Request for Information Regarding Security Considerations for Artificial Intelligence Agents

Docket No. NIST-2025-0035

91 FR 698 • January 6, 2026

Public Comment

ATTESTED INTELLIGENCE

Jack Brennan

admin@attestedintelligence.com

Illinois File No. 17233815

USPTO Patent Application No. 19/433,835

USPTO Trademark Serial No. 99677085

March 4, 2026

AttestedIntelligence.com

TABLE OF CONTENTS

INTRODUCTION

Attested Intelligence Holdings LLC is a security engineering firm specializing in cryptographic governance infrastructure for autonomous systems operating in adversarial and disconnected environments. Our technical foundation includes a patent-pending architecture (USPTO Application No. 19/433,835) with 20 claims, a validated cryptographic core, and working demonstration applications for SCADA process enforcement and autonomous vehicle command governance. We submit this comment not as a theoretical contribution but as a report from active engineering work on the exact class of problems this RFI identifies.

The architecture we describe employs what we term an **"Active Immutable Object"** pattern: generating sealed Policy Artifacts from successful attestation and deploying those sealed objects as authoritative runtime configurations enforced by a portal execution boundary. We refer to these sealed objects as **Attested Governance Artifacts (AGA)**.

> **Active Immutable Object**
>
> An architectural pattern in which a sealed, cryptographically signed Policy Artifact is deployed as an authoritative runtime-governing object. The artifact dictates rules; a portal enforcement boundary enforces rules; the portal writes back cryptographic proof of enforcement. The artifact is immutable once sealed. Any modification invalidates its signature.

EXECUTIVE SUMMARY

This comment proposes a concrete, patent-pending architectural framework for securing AI agent systems through cryptographic runtime governance: **Attested Governance Artifacts (AGA)**. Rather than offering abstract principles, we describe an engineered system with defined claims, a working reference implementation, and direct applicability to the security challenges identified in this RFI. Our key recommendations to CAISI:

1. **Mandate Sealed Reference States.** AI agent systems should operate under cryptographically sealed policy artifacts encoding permitted behavior, measurement cadences, and enforcement triggers. Agents should not execute absent a valid, signed governance artifact.

2. **Require Continuous Runtime Measurement.** Post-deployment verification must be continuous, not periodic. Each measurement should produce a signed receipt appended to a tamper-evident audit chain. This is cryptographic proof of monitoring, not merely logs.

3. **Adopt Tiered Verification Levels.** Governance rigor should scale with deployment consequence through standardized tiers (Device-Attested, Server-Attested, Blockchain-Anchored), analogous to FIPS 140-2/3 security levels.

4. **Mandate Offline Verifiability for Critical Infrastructure.** AI agents in defense, SCADA, and air-gapped environments must be governable and auditable without network connectivity. Evidence bundles containing artifacts, receipts, and Merkle proofs should enable fully portable verification.

5. **Standardize Governance Artifact Formats.** A machine-parseable, cryptographically signed artifact schema should be standardized to enable interoperability across vendors, enforcement tools, and verification systems.

6. **Require Privacy-Preserving Governance Disclosure.** Cross-organizational governance evidence sharing should employ policy-gated disclosure with automatic substitution, preventing sensitive data exposure during compliance verification.

Questions Addressed

RFI Question	Topic	Section
1(a), 1(d)	Unique threats; threat evolution	Section 1
2(a), 2(e)	Technical controls; framework alignment	Section 2
3(a), 3(b)	Assessment methods; tiered verification	Section 3
4(a)-4(d)	Environment constraints; counterparties; monitoring	Section 4
5(a)-5(c)	Adoption; collaboration; research priorities	Section 5

1. SECURITY THREATS, RISKS, AND VULNERABILITIES

Response to Question 1(a): Unique Security Threats

AI agent systems introduce a category of security risk absent from traditional software: **runtime behavioral drift that is undetectable by conventional integrity monitoring**. Unlike traditional applications where unauthorized modification manifests as file changes or process anomalies, AI agent systems can exhibit compromised behavior through model weight manipulation, prompt injection, context poisoning, or tool-use hijacking. None of these alter the system's binary footprint in ways that standard endpoint protection detects.

