WHITE PAPER — SECURITY & IDENTITY

Q1 2026 | Privileged and Confidential

Proof, Not Promise: Cryptographic Execution Identity for Autonomous AI

Why Identity and Access Management Was Not Built for Agents — and What a Cryptographic Execution Ledger Changes

Published by

Selfient.xyz | ROSÉ Federation

helen@selfient.com | Roko.Network

Abstract

Autonomous AI agents are no longer a roadmap item. They are operating in production environments today, executing consequential decisions across hiring, payments, security policy, and operational control — often under borrowed identities, with access privileges inherited from human accounts, and without any mechanism to produce non-repudiable proof of what they did, when, and under what rules.

This paper examines the structural incompatibility between traditional Identity and Access Management (IAM) architectures and the operational characteristics of autonomous AI agents. Drawing on March 2026 survey data from the Cloud Security Alliance and the February 2026 CSA/Strata Identity research, it maps the specific failure modes that arise when agent identity is borrowed rather than native — and when execution records are logs rather than cryptographic proof.

The central argument is architectural, not merely procedural: the governance gap in autonomous AI cannot be closed by extending existing IAM frameworks. It requires a fundamentally different primitive — one that binds cryptographic identity to execution events at the moment they occur, producing immutable, verifiable proof that satisfies regulators, courts, auditors, and incident response teams alike.

The paper concludes with a description of the Execution Ledger architecture that this problem demands, and the properties that any implementation must satisfy to close the gap.

I. The Production Reality — Agents Are Already Running

The enterprise conversation about autonomous AI agents shifted decisively in 2025 from feasibility to deployment. The question is no longer whether organizations will run agents in production — they already are. The question is whether the infrastructure assumptions beneath those agents are sound.

The March 2026 Cloud Security Alliance survey, Identity and Access Gaps in the Age of Autonomous AI (n=228 IT and security leaders), provides the most current quantitative snapshot of where enterprise deployments actually stand. The findings

are unambiguous and, for any security professional who has been watching this space, unsurprising in the worst possible way.

Source: CSA / Aembit, March 2026 (n=228)

Those three numbers, read in sequence, describe a security posture that should concern every CISO operating at scale. The overwhelming majority of organizations are running agents. More than two-thirds cannot tell, from their own logs, whether a given action was taken by a human or an agent. And fewer than one in five are highly confident that their IAM infrastructure adequately governs agent identities.

The February 2026 CSA/Strata Identity research adds further texture: only 28% of organizations report the ability to reliably trace agent actions across environments. That figure is worth dwelling on. Traceability is not an aspirational capability — it is the minimum threshold for meaningful incident response, regulatory compliance, and forensic defensibility. More than seven in ten enterprises currently operating autonomous AI agents in production cannot meet that threshold.

85% of organizations are running autonomous AI agents in production. 68% cannot tell, from their own logs, whether a given action was taken by a human or an agent.

II. The Identity Architecture Problem — Why IAM Was Not Built for Agents

To understand why the numbers above are not an operational failure but an architectural one, it is necessary to examine the foundational assumptions on which modern IAM was designed — and the ways in which autonomous AI agents systematically violate each of them.

2.1 The Human-Centric Identity Model

Contemporary IAM frameworks — whether built on LDAP, OAuth 2.0, SAML, or zero-trust ZTNA architectures — share a common design premise: the identity subject is a human being, or a static, predictable workload acting on behalf of a human being. The provisioning model assumes that identities can be enumerated in advance, that their behavioral patterns are bounded and predictable, and that access privileges can be assigned to a known, stable principal.

Autonomous AI agents break every one of these assumptions simultaneously. An agent's identity is rarely native — it is almost universally borrowed from an existing account: a service principal, a shared workload identity, or, in the cases that should alarm every security architect reading this paper, a human user account. The agent's behavioral envelope cannot be enumerated in advance, because the combinatorial explosion of possible execution paths in a goal-directed agentic system is, for practical purposes, unbounded. And the agent's access requirements shift dynamically as it pursues multi-step tasks that were not fully specified at provisioning time.

ARCHITECTURAL OBSERVATION

The problem is not that organizations are managing agents badly within their existing IAM frameworks. The problem is that the IAM framework was designed for a class of principal that agents are not — and no amount of policy refinement within that framework resolves the structural mismatch.

2.2 The Borrowed Identity Attack Surface

When an agent operates under a borrowed or shared identity, the security implications cascade across four distinct dimensions:

Failure Mode	Security and Compliance Implication
Attribution loss	Log entries become ambiguous — the fundamental question of forensic analysis, 'which principal performed this action?', cannot be answered with confidence. Incident response timelines extend dramatically.
Blast radius amplification	An agent operating under a privileged service account or a human's credentials inherits the full access scope of that account. A single misbehaving or compromised agent can move laterally across the blast radius of the account it borrowed.
Principle of least privilege violation	The CSA data confirms that agents routinely receive more access than necessary — because access is provisioned for the identity, not the task. Dynamic, task-scoped access management does not

	exist in most current IAM implementations.
Non-repudiation failure	When a consequential action — a hiring decision, a payment release, a security policy modification — is attributed to a shared identity, neither the organization nor any regulator or court can establish with certainty that a specific agent took that specific action under specific rules. That is a non-repudiation failure by definition.

2.3 The Log Fidelity Problem

Even where organizations maintain comprehensive logs, the 68% figure from the CSA survey reveals a more fundamental issue than volume or retention: it is a fidelity problem. A log entry that records an action against a shared identity does not establish what actually happened. It establishes that something happened under a particular credential at a particular server clock time. Those are not equivalent propositions.

Four specific log fidelity deficiencies characterize current enterprise environments:

- **Identity ambiguity:** The log records the credential, not the agent. In environments where multiple agents share a service principal, log entries are categorically unable to establish which agent acted.
- **Timestamp unreliability:** Server clock timestamps are subject to drift, NTP misconfiguration, and — in adversarial scenarios — manipulation. They do not constitute hardware-attested time and will not withstand rigorous forensic scrutiny.
- **Context absence:** Logs record actions, not the reasoning, inputs, model state, or governance rules that produced them. Reconstructing the decision context from a log entry is, in most current architectures, impossible.
- **Mutability:** Standard log infrastructure does not produce tamper-evident records. In a post-incident or litigation context, the provenance of log data is itself challengeable.

A log entry is a record that something happened. Cryptographic proof is evidence of what happened, in what order, under what rules, at a verifiable point in time. These are not the same thing — and the difference is the entire gap between a defensible organization and an indefensible one.

III. The Regulatory Convergence — Why This Gap Is Now a Liability

The IAM gap described above would be a significant operational problem in any regulatory environment. In the current and emerging environment, it is an acute liability exposure that is accelerating on multiple jurisdictional fronts simultaneously.

3.1 The EU AI Act — High-Risk AI and the Proof Imperative

The EU AI Act's classification of employment-related AI as high-risk carries specific technical obligations that the borrowed-identity, log-based architecture described above cannot satisfy. The Act requires automatic logging throughout the system lifecycle that is sufficient to identify risks and track modifications — not simply to record that actions occurred. It requires documentation that demonstrates, to a regulator's satisfaction, what the system did and under what governance rules. And it requires human oversight infrastructure capable of genuine intervention, which presupposes the ability to reconstruct exactly what the agent did before the oversight decision is made.

When 68% of organizations cannot distinguish agent from human actions in their existing logs, the prospect of satisfying an EU AI Act audit with those same logs is not credible. The Act's August 2026 full enforcement date — now months away — makes this an immediate compliance engineering problem, not a future planning consideration.

3.2 Employment and Algorithmic Decision Law

Beyond the EU AI Act, a rapidly converging set of algorithmic employment decision laws creates additional proof obligations specifically in the hiring and workforce management contexts where autonomous agents are most actively deployed:

- **New York City AEDT Law:** Requires bias audits and candidate notification for automated employment decision tools, with audit documentation that must withstand regulatory inspection.
- **Illinois Artificial Intelligence Video Interview Act:** Requires disclosure and consent for AI analysis of candidate videos, with retention and audit obligations.
- **Colorado AI Act:** Imposes impact assessment and transparency requirements on high-risk AI decisions, including employment decisions.

- **EU GDPR Article 22:** Confers the right not to be subject to solely automated decisions, enforceable by candidates who request human review — which requires the organization to produce the decision record and demonstrate it can be meaningfully reviewed.

COMPOUNDING EXPOSURE

Each of these frameworks independently requires documentation that the current log-based, borrowed-identity architecture cannot reliably produce. In combination — and an organization hiring across jurisdictions faces all of them simultaneously — the exposure is not additive. It compounds.

3.3 The Incident Response and Insurance Dimension

The governance gap also has direct implications for cyber insurance and incident response that security architects are increasingly required to address at the underwriting stage. Insurers covering AI-related operational failures and regulatory actions are beginning to require documentation of agent governance infrastructure as a condition of coverage. Organizations that cannot demonstrate native agent identity, tamper-evident execution records, and auditable governance rules face either coverage exclusions or premium structures that price the gap directly.

The Gartner projection that more than 40% of agentic AI projects will be canceled by the end of 2027 due to escalating costs, unclear value, and inadequate risk controls is, in part, a reflection of this dynamic: organizations discovering, after deployment, that their agent governance infrastructure is not insurable at acceptable cost.

IV. The Failure Mode Taxonomy — What Goes Wrong and When

It is instructive to map the specific failure modes that arise from the architectural gap, because the security and compliance communities often treat these as distinct problems when they share a common structural root.

Failure Mode	Proximate Cause	Downstream Impact
Attribution failure in incident response	Shared identity: log cannot establish which agent acted	Root cause isolation takes weeks; damage amplifies during reconstruction delay

Regulatory non-compliance finding	Logs insufficient to satisfy EU AI Act or AEDT audit	Fines up to 3-7% of global turnover; mandatory remediation orders
Lateral movement from compromised agent	Excessive inherited privileges from borrowed account	Blast radius limited only by the scope of the borrowed credential
Indefensible employment dispute	No cryptographic record of decision rules at execution time	Organization cannot demonstrate compliance with hiring law; settlement exposure
Insurance coverage gap	No auditable agent governance documentation	Exclusions or uncovered losses on AI-related operational failures
Governance policy drift	No mechanism to prove which rules governed which execution event	Policy changes retroactively applied in reconstructed records; audit integrity compromised

The common thread across every row in that table is not a configuration failure, a policy gap, or an operator error. It is a missing primitive: the ability to produce, at the moment of execution, a cryptographically bound, tamper-evident record that ties a specific agent identity to a specific action taken under specific, immutable rules at a verified point in time.

V. What the Architecture Must Provide — Required Properties of an Execution Ledger

The architectural response to the problem described above is not an extension of existing IAM. It is a new primitive that operates at the execution layer — below the application, above the substrate — and produces the cryptographic guarantees that neither logs nor conventional identity frameworks can provide.

Any implementation that claims to close this gap must satisfy the following five properties. Organizations evaluating solutions in this space should treat these as threshold requirements, not differentiating features.

5.1 Native Cryptographic Agent Identity

Each agent must carry an identity that is cryptographically distinct, non-inherited, and non-transferable. This is architecturally incompatible with the current practice of issuing agents service account credentials or shared workload identities. A native agent identity is provisioned for the agent, not borrowed from a human or shared workload principal. It is bound to the agent's execution context and cannot be replicated, delegated, or assumed by another principal.

This property alone eliminates the attribution ambiguity that renders 68% of enterprise logs forensically unreliable. When every agent action is signed with a native cryptographic identity, the question 'was this a human or an agent?' is answered not by inference from behavioral patterns but by cryptographic proof.

5.2 Hardware-Attested Timestamping

The timestamp bound to each execution event must be derived from hardware-attested time, not server clock time. The distinction is not academic. Server clocks can drift, can be misconfigured, and — in adversarial scenarios — can be manipulated. Hardware security modules providing precision time produce timestamps that satisfy the evidentiary standard required for regulatory audit, legal discovery, and forensic chain of custody.

The Precision Time Protocol (PTP) and GPS-synchronized time sources provide the hardware attestation layer that transforms a timestamp from a metadata annotation into an auditable fact. An execution ledger that relies on server clock time does not satisfy the proof standard that the EU AI Act and analogous frameworks will apply.

5.3 Immutable Execution Ordering

In multi-agent and multi-step agentic workflows, the sequence of execution events is often as legally and operationally significant as the events themselves. Which agent acted first, which governance rule was applied before or after a state change, whether a human oversight checkpoint was traversed before a consequential action — these are questions of ordering, not merely of occurrence.

An execution ledger must establish and preserve execution ordering cryptographically, such that the sequence cannot be retroactively altered or disputed. This requires consensus-level ordering guarantees, not application-layer sequencing that can be rewritten by anyone with database access.

5.4 Governance Rule Immutability at Execution

One of the most insidious vulnerabilities in current enterprise AI governance is the possibility — and in some architectures, the practical reality — that governance rules can change after an execution event has occurred, making it impossible to establish which rules governed a specific decision. Policy updates, model version changes, and access control modifications can retroactively alter the apparent governance context of historical decisions.

A compliant execution ledger must lock the governance rules in force at execution time as part of the cryptographic proof bundle for that event. The rules that governed a hiring decision made on a specific date must be recoverable from the ledger entry for that decision — permanently and immutably — regardless of subsequent policy changes.

5.5 Auditable Proof Bundles — Seconds, Not Months

The operational test of any governance architecture is the time required to answer the regulator's or auditor's question: what exactly happened, when, under what rules, in what order? In current enterprise environments, the answer to that question — assembled from distributed logs, access records, model version histories, and policy documentation — typically requires weeks to months of forensic reconstruction.

An execution ledger must produce verifiable proof bundles that answer that question in seconds. Not because speed is the primary value, but because the ability to produce proof on demand is what distinguishes an infrastructure investment from a forensic archaeology project. The regulator who can receive a proof bundle within minutes of a request is encountering a different kind of organization than the one that takes six months to reconstruct what its agents did.

The question is not whether you can eventually reconstruct what your agent did. It is whether you can prove it — cryptographically, immediately, to a standard that satisfies a court.

VI. From Failure to Intelligence — Proof as Operational Resilience

A consideration that receives insufficient attention in the IAM and governance literature is the operational value of cryptographic proof infrastructure beyond its defensive and compliance applications. The security community has largely framed the agent governance problem as a risk mitigation problem — how do we prevent bad things from happening? That framing, while important, understates the value of knowing precisely what happened.

In a system equipped with an execution ledger, every agent failure — every misbehavior, every unexpected output, every compliance near-miss — becomes high-fidelity training data. The organization is not left asking 'what went wrong?' because the ledger records every input, intermediate reasoning step, tool call, governance rule application, and output in a tamper-evident sequence. Root cause is isolated in minutes. The failure mode is characterized precisely. And the governance policy, the model prompt, or the oversight rule that needs adjustment is identified with specificity rather than estimated from fragmentary evidence.

This transforms the economics of agentic AI deployment. Organizations that fear deploying more capable agents because the blast radius of failure is unknown are, in significant part, reacting to the absence of proof infrastructure. When every execution event is cryptographically bounded, the blast radius of failure is no longer unknown — it is precisely recoverable. That knowledge changes the risk calculus materially.

VII. The ROSÉ Execution Ledger — An Implementation of These Properties

The five properties described in Section V are not hypothetical requirements. They are implemented in the ROSÉ Execution Ledger, a purpose-built infrastructure layer developed by the ROSÉ Federation — a consortium of six specialized technology systems assembled by Selfient.xyz.

ROSÉ was not adapted from cloud infrastructure originally designed for human-era workloads. It was designed from first principles for the operational reality that

autonomous agents are executing consequential decisions at machine speed and at enterprise scale — and that every such decision must be provable.

7.1 How the Execution Ledger Closes the Five Gaps

Required Property	Current Gap	ROSÉ Implementation
Native cryptographic agent identity	Agents borrow service accounts or shared credentials; attribution is ambiguous	Each agent carries a non-transferable cryptographic identity bound at execution — no shared accounts
Hardware-attested timestamps	Server clock timestamps are unreliable and potentially manipulable	TimeBeat provides hardware security module-grade precision timestamps independent of server infrastructure
Immutable execution ordering	Log sequencing is application-layer and rewritable by privileged users	ROKO provides L1 substrate ordering with consensus-level finality; sequence is cryptographically sealed
Governance rule immutability	Policy changes can retroactively alter the apparent governance context of historical decisions	Selfient smart contracts lock governance rules as part of the execution proof bundle at the moment of execution
On-demand proof bundles	Forensic reconstruction takes weeks to months from distributed logs	Matric notary artifacts and Fortémi queryable records produce verifiable proof bundles on demand

7.2 Target Environments

The Execution Ledger is designed for high-risk domains where the consequences of unattributable or unverifiable agent actions are most acute:

- Employment and talent acquisition — AI-mediated hiring, screening, and onboarding decisions
- Financial services — algorithmic trading, credit decisioning, payment authorization, and KYC/AML agent workflows
- Healthcare — diagnostic support, treatment recommendation, and clinical workflow automation
- Insurance — underwriting automation and claims adjudication

- Public sector — benefits determination, enforcement support, and regulatory compliance workflows

In each of these environments, the organization's ability to operate autonomous agents at scale depends ultimately on its ability to prove what those agents did. The Execution Ledger provides that capability.

VIII. Conclusion

The Cloud Security Alliance data from early 2026 describes an industry in the early stages of a reckoning that security architects have seen before, in different forms: a powerful new class of system deployed at scale before the governance infrastructure appropriate to it was in place. The difference, this time, is that the systems are goal-directed, the decisions are consequential, and the regulatory environment demanding proof of governance is not emerging — it is here.

The 68% of organizations that cannot distinguish agent actions from human actions in their logs are not, in most cases, making a negligent choice. They are operating the only infrastructure that existed when they needed to deploy. The question before the security and compliance community now is not whether that infrastructure was adequate then — it is whether it will be adequate when the first regulatory audit arrives, the first employment discrimination claim is filed, or the first agentic system failure requires forensic reconstruction.

The architectural answer is not a better log. It is a cryptographic execution ledger that makes every agent action provably accountable — by identity, by sequence, by governance rule, by verified time. Organizations that build or adopt that infrastructure before enforcement pressure arrives will find themselves in a categorically different position than those that build it in response to a finding.

The age of autonomous AI is not coming. It is operating, right now, in 85% of enterprise environments. The governance infrastructure it requires is not optional. The only variable is when it gets built — and whether it gets built before or after the event that makes its absence undeniable.

Proof, Not Promise. The organizations that scale agentic AI with confidence will be those that can prove what their

agents did — not merely assert that their logs suggest it probably happened correctly.

About ROSÉ and Selfient.xyz

ROSÉ is the Regulated Ordered and Signed Execution federation — purpose-built execution proof infrastructure for autonomous AI. Operating as a federation of six specialized systems under Selfient.xyz, ROSÉ provides the cryptographic execution ledger that the EU AI Act, ISO 42001, and every emerging AI governance regime demands.

ROSÉ is led by Helen Sharron, CEO and Co-Founder, a 25-year veteran of enterprise technology leadership at State Street Bank, Putnam Investments, and BNYMellon, and a two-time founder with a successful exit in 2013. The federation includes Matric, ROKO.Network, Latitude, Fortémi, and TimeBeat.

To engage with the ROSÉ team or discuss design partnership opportunities:

helen@selfient.com | +1 413 446-3032 | Roko.Network

ROSÉ · Selfient.xyz · Privileged and Confidential · Q1 2026 · References: Cloud Security Alliance, Identity and Access Gaps in the Age of Autonomous AI (March 2026, n=228); CSA/Strata Identity research (February 2026); Gartner Agentic AI Forecast 2027.