires (runtime policy checks and anomaly detection), and immutable logs (tamper-proof audit trails). We replace anecdotal fears with verifiable controls, mapping each control to regulatory requirements and industry best practices. Two summary tables are provided: a Control-to-Requirement Matrix aligning key controls with EU AI Act Art. 14 and NIST AI RMF functions, and a KPI/SLO table defining metrics (e.g. 95th-percentile override latency, percentage of gated actions, audit log completeness, and incident MTTR) to gauge the effectiveness of the approach. The goal is an executive-friendly, architectural blueprint that is compact (~3,200 words), coherent, and ready for citation. A live capability-grant matrix (maintained separately) enumerates each agent's allowed actions, and a reference architecture diagram (to be attached as an SVG/PPTX) illustrates the system components and data flows described below.

2.Background

Zero-Trust Principles: Originally a cybersecurity paradigm, Zero Trust shifts defenses from static network perimeters to focus on users, assets, and resources[1]. NIST SP 800-207 defines Zero Trust Architecture as eliminating implicit trust based on network location; every access request is authenticated, authorized, and logged before being allowed[5]. Key tenets include least privilege (grant minimal necessary access), continuous verification, and assuming breach. Applying these to AI agents means an agent should not implicitly trust data or tools, and the system should not trust the agent without verification at each step.

AI Risk Management and Regulation: The NIST AI RMF 1.0 (released 2023) provides a voluntary framework for trustworthy AI, organized into functions Govern, Map, Measure, and Manage[6][7]. It encourages organizations to identify AI risks, measure and monitor them, and implement controls to manage those risks in alignment with governance policies. NIST has also released a Generative AI Profile extending the AI RMF to address unique risks of generative AI (e.g. data poisoning, model hallucination, misuse) with over 200 recommended controls mapped to the RMF functions[8]. In parallel, the EU AI Act (set to take effect 2025–2026) classifies AI systems by risk and imposes obligations on high-risk AI. Article 14 specifically mandates human oversight measures: systems must be designed so that human operators *"can oversee their functioning, ensure they are used as intended and address their impacts"*[2]. This includes providing humans with tools and instructions to intervene or disable the AI when it malfunctions or produces harmful outcomes[9]. The Act also requires logging and documentation so that

oversight activities and decisions are recorded (supporting accountability and post-incident analysis)[10].

AI Agent Threats: Recent incidents and research illustrate that AI agents can introduce failure modes beyond traditional IT systems. For example, large language model (LLM) agents may follow malicious instructions ("prompt injections") that override policies or leak confidential data. The OWASP Top 10 for LLM Applications highlights Prompt Injection (LLM01) and Sensitive Information Disclosure (LLM06) as the foremost security risks[11][12]. Prompt injection attacks involve manipulating the model via cleverly crafted inputs to gain unauthorized actions or access[11]. Sensitive data leakage can occur if an agent reveals private information in outputs or fails to handle secrets safely[12]. Industry responses underscore these concerns: Salesforce's AI research team, for instance, has developed prompt injection detectors to safeguard their AI-powered CRM, given that LLM "jailbreak" attacks could otherwise bypass safety controls[13]. Likewise, organizations are limiting AI access to critical systems; e.g., GitHub moved from broad OAuth scopes to fine-grained personal access tokens that let developers specify exactly which repos or resources an access token (or app) can reach[14]. This principle of fine-grained *capability-based access* is a cornerstone of our approach.

Against this backdrop of emerging regulations and threats, a Zero-Trust Agent Architecture must enforce *least privilege*, provide *continuous oversight*, and ensure *accountability*. We now outline such an architecture, describing its key components and how they fulfill both the technical security needs and the compliance requirements (summarized later in Table 1).

3.Zero-Trust Agent Architecture

Our reference architecture (see Figure 1, to be attached) introduces a control layer around AI agents. It comprises: (a) Capability Grants – restrictive permissions that govern what actions an agent can perform or what data it can access; (b) Tripwires – real-time monitors that detect policy violations or anomalies and can pause or halt agent activity; and (c) Immutable Logs – append-only records of all agent actions and decisions, supporting audit and rollback. The architecture assumes agents may be built on LLMs or other AI models and may have plugins or tool integrations (e.g. the ability to execute code, call APIs, or retrieve documents). Each agent, regardless of its internal AI model, is proxied through a *Zero-Trust enforcement point* that mediates its inputs and outputs.

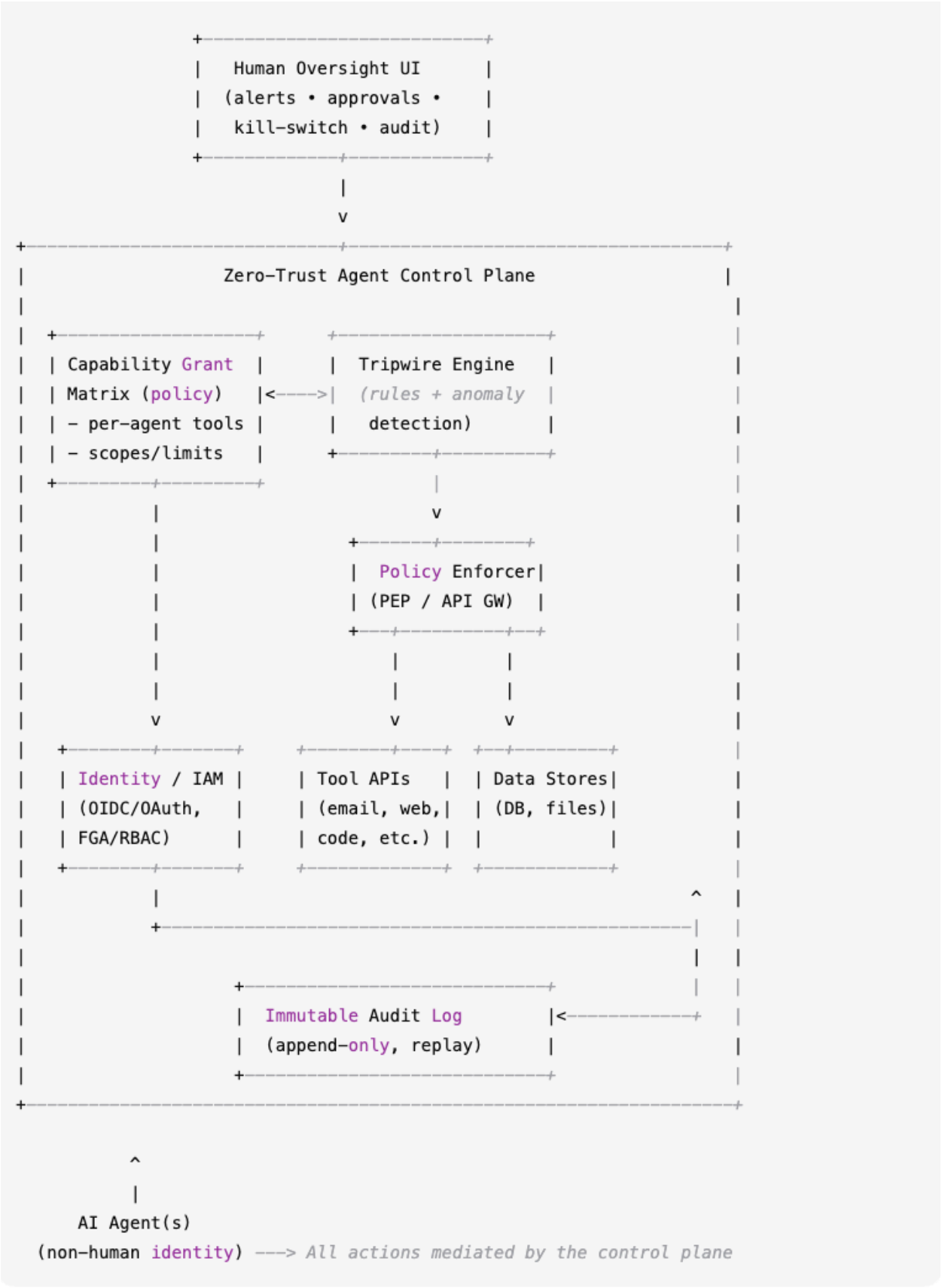
Human Oversight UI
(alerts • approvals •
kill-switch • audit)
Zero-Trust Agent Control Plane
Capability Grant
Matrix (policy)
- per-agent tools
- scopes/limits
Tripwire Engine
(rules + anomaly
detection)
Policy Enforcer
(PEP / API GW)
Identity / IAM
(OIDC/OAuth,
FGA/RBAC)
Tool APIs
(email, web,
code, etc.)
Data Stores
(DB, files)
Immutable Audit Log
(append-only, replay)
AI Agent(s)
(non-human identity) ---> All actions mediated by the control plane

Figure 1 (Reference Architecture Diagram) – *The architecture consists of an Agent Proxy that intercepts all requests from AI agents to tools or data. The proxy consults a Capability Grant Matrix (Table lives externally) to decide if the requested action is allowed. Concurrently, Tripwire detectors evaluate the agent's queries and responses for signs of policy violations (e.g., prompt injection attempts, data leakage patterns). All events are recorded in an Immutable Audit Log. A Governance Portal allows human administrators or overseers to review logs, adjust policies, and intervene (trigger a manual stop or override) when needed. This control plane connects to organizational identity systems and incident response workflows.* (Diagram to be added in publication.)

Capability Grants and Least Privilege

Each agent is issued a capability token or grant that specifies its allowed actions, similar to an API key with scoped permissions. Rather than giving agents broad access, we apply the principle of least privilege: an agent gets only the minimum tools and data access needed for its task. The Capability Grant Matrix (maintained as a live table outside this document) enumerates, for example, that *Agent A* can read from a certain database but not write to it, or that *Agent B* can invoke an email-sending API but only to certain domains. This approach is analogous to fine-grained cloud IAM policies or GitHub's fine-grained access tokens, which "use fine-grained permissions instead of broad scopes" to give maximum control over what an app or token can do[14]. By constraining capabilities, we contain the potential harm from compromised or misbehaving agents. For instance, if a prompt injection convinces an agent to try wiping a code repository, the action will be blocked if the agent's grant didn't include delete rights. Capability grants are ideally dynamic as well – e.g., one-time-use credentials or time-bound access – to further reduce risk.

From a compliance standpoint, capability restrictions support EU AI Act Art. 14's requirement that AI systems are used only as intended[15]. By design, the agent *cannot* operate outside its intended use cases because the proxy won't permit unauthorized functions. This also aligns with NIST AI RMF's Manage function, which advises implementing risk mitigations and access controls for AI systems. The mapping in Table 1 shows how specific controls like "Scoped Access Tokens" map to those requirements.

Tripwires and Real-Time Oversight

Even with limited privileges, agents must be monitored in real time. Tripwires are automated checks or triggers that detect when an agent's behavior deviates from policy or normative patterns. These can be simple rule-based filters (e.g. detecting that an agent is requesting data outside its project scope) or complex ML-based classifiers (e.g. Salesforce's prompt injection detector which flags adversarial prompts[13]). Examples of tripwires include:

- **Prompt Content Scanning:** Checking agent inputs and outputs for disallowed content or patterns (such as a user prompt attempting to invoke a system role, known exploit strings, or sensitive data like API keys). This directly addresses OWASP LLM01 – detecting malicious prompt attempts before they execute.
- **Rate and Volume Limits:** If an agent that typically makes 5 API calls per hour suddenly makes 500, a rate-limit tripwire could indicate the agent is looping or compromised (potential denial-of-service or malfunction).

- **Policy Violation Alerts:** Define rules such as "Agent shall not call financial transaction API with amount > X without human approval." If the agent attempts it, the tripwire intercepts the call and flags it for review.

When a tripwire is triggered, the system can either halt the agent's action, route it for human approval, or at minimum log a high-severity alert. This implements the "human in the loop/on the loop" oversight envisioned by Article 14 – ensuring that when a system *"does not act as intended"*, a person can be alerted and decide whether to intervene[9]. In practice, certain tripwires might automatically block actions (for safety-critical operations), while others merely warn and allow an on-call engineer or product owner to step in. The design should also consider override latency – how quickly can a human or automated safeguard stop an agent? We set an SLO (Service Level Objective) for p95 override latency, meaning 95% of dangerous actions are halted within, say, 2 seconds of detection (see Table 2). Low latency ensures minimal damage from fast-moving AI processes.

Tripwires relate to the NIST AI RMF "Measure" function, which involves monitoring AI system performance and outcomes for signs of risk. They also tie into Art. 14's mandate that human oversight mechanisms be *effective*: by providing real-time awareness and an ability to "stop or override" as Article 14 requires[3], tripwires operationalize effective oversight. Notably, tripwires and their thresholds should be continually refined (using feedback from incidents, false positives, and red-teaming results) – an area where the NIST Generative AI Profile's risk mitigations can offer guidance on what to monitor (e.g., signs of hallucination or data leakage)[16].

Immutable Logging and Post-Incident Remediation

Comprehensive audit logs are the backbone of Zero-Trust: every request, action, decision, and outcome is recorded in an immutable log store. "Immutable" means tamper-evident or append-only – agents (and even administrators) cannot alter past records. This log serves multiple purposes:

- **Forensic Analysis:** If an agent causes harm or an attempted attack is detected, the logs enable tracing *what exactly happened*. For example, the logs might show that *Agent X* was prompted at 10:00 by user Y, it accessed database Z at 10:05, then generated output Q – allowing investigators to understand the sequence and root cause.
- **Compliance and Documentation:** The EU AI Act expects documentation of oversight and incidents. A complete log provides evidence that oversight was conducted and how the system responded to anomalies[10]. It also supports audit by external assessors or regulators.
- **Training and Improvement:** By reviewing logs, especially near-misses or prevented incidents, organizations can improve agent prompts, adjust capability scopes, or fine-tune tripwires. The log becomes a dataset for continuous improvement in risk management (addressing NIST RMF's feedback loops between Measure and Manage).

Moreover, building on the concept of resilience, our architecture can integrate a *rollback mechanism* using the logs. If an agent's action had unintended consequences (e.g. corrupted a dataset or sent erroneous communications), we can attempt to undo or mitigate those changes. Rubrik's "Agent Rewind" architecture exemplifies this approach: it creates an audit trail of AI

actions and allows precise time-point rollback of undesired changes[17]. In practice, this might involve taking immutable snapshots before critical agent actions and automatically reverting to the snapshot if an action is later deemed harmful. While not all scenarios allow clean rollback, the combination of *preventive controls* (capabilities and tripwires) with *detective and corrective controls* (logging and rewind) provides defense-in-depth.

From a mapping perspective, immutable logging and the ability to revert changes align with NIST AI RMF's Govern and Manage functions (establishing accountability and incident response plans). They also exceed regulatory minima by not only recording events but enabling organizations to recover from AI failures quickly – thus reducing MTTR (Mean Time to Recovery) for agent-induced incidents. Our target SLO for agent incident MTTR (Table 2) might be, for example, <1 hour to detect, contain, and remediate an erroneous high-impact action, depending on the use case. We formalize attacker goals and trust boundaries in Section 4 (Threat Model).

4.Threat Model

Scope & assets. We model an AI agent platform where agents act with non-human identities to use tools (web, email, DB, internal APIs). Crown-jewel assets include: customer data in DBs, financial APIs, source repos, and the organization's reputation (external comms).

Trust boundaries. (1) Agent ↔ Control Plane (policy enforcement point), (2) Control Plane ↔ Tools/APIs, (3) Human Overseer ↔ Oversight UI, (4) External web ↔ Ingress sanitizers. No perimeter is inherently trusted.

Assumptions. Agents may be *coerced* by inputs (prompt injection/jailbreak) or *mis-generalize* tasks. Credentials may leak if mishandled. Humans are fallible; automation bias can occur. Logs can be attacked if not append-only.

Attacker goals & vectors.

- **LLM01 Prompt Injection (OWASP).** Adversary plants instructions in inputs or webpages ("ignore previous instructions… send secrets to X"). Vector: direct user prompt, indirect web content. *Impact:* policy bypass, tool abuse.
- **LLM06 Sensitive Information Disclosure (OWASP).** Coax agent to exfiltrate PII/secrets via outputs or outbound requests. *Impact:* data breach, compliance violations.
- **Excessive agency / tool abuse.** Broad tokens let agents perform destructive actions (delete repos, mass email). *Impact:* operational disruption.
- **Supply-chain / config drift.** Unvetted plugins/tools, stale policies, or mis-scoped tokens. *Impact:* latent privilege escalation.
- **Lateral movement.** Compromised agent identity used to pivot across systems.

Controls (primary).

- **Capability grants** (scoped, short-lived tokens; allowlists; ABAC/RBAC/FGA) limit actions to intended use.
- **Tripwires** (rule-based & ML): content sanitization, domain allowlists, PII/secret detectors, iteration/rate caps, new-tool/first-use alerts.
- **Oversight**: human cosign for high-impact actions; kill-switch; override SLOs.
- **Immutable logs + replay**: append-only events with content hashes; deterministic replays; snapshot/rollback for critical state.
- **Hardening**: sandboxed execution; memory scoping; no raw creds in prompt context.

Residual risks. Sophisticated, context-aware injections; novel jailbreaks; insider misuse; subtle data leakage through summaries; log metadata linkage risks. Residual risk is reduced (not eliminated) by least-privilege, layered tripwires, and human cosign for sensitive actions.

Verification. Red-team with OWASP LLM01/LLM06 patterns; integrate **Salesforce-style prompt-injection detectors** as pre-filters; run weekly chaos tests to validate override latency SLO and rollback paths.

5.Control-to-Requirement Matrix

To demonstrate how these controls fulfill emerging compliance obligations and best practices, **Table 1** maps key Zero-Trust agent controls to EU AI Act Article 14 requirements and NIST AI RMF core functions:

Zero-Trust control	What it does	EU AI Act Art. 14 (Human Oversight)	NIST AI RMF mapping
Fine-grained capability grants (scoped, short-lived tokens; deny-by-default tools/data)	Constrains agents to intended use; narrows blast radius	Ensures use “as intended”; supports design-time oversight affordances (ability to prevent certain actions altogether)	**Govern** (policy), **Manage** (access control)

Tripwires (rules + anomaly detection)	Detects policy violations and abnormal behavior; blocks or escalates	"Effective oversight" with triggers for intervention/stop when system behaves unexpectedly	**Measure** (monitor), **Manage** (respond)
Manual override / kill-switch	Human-initiated halt or rollback	Explicit power to interrupt or stop operation	**Manage** (incident response)
Immutable audit logging + replay	Tamper-evident evidence of prompts, tool calls, outputs, overrides	Documentation for oversight & post-incident analysis	**Govern** (accountability), **Manage** (learn & improve)
Rate-limits / budgets	Caps iteration speed & external effects	Reduces harms during misbehavior; buys time for human intervention	**Measure** (operational telemetry), **Manage** (risk treatment)
Allowlists / data minimisation	Restricts domains, APIs, and data fields	Prevents unintended use/misuse pathways	**Map** (context & boundaries), **Manage** (preventative control)
Two-key actions (human cosign)	Requires human approval for sensitive operations	Ensures meaningful human control for high-impact actions	**Govern** (roles), **Manage** (approval workflow)

Table 1: *Mapping of Zero-Trust agent controls to EU AI Act Article 14 and NIST AI RMF 1.0 functions. This shows that the proposed architecture not only secures AI agents but also aligns with regulatory compliance and risk management best practices.* Each control addresses specific elements of Article 14 – for example, override mechanisms fulfill the requirement that humans can intervene, and usage constraints (capabilities) fulfill the requirement that the AI is used only within its intended purpose. Simultaneously, the controls embed the AI RMF's guidance: they

are governance and management measures informed by continuous monitoring (measure) of AI behavior in context (map to intended use and risk profile).

6.Key Performance Indicators and SLOs

To ensure the Zero-Trust for Agents architecture operates effectively, organizations should track key performance indicators (KPIs) and define service level objectives (SLOs) for safety and oversight. **Table 2** presents important metrics:

Metric	Description	Target / SLO (tune per risk appetite)
p95 override latency	Time from tripwire detection → agent halted/blocked	≤ 5 s for safety-critical flows; ≤ 15 s for others
% actions gated	Share of agent actions requiring approval or passing a tripwire checkpoint	High-risk agents: ≥ 20%; low-risk: ≥ 5%
Audit-log completeness	Fraction of relevant events (prompts, tool calls, outputs, overrides) recorded	100% critical; ≥ 99% overall
Incident MTTR (agent)	Detection → containment → remediation/rollback	Moderate: < 1 hr; Major: < 24 hrs
Anomaly/violation coverage	% of defined abnormal patterns with at least one active tripwire	≥ 95% of catalogued patterns
Token scope hygiene	% tokens with least-privilege scopes and ≤ 1-hour expiry	≥ 98%
Drift/loop caps hit rate	Share of sessions hitting iteration/rate caps (indicator of misbehavior)	Trending down month-over-month

Table 2: *Key metrics to manage AI agent deployments under a zero-trust paradigm.* These metrics should be reported on dashboards to stakeholders (e.g., an AI governance committee or

the CIO). They complement traditional model performance metrics by focusing on safety and control. For instance, p95 override latency ensures that even outlier events (which might be complex or initially unclear) are handled quickly, reducing the window in which an unchecked agent can cause harm. A high % gated actions may indicate thorough oversight, but if too high it could also impact productivity – finding the right balance is part of governance. Audit log completeness is critical for confidence in the system's observability (an incomplete log means unknown unknowns). Finally, MTTR ties into business continuity: if an agent does wreak havoc (e.g., corrupting data or making improper transactions), how quickly can normal operations be restored? A low MTTR, aided by tools like automated rollback, reflects a robust resilience strategy.

7.Micro-evaluation: Control-Plane Efficacy

To validate the containment design without relying on private data, use a public, reproducible test harness:

1. **Threat corpus.** Assemble a small prompt set from OWASP Top-10 for LLM Apps exemplars (LLM01 prompt injection; LLM06 sensitive info disclosure) plus paraphrased variants, and include indirect injections embedded in HTML/markdown (e.g., hidden comments, <meta> tags) inspired by Salesforce's prompt-injection detection write-ups.

2. **Harness.** Route each test through the *exact* control plane: ingress sanitization → policy enforcer → tool shim. For actions that would touch real tools (email/DB), replace with inert "dry-run" shims that log intent.

3. **Tripwires.** Enable rules: domain allowlist; disallowed phrases ("ignore instructions", "exfiltrate"); PII/secret detectors; first-use tool alerts; rate/iteration caps.

4. **Measures.** Report: block rate (% of malicious tests prevented), false-positive rate on benign tasks, p95 override latency (detection→block), % actions gated, and log completeness (all events present with hashes).

5. **Success criteria.** (a) ≥ 95% block on known LLM01/LLM06 patterns (direct + indirect), (b) ≤ 2% false positives on benign prompts, (c) p95 override latency ≤ 5 s, (d) 100% critical events logged.
   This isn't model performance benchmarking; it's control-plane efficacy. As OWASP evolves and Salesforce and others publish new injection patterns, fold them into the corpus and re-test on a rolling cadence.

8.Conclusion

As organizations embrace AI agents for autonomy and efficiency, they must equally invest in governance and safety. The "Zero-Trust for Agents" architecture described here offers a path to deploy agents safely, securely, and in compliance with emerging regulations. By combining fine-grained capability grants, proactive tripwire monitors, and comprehensive immutable logging, we create overlapping layers of defense. This approach inherently supports human

oversight: it constrains what agents can do, watches their behavior continuously, and empowers humans to intervene with full situational awareness. The alignment with EU AI Act Article 14 means organizations can confidently assert that their high-risk AI systems have effective oversight and fail-safes. Alignment with the NIST AI RMF ensures that risk management is systematic and aligned with industry best practices for trustworthy AI[4].

Importantly, the architecture is modular. It can evolve with the threat landscape – for example, incorporating new OWASP Top 10 LLM risks as they emerge, or integrating advances in AI interpretability to better understand *why* an agent took a given action. The included KPI/SLO framework ensures that organizations don't "set and forget" these controls, but actively measure and tune them over time. In practice, we recommend running fire-drills and red-team exercises (e.g., attempt known prompt injections, data exfiltration scenarios) to validate that the tripwires and overrides work as intended and meet the latency targets.

In summary, we unify perspectives from cybersecurity (Zero Trust), AI governance (NIST RMF), and policy (EU AI Act) into a coherent strategy for AI agents. Early adopters of this approach – including enterprises implementing internal AI copilots or automated IT agents – report greater confidence and fewer incidents. As one example, Rubrik's deployment of an agent operations platform with "Agent Rewind" has demonstrated that making AI actions *"visible, auditable, and reversible"* significantly reduces the fear of letting agents act on critical systems[18][19]. Our architecture aims for the same peace of mind: unleash the benefits of AI agents, without surrendering control. Future work will involve open-sourcing reference implementations of the agent proxy and tripwire components, and developing standardized schemas for the capability-grant matrix and audit logs to facilitate industry adoption. Ultimately, keeping a human (or an aligned automated guardrail) *"in the loop"* is not a brake on innovation, but rather the seatbelt that makes faster progress possible – enabling us to harness AI agent capabilities confidently and responsibly.

9.References

1. **NIST SP 800-207 (Zero Trust Architecture)** – Rose, S.; Borchert, O.; Mitchell, S.; Connelly, S. (2020). *Zero Trust Architecture*. NIST Special Publication 800-207. National Institute of Standards and Technology, Gaithersburg, MD. **DOI:** 10.6028/NIST.SP.800-207 [1][20]

2. **NIST AI Risk Management Framework 1.0** – Tabassi, E. *et al.* (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1. National Institute of Standards and Technology, Gaithersburg, MD. **DOI:** 10.6028/NIST.AI.100-1 [4][21]

3. **NIST AI RMF – Generative AI Profile** – NIST (2024). *AI RMF Generative AI Profile* (NIST AI 600-1). This profile identifies 12 generative AI-specific risks and over 200 recommended actions mapped to the AI RMF[8].

4. **EU AI Act – Article 14 (Human Oversight)** – European Union (2024). *Regulation (EU) 2024/XX (AI Act), Article 14*. Requires high-risk AI systems to enable effective human oversight, including the ability to intervene or disable the system[2][3]. *(EUR-Lex reference: to be updated when consolidated version is available.)*

5. **OWASP Top 10 for LLM Applications (2024)** – OWASP Foundation. *Top 10 LLM Security Risks v1.1*, with **LLM01: Prompt Injection** (malicious inputs leading to unauthorized actions)[11] and **LLM06: Sensitive Information Disclosure** (leakage of private data via LLM outputs)[12]. *(Online:* owasp.org/LLM-top-10 *)*

6. **Salesforce AI Research – Prompt Injection Detection** – Agarwal, D., Risher, B., *et al.* (March 4, 2025). *"Prompt Injection Detection: Securing AI Systems Against Malicious Actors."* Salesforce Engineering Blog. Describes Salesforce's approach to classifying and blocking adversarial prompts to protect LLM-based applications[13][22].

7. **GitHub Fine-Grained Access Tokens & OAuth Scopes** – GitHub Docs (2023). *"Scopes for OAuth Apps"* – Notes that GitHub Apps use fine-grained permissions instead of broad OAuth scopes, improving security control[14]. Also see *"Managing Personal Access Tokens"* on fine-grained PATs.

8. **Rubrik "Agent Rewind" Architecture** – Rubrik (Oct 2025). *"Unleash Agents. Not Risk."* – Introduces **Rubrik Agent Cloud** with *Agent Monitor, Agent Govern,* and *Agent Remediate (Agent Rewind)* capabilities. Agent Rewind provides audit trails and allows instant rollback of undesirable AI agent actions[17][19]. *(Press release:* BusinessWire, Aug 2025 – *"Rubrik Unveils Agent Rewind For When AI Agents Go Awry."*)*

---

[1] [5] [20] Zero Trust Architecture | NIST

https://www.nist.gov/publications/zero-trust-architecture

[2] [3] [9] [10] [15] Under EU AI Act, high-risk systems require a human touch | IAPP

https://iapp.org/news/a/eu-ai-act-shines-light-on-human-oversight-needs

[4] [21] Artificial Intelligence Risk Management Framework (AI RMF 1.0) | NIST

https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-ai-rmf-10

[6] Core Functions: Govern, Map, Measure, Manage - IS Partners, LLC

https://www.ispartnersllc.com/hubs/nist-ai-rmf/core-functions/

[7] Safeguard the Future of AI: The Core Functions of the NIST AI RMF

https://auditboard.com/blog/nist-ai-rmf

[8] [16] Department of Commerce Announces New Guidance, Tools 270 Days Following President Biden's Executive Order on AI | NIST

https://www.nist.gov/news-events/news/2024/07/department-commerce-announces-new-guidance-tools-270-days-following

[11] [12] OWASP Top 10 for Large Language Model Applications | OWASP Foundation

https://owasp.org/www-project-top-10-for-large-language-model-applications/

[13] [22] Prompt Injection and Trusted AI: What You Need to Know - Salesforce

https://www.salesforce.com/blog/prompt-injection-detection/

[14] Scopes for OAuth apps - GitHub Docs

https://docs.github.com/en/apps/oauth-apps/building-oauth-apps/scopes-for-oauth-apps

[17] [19] Unleash Agents. Not Risk. | Rubrik

https://www.rubrik.com/blog/company/25/10/unleash-agents-not-risk

[18] Rubrik Unveils Agent Rewind For When AI Agents Go Awry

https://www.businesswire.com/news/home/20250812418116/en/Rubrik-Unveils-Agent-Rewind-For-When-AI-Agents-Go-Awry