We identify three threat categories that existing security frameworks inadequately address:

Semantic Drift Without Binary Modification. Traditional integrity measurement (file hashing, binary attestation) fails to detect behavioral compromise through adversarial inputs. Governance mechanisms must measure behavioral outputs and decision patterns in addition to static artifacts.

Privilege Accumulation Through Tool Use. AI agents can incrementally expand effective privileges by chaining tool invocations in unanticipated sequences. Each individual call may be authorized; the composition produces unauthorized state changes. Threat models must account for action trajectories, not merely individual operations.

Non-Deterministic Governance Gaps. Traditional security assumes deterministic behavior. AI agents violate this assumption fundamentally. Governance must bind to policy-defined behavioral boundaries and generate cryptographic evidence of compliance or violation at each measurement point.

Our architecture addresses these threats under an explicit adversary model: the system assumes an adversary who possesses **full access to the local database and network environment** but does not possess the cryptographic signing keys of the issuing authority or the portal runtime. The system provides structural integrity guarantees independent of payload confidentiality. Even if event contents are intercepted, the chain's structural integrity and enforcement history remain cryptographically verifiable.

Response to Question 1(d): Evolution of Threats

These threats will intensify as AI agent systems are deployed in cyber-physical environments. Our patent application describes specific use cases: a SCADA control process where drift detection triggers quarantine, severing connections to physical actuators while logging attacker commands; and an autonomous vehicle whose flight control software triggers return-to-home enforcement upon drift detection, severing attacker control and executing safe-landing protocols.

The convergence of AI agent autonomy with physical-world actuation will create incidents where **the window between compromise and irreversible damage is measured in milliseconds**, making human-in-the-loop security responses inadequate. The enforcement decision must be pre-committed. This drives the requirement for autonomous enforcement actions encoded in sealed artifacts before deployment, not improvised after compromise.

2. SECURITY PRACTICES FOR AI AGENT SYSTEMS

Response to Question 2(a): Technical Controls

A robust security posture for AI agent systems requires what we term a **Sealed Governance Layer**: a system-level enforcement boundary operating independent of the specific AI model or framework. The following Attested Governance Artifact (AGA) framework serves as a **reference architecture** for this requirement, grounded in the Active Immutable Object pattern. We present it here not as a proprietary solution but as an implementable blueprint that CAISI may find useful in developing vendor-neutral guidelines. We are prepared to contribute the AGA schema to a standards development process under fair, reasonable, and non-discriminatory (FRAND) terms to facilitate rapid, vendor-neutral adoption.

Sealed Policy Artifacts as Active Governance Programs

Before an AI agent is authorized to execute, a sealed policy artifact is generated from a successful attestation of the agent's known-good state. This artifact is an immutable, cryptographically signed object encoding: the agent's identity binding (cryptographic hashes of normalized bytes and canonicalized metadata); a content-addressable reference to the governing policy; enforcement

parameters including measurement cadence, time-to-live (TTL), and enforcement triggers; a sealed hash value representing the known-good reference state; and a cryptographic signature binding all fields such that any modification is detectable.

To satisfy privacy requirements, attestation evidence is stored as **salted commitments**, computed as Hash(Content || Salt), rather than raw evidence. The original content remains with the owner and can be selectively revealed under policy-controlled disclosure rules. The artifact transforms passive compliance documentation into an active governance program.

Reference Schema: Policy Artifact Structure

```
{ "schema_version": "1.0",
  "subject_identifier": {
    "bytes_hash": "sha256:e3b0c4...",
    "metadata_hash": "sha256:7f83b1..." },
  "policy_reference": "sha256:2cf24d...",
  "sealed_hash": "sha256:9f86d0...",
  "enforcement": {
    "measurement_cadence_ms": 100,
    "ttl_seconds": 86400,
    "triggers": ["TERMINATE","QUARANTINE","SAFE_STATE"] },
  "signature": "ed25519:base64..." }
```

The Portal: A Mandatory Runtime Enforcement Boundary

The AI agent executes within a mandatory runtime boundary (termed a “portal” or “sentinel”) that intercepts interactions with external resources: tool calls, API invocations, actuator commands, and data access. The portal parses the sealed policy artifact and continuously measures the agent’s runtime state against the sealed reference using configurable measurement embodiments:

> *Executable image digests • Loaded module digests (DLLs, shared objects, kernel modules) • Container image digests • Configuration manifest digests • Software bill of materials (SBOM) digests • Trusted execution environment (TEE) quotes • Memory region samples • Control flow measurements • File system state digests • Network configuration digests*

Different measurement embodiments may be applied at different cadences: executable image verification on startup, configuration manifest every minute, memory sampling every second. Upon detecting drift, the portal automatically executes predetermined enforcement actions without human intervention.

Mitigating Irreversible Damage Through Graduated Enforcement and Phantom Execution

The architecture supports graduated enforcement responses calibrated to the deployment context. Enforcement may be implemented via physical mechanisms (severing power, hardware interlocks) or logical gating (API token revocation, routing table updates, port closure). Actions include: process termination;

quarantine with phantom execution; network isolation; key revocation; token invalidation; actuator disconnection; and safe-state transitions.

> **Phantom Execution (Quarantine Mode)**
>
> Upon drift detection, the portal transitions the compromised agent to a sandboxed phantom environment. All connections to protected resources (physical actuators, network endpoints, data stores) are severed. The agent continues executing, believing it is operating normally, while all outputs are captured rather than delivered. The portal continues delivering inputs, including attacker commands, to capture the full attack sequence. All attempted outputs and state changes are logged as forensic receipts. This transforms a security incident into an intelligence-gathering opportunity while containing damage.

Each enforcement action generates a signed receipt documenting the measurement results, the drift detected, and the action taken.

Tamper-Evident Accountability: Continuity Chains with Checkpoint Anchoring

All enforcement receipts are appended to an append-only continuity chain initialized with a genesis event containing a protocol version identifier, a root fingerprint derived from the chain's cryptographic key pair, and a content-addressable specification hash. Subsequent events are typed (POLICY_ISSUANCE, INTERACTION_RECEIPT, REVOCATION, ATTESTATION, ANCHOR_BATCH) and linked via structural metadata hashes.

The leaf hash for each event is computed using **only structural metadata**: schema version, protocol version, event type, event identifier, sequence number, timestamp, and previous leaf hash. Payload data is **deliberately excluded**. This enables third-party verification of chain integrity without requiring access to potentially sensitive payload contents. Payload integrity is independently protected through event signatures computed over the complete event including payload, plus a content-addressable payload hash stored within the event metadata.

The chain is periodically checkpointed by computing a Merkle root over batched events and anchoring that root to an immutable append-only storage network, preventing history rewriting even if the local database is compromised.

Implementation Maturity and Operational Considerations

This approach has reached a **working reference implementation with formal specification**. The cryptographic core (Ed25519 signatures, SHA-256/BLAKE2b hashing, HKDF-SHA256 key derivation, RFC 8785 JSON canonicalization) has been implemented and validated. Demonstration applications address SCADA process enforcement and autonomous vehicle command governance. The architecture is the subject of USPTO Application No. 19/433,835 with 20 claims covering runtime governance, sealed policy artifacts, enforcement boundaries, continuity chains, offline-verifiable evidence bundles, and privacy-preserving disclosure.

Continuous runtime measurement introduces computational overhead. We address this through configurable measurement cadences: high-frequency memory sampling (sub-second) for life-critical systems, lower-frequency configuration checks (minutes) for less consequential deployments. The tiered verification model (Section 3) further calibrates overhead to risk. Bronze-tier agents incur minimal latency from self-signed device attestation; Gold-tier agents accept higher overhead as the cost of blockchain-anchored assurance. For legacy systems lacking modern cryptographic primitives, the portal architecture operates as an external wrapper, intercepting I/O at the system boundary rather than requiring instrumentation of the monitored process itself.

A critical edge case: what happens when enforcement triggers during a life-critical operation? Consider a surgical procedure, an active flight maneuver, a chemical process at a critical transition point. The architecture addresses this through **safe-state transitions** (Claim 5 of USPTO Application No. 19/433,835) rather than hard termination. For autonomous vehicles, this means return-to-home or controlled landing. For industrial processes, controlled shutdown sequences. For medical devices, fail-to-last-known-good-state. The policy artifact specifies which enforcement action applies to which context. Termination is never the only option.

Response to Question 2(e): Relevant Cybersecurity Frameworks

Several existing frameworks provide partial coverage but lack runtime enforcement and cryptographic accountability:

Framework	Alignment / Gap	AGA Implementation
SP 800-53 Rev. 5	SI family addresses integrity but lacks continuous measurement against sealed references	Portal enforces autonomous runtime measurement with signed receipts
AI RMF (AI 100-1)	Measure function aligns with AGA lifecycle but prescribes no enforcement mechanisms	Operationalizes Measure through continuous hash comparison with cryptographic receipts
SP 800-218A	Addresses secure development, not post-deployment runtime enforcement	Provides the enforcement layer for software produced under SSDF practices
AI 100-2e2025	Taxonomic coverage of adversarial ML; focuses on model-level defenses	System-level enforcement boundary that contains damage when model defenses fail

A gap exists across all current frameworks: **none mandates cryptographic binding between a known-good state, continuous runtime measurement, and autonomous enforcement with tamper-evident accountability**. CAISI guidelines should incorporate requirements for sealed reference states, mandatory runtime measurement, and cryptographically committed enforcement evidence.

3. ASSESSING THE SECURITY OF AI AGENT SYSTEMS

Response to Question 3(a): Assessment Methods

Pre-Deployment Attestation. Before deployment, the agent's complete state-model artifacts, inference engine, tool configurations, dependency chain-should undergo cryptographic attestation against a defined policy, producing sealed reference hashes and salted evidence commitments for subsequent monitoring.

Continuous Runtime Measurement. The portal enables continuous assessment by computing runtime state hashes at policy-defined cadences. Each measurement generates a signed receipt-a verifiable assessment record detecting any deviation from the attested known-good state without requiring prior attack signature knowledge.

Offline Evidence Bundle Verification. Portable evidence bundles (per Claim 9) containing sealed artifacts, signed receipts, Merkle inclusion proofs, and checkpoint references enable cross-organizational verification without network connectivity-critical for defense and classified environments.

Response to Question 3(b): Assessing Particular Systems

Assessment should be calibrated through tiered verification levels:

Tier	Verification Method	Trust Assumption	Deployment Context
Bronze	Device-Attested (self-signed receipts)	Device private key not compromised	Low-consequence agents, lab testing, development
Silver	Server-Attested (central signing + key pinning)	Server infrastructure integrity; key pinning detects compromise	Enterprise agents, moderate-consequence deployments
Gold	Blockchain-Anchored (full Merkle proofs)	Immutable storage network provides consensus finality	Critical infrastructure, regulatory compliance, defense

Artifacts and receipts indicate their verification level, enabling relying parties to apply appropriate confidence thresholds.

4. DEPLOYMENT ENVIRONMENT CONTROLS

Response to Question 4(a): Constraining Deployment Environments

The portal operates as a **mandatory interception layer** mediating all interactions between the AI agent and external resources. Constraint mechanisms include: policy-defined resource authorization; configurable measurement cadences; enforcement triggers; and TTL-based artifact expiration requiring re-attestation.

The portal implements **fail-closed semantics**: if the policy artifact cannot be parsed, if the signature does not verify, if the effective period has expired, or if the initial measurement does not match the sealed reference, the agent is **not permitted to execute**. This inverts the traditional model where agents operate by default.

For intermittent connectivity environments, the architecture supports **graceful degradation** (per Claim 12): the portal maintains a cryptographically signed local cache of the active policy artifact, continues enforcement during connectivity loss for a pre-defined TTL period, and upon TTL expiration executes safe-state transition with local logging.

Response to Question 4(b): Modifying Environments to Mitigate Risk

Upon detecting drift, the portal transitions the agent to a sandboxed phantom environment (per Claim 11), maintaining forensic observation while containing damage. Selective quarantine isolates individual compromised agents while the broader system continues operating.

The continuity chain provides the evidentiary foundation for rollback decisions: the receipt chain identifies precisely when drift occurred, what enforcement actions executed, and what the agent attempted after compromise. This is the cryptographic evidence needed to determine *what* to roll back and *to which point* in time.

Response to Question 4(c): Managing Counterparty Interactions

Mechanical Systems and IoT (4(c)(iii)). For AI agents controlling physical actuators, enforcement includes actuator disconnection and safe-state transitions implemented via physical mechanisms (severing power, hardware interlocks) or logical gating (command routing, control signal interception).

Authentication Mechanisms (4(c)(iv)). The portal maintains pinned public keys for policy issuers and rejects artifacts with invalid signatures. Enforcement actions include key revocation and token invalidation.

Other AI Agent Systems (4(c)(v)). Our architecture employs a non-biometric identity model: AI agent identity is derived from cryptographic key pairs bound to an append-only attestation history. Authority and rank are derived from the history of valid signatures. This enables AI agents to hold cryptographic identity, authority, and verifiable governance history indistinguishable from human operators.

Response to Question 4(d): Monitoring Deployment Environments

Every measurement generates a signed receipt-match or mismatch. The result is an unbroken, tamper-evident record of the agent's governance posture across its entire operational lifetime. Structural metadata linking ensures that modification of any event invalidates all subsequent leaf hashes. An attacker who compromises the agent cannot also compromise the monitoring record. The receipt chain is **cryptographic proof of continuous monitoring**, independently verifiable by any party possessing the evidence bundle.

5. ADDITIONAL CONSIDERATIONS

Response to Question 5(a): Methods to Aid Rapid Adoption

Standardize Sealed Governance Artifact Formats. A machine-parseable, cryptographically signed schema would enable interoperability across governance providers, enforcement tools, and verification systems.

Establish Tiered Verification Requirements. CAISI should define tiered requirements matching governance rigor to deployment consequence, analogous to FIPS 140-2/3 security levels.

Mandate Offline Verifiability for Critical Infrastructure. Governance evidence must be verifiable using only locally available cryptographic material without real-time connectivity to any external service.

Require Privacy-Preserving Governance Disclosure. When a sensitive governance claim is requested but denied under the active policy, the system should automatically traverse an ordered substitute list, select the first permitted substitute of lower sensitivity, and generate a signed substitution receipt. Before disclosure, the system should verify that disclosed claims do not enable inference of denied claims.

Response to Question 5(b): Government Collaboration Priorities

Collaboration is most urgent in two areas. First, establishing reference architectures for runtime governance of AI agents in critical infrastructure (SCADA, autonomous vehicles, defense systems) where the consequence of compromise is physical and irreversible. These reference architectures would directly support agency compliance with **Executive Order 14110** (Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence), which mandates safety testing and red-teaming for AI systems with potential impact on critical infrastructure. The AGA framework provides the enforcement and accountability layer that EO 14110 envisions but does not prescribe. Second, developing interoperability standards for governance evidence (artifacts, receipts, evidence bundles) so that compliance posture can be verified across organizational boundaries without requiring identical technology stacks, enabling the vendor-neutral ecosystem that both NIST and the broader federal procurement framework require.

Response to Question 5(c): Research Priorities

Behavioral Integrity Measurement. Techniques that capture the agent's behavioral envelope-decision distributions, tool-use patterns, output characteristics-and bind those measurements to sealed references, extending binary drift detection to semantic drift detection.

Multi-Agent Governance. Governance artifacts specifying permitted interaction patterns between agents, enforcement boundaries for agent-to-agent communication, and continuity chains capturing cross-agent event sequences.

Quantum-Resistant Governance Primitives. Governance evidence must remain verifiable for decades. Post-quantum signature schemes (SPHINCS+, Dilithium) should be standardized for governance artifacts now. Our architecture's alternative embodiments already account for these as drop-in replacements.

CONCLUSION

AI agent systems operating with autonomous authority in physical-world environments represent a qualitatively different security challenge from traditional software. The Attested Governance Artifact architecture provides a concrete approach: sealed policy artifacts deployed as active governance programs that not only prove a state existed but **govern what states are permitted to exist**. The architecture delivers continuous measurement, autonomous enