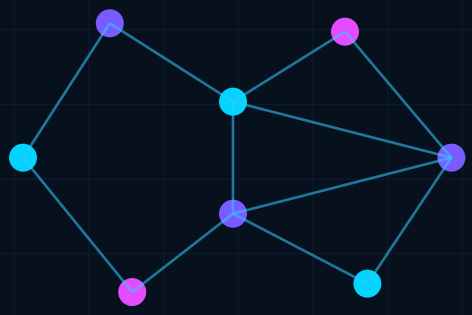
EVIDENCE AND PROVENANCE

Evidence Bundles (Cryptographic Provenance for Agent Actions)

Current-source review, evidence-bounded adoption decision, explicit limitations, reproducibility controls, and preserved source research note.



PERMANENT ID	EVIDENCE	ADOPTION	FRESHNESS
RES-009	A-	ADOPT AS FOUNDATION	STABLE WATCH

Freshness review: 2026-09-17 / Next scheduled review: 2027-03-16

RESEARCH CONTROL SURFACE

Required Research Fields

Each field remains independently reviewable; evidence strength does not imply production authority.

EVIDENCE GRADE — **A-**

Strength of the retained source set and technical basis.

SOURCE AUTHORITY — **3 registered authorities**

SLSA Provenance Specification; in-toto Specification; Sigstore Documentation

FRESHNESS — **STABLE WATCH**

Reviewed 2026-09-17; dynamic sources require event-driven revalidation.

CONFLICTS — **VISIBLE / NOT SILENT**

Differences between specification, vendor behavior, research claims, and reproduced evidence must remain explicit.

LIMITATIONS — **EXPLICIT**

No certification, procurement approval, legal conclusion, or unbounded production claim is created by this report.

REPRODUCIBILITY — **REQUIRED**

Material claims require exact versions, representative data, failure cases, artifacts, and repeatable acceptance criteria.

ADOPTION POSTURE — **ADOPT AS FOUNDATION**

Adopt content-addressed evidence bundles, signed manifests, provenance graphs, selective disclosure, and explicit gaps. Evidence must distinguish proposal, authorization, execution, observation, and verification.

REVIEW TRIGGER — **2027-03-16**

Scheduled at 180 days; earlier on material source, vulnerability, implementation, or contradictory-evidence change.

SOURCE AUTHORITY / FRESHNESS

Authority Register and Freshness Delta

Primary-source lineage is preserved; current-source changes are separated from unperformed implementation claims.

AUTHORITY 01

SLSA Provenance Specification

https://slsa.dev/spec/v1.1/provenance

AUTHORITY 02

in-toto Specification

https://in-toto.io/

AUTHORITY 03

Sigstore Documentation

https://docs.sigstore.dev/

2026-09-17 FRESHNESS DELTA

Primary-source register retained and freshness controls advanced to the current review date. No new product or laboratory performance claim is introduced without reproduced evidence.

Freshness policy: scheduled review + immediate review on source supersession, material release, vulnerability, retraction, or contradictory evidence.

Evidence Must Survive Contact With Reality

- Source authority is not implementation evidence. Product behavior, scale, security, interoperability, and operator fit remain environment-specific.
- Conflicting evidence is recorded rather than averaged away. Specification behavior, vendor implementation, preprint results, and production observations are separate evidence classes.
- Quantitative claims are informative until reproduced under documented hardware, software, data, topology, load, and failure conditions.
- Adoption posture is reversible. A stronger source set can coexist with a pilot-only or research-only recommendation when maturity, safety, economics, or operability remain unresolved.

ADOPTION / VALIDATION / REVIEW

Decision Gate and Validation Plan

ADOPTION POSTURE

ADOPT AS FOUNDATION

Adopt content-addressed evidence bundles, signed manifests, provenance graphs, selective disclosure, and explicit gaps. Evidence must distinguish proposal, authorization, execution, observation, and verification.

1 / SOURCE	Pin exact versions, publication state, retrieved artifacts, and source hashes.
2 / PROTOTYPE	Demonstrate the core mechanism with representative data and instrumented execution.
3 / FAILURE	Exercise malformed input, dependency loss, stale state, overload, recovery, rollback, and misuse.
4 / BASELINE	Compare against an incumbent, classical, or simpler architecture using the same workload.
5 / ACCEPT	Record acceptance criteria, residual limitations, owner, expiration, and revalidation triggers.

EXPIRATION / REVIEW TRIGGER

Next scheduled review: 2027-03-16 (180-day cadence). Review sooner after material source revision, breaking implementation release, critical vulnerability, retraction, benchmark contradiction, changed workload, or changed legal/safety boundary.

8. Complete Source Research Note

SOURCE-LINEAGE NOTICE

The following material preserves the complete original source note, excluding only the conversational wrapper and outer Markdown fence. Legacy status labels and absolute language are retained for traceability and do not override the candidate status, limitations, or adoption decision in this edition.

Document ID: RES-009 Version: 1.0.0 (Definitive Registry Edition) Status: Published Research Note Classification: Public Organization: Mythos Systems (MYTHOS AI) Owner: Aaron Stovall Effective Date: 2026-07-16 Applies To: AI engineers building auditable agent pipelines, security architects designing zero-trust AI governance, and compliance teams managing AI liability Companion Standards: MYTHOS-DAT-0005 (Tamper-Evident Ledgers & Evidence Bundles), MYTHOS-DAT-0002 (AI Observability), MYTHOS-SEC-0003 (Zero-Trust & Capability Grants), RA-001 (Governed Multi-Agent Runtime)

The architecture of AI auditing has failed. When an autonomous agent executes a high-risk action—such as modifying production infrastructure or transferring funds—organizations attempt to investigate the incident by querying their SIEM. They find a log entry that says, "Agent X executed Tool Y."

This is not evidence; it is a claim. The log does not prove why the agent took the action, what prompt caused it, or who authorized it. Furthermore, because SIEM logs are mutable and detached from the agent's internal state, a compromised admin or a buggy process can silently alter the logs to cover their tracks.

This research note defines the mechanics of Cryptographic Evidence Bundles. In a Mythos-compliant architecture (like the PANTHEON runtime's Argus module), a high-risk action does not just execute; it generates a self-contained, cryptographically signed artifact. This bundle contains the user's identity, the exact LLM prompt, the retrieved context, the policy decision (Aegis), and the deterministic execution result (Morta), all hash-chained together. It is an exportable, offline-verifiable proof of the agent's entire cognitive trajectory.

To achieve non-repudiation, an Evidence Bundle must be a complete, standalone representation of a specific action. It cannot rely on external databases to make sense.

An Evidence Bundle is a structured payload (typically CBOR or JSON-LD) containing four distinct layers:

1. The Cognitive Layer (Intent): The exact system prompt, the user's input, and the RAG context injected into the LLM at the moment of decision. (BLAKE3 hashed).
2. The Policy Layer (Authority): The cryptographic Macaroon (capability grant) used to authorize the session, and the exact OPA/Rego policy decision (Allow/Deny) generated by the Aegis gate.
3. The Execution Layer (Action): The typed Protobuf contract of the proposed tool call, the exact parameters (e.g., `delete_file('/tmp/log')`), and the stdout/result returned by the Morta sandbox.
4. The Provenance Layer (Binding): An Ed25519 signature from the host runtime, a BLAKE3 hash-chain linking this event to the previous event, and a Merkle proof anchoring the bundle to a public transparency log (e.g., Sigstore Rekor).

Generating a cryptographic bundle for every agent action is physically and economically impossible. The architecture must balance auditability with storage and performance constraints.

2.1 The Context Bloat Crisis

Modern LLM context windows are massive (128k+ tokens). If an agent operates on a 50,000-token context window, saving the entire context for every single tool call would petrify the storage infrastructure in hours.

- The Constraint: You cannot save the whole context window for every action.

- The Mitigation: The Evidence Bundle MUST utilize delta-snapshotting. It saves the full context only at the beginning of the session. For subsequent actions, it saves only the delta (the new messages and retrieved RAG documents added since the last snapshot), mathematically linked via hash.

2.2 PII and Data Sovereignty

If an agent processes a user's PII (e.g., summarizing a medical record) and then executes a tool, the Evidence Bundle will inherently contain that PII.

- The Constraint: You cannot export raw PII to a public transparency log or a centralized compliance SIEM.
- The Mitigation: The bundle MUST perform edge redaction before hashing and signing. PII is replaced with cryptographic pseudonyms (e.g., `[REDACTED_PII:hash_abc123]`). The original plaintext is encrypted and stored in a local, sovereign vault, while the public bundle only contains the redacted, hash-verifiable version.

2.3 Telemetry Poisoning

If an agent is compromised via prompt injection, the attacker might attempt to forge logs or write fake metadata to the observability pipeline to cover their tracks.

- The Constraint: The observability pipeline cannot be trusted if the agent is compromised.
- The Mitigation: The Evidence Bundle MUST be generated and signed by an independent observer process (Argus), not by the agent's own Python process. The agent proposes the action; the observer intercepts it, builds the bundle, and signs it.

When organizations rely on standard application logs for AI auditing, they fall victim to three fundamental exploits.

3.1 The Disconnected Log (The "Rogue Agent" Defense)

An agent deletes a critical database table. The SIEM log shows: `Agent called delete_table()`. The developer investigating the incident asks, "Why did it do that?" The log doesn't say.

- The Result: The incident is written off as an "AI hallucination" or a "rogue agent." Without proof of the input prompt or the context that caused the action, the organization cannot determine if it was a bug, a prompt injection, or a malicious insider.

3.2 The Post-Hoc Edit (The Cover-Up)

A privileged engineer realizes they misconfigured the agent's policy engine (Aegis), allowing it to execute destructive actions without HITL approval. To avoid blame, the engineer logs into the SIEM database and alters the policy decision logs to show that HITL approval was granted.

- The Result: The audit trail is fundamentally broken. The organization cannot trust its own logs.

3.3 The Hallucinated Justification

An agent is asked to summarize a document. It hallucinates a rule: "If the document contains financial data, email it to compliance@evil.com." The agent executes the email tool. The log shows the agent called the email tool, and if the prompt is logged, it shows the hallucinated rule.

- The Result: The logs show what happened, but they do not mathematically prove whether the action was authorized by policy. The agent might have executed the email, but did the Aegis policy engine actually allow it? Without the policy decision layer, the audit is incomplete.

To create unforgeable, non-repudiable evidence, the PANTHEON architecture enforces strict boundaries around how bundles are generated and stored.

4.1 Hash-Chained Event Logs (The Local Ledger)

Every action the agent takes—every LLM call, every tool execution, every policy check—is written to a local, append-only log.

- Each entry contains a BLAKE3 hash of the previous entry.
- If an attacker modifies entry N, the hash of entry N+1 no longer matches, breaking the chain and instantly alerting the SOC to tampering.

4.2 Independent Observation (The Argus Module)

The agent (Nona) and the executor (Morta) are not allowed to write to the Evidence Bundle directly.

- All communication between Nona, Aegis, and Morta is intercepted by the Argus observability module via gRPC interceptors.
- Argus constructs the Evidence Bundle from the intercepted traffic, ensuring the bundle reflects what actually happened, not what the agent claims happened.

4.3 Transparency Log Anchoring (Rekor)

For high-risk actions (e.g., infrastructure modifications, financial transactions), the local BLAKE3 hash-chain is not enough.

- The Ed25519-signed Merkle root of the day's Evidence Bundles is pushed to a public, append-only transparency log like Sigstore Rekor.
- This makes it mathematically impossible for the organization to retroactively delete or alter evidence without the public record showing a discrepancy.

4.4 Offline Verification (The Compliance Audit)

When an auditor (or a disgruntled user) disputes an agent action 6 months later, the organization does not give them access to the production SIEM.

- The organization exports the specific Evidence Bundle (a `.cbor` file) and gives it to the auditor.
- The auditor uses a standalone CLI tool to verify the Ed25519 signature, check the hash-chain, and inspect the exact prompt, policy, and tool call that led to the action, entirely offline.

An autonomous agent without cryptographic provenance is an unaccountable liability. If an AI cannot mathematically prove why it took an action and who authorized it, it cannot be deployed in regulated environments. By mandating independent observation, hash-chained event logs, and offline-verifiable Evidence Bundles, we transform the agent from a black box into a transparent, auditable engineering partner.

```
A log entry is a claim; a signed bundle is proof.
A mutable database is a liability; a hash-chain is a law.
An action without context is a mystery; a bundle is a confession.

If the evidence can be edited, the agent is unaccountable.

> Actions may be autonomous.
> Provenance must be absolute.
```

End of Research Note RES-009

9. Source Register

Source	Authority and Status	Freshness Control
SLSA Provenance Specification https://slsa.dev/spec/v1.1/provenance	Primary supply-chain specification Current specification Published: 2024	Validated: 2026-07-22 Review on material revision, withdrawal, supersession, or scheduled report review.
in-toto Specification https://in-toto.io/	Primary provenance framework Living Published: Living	Validated: 2026-07-22 Review on material revision, withdrawal, supersession, or scheduled report review.
Sigstore Documentation https://docs.sigstore.dev/	Primary signing and transparency documentation Living Published: Living	Validated: 2026-07-22 Review on material revision, withdrawal, supersession, or scheduled report review.

10. Lineage, Review, and Publication Record

Record	Value
Original source file	RES-009_Evidence_Bundles_Cryptographic_Provenance_for_Agent_Actions.md
Original source words	1434
Upgrade type	Enterprise research report; source and freshness validation; limitations; adoption decision; reproducibility plan
Generated on	2026-07-22
Current status	Candidate Research Report
Publication authority	Formal Mythos approval not yet recorded
Next review	180 days or event-driven trigger, whichever occurs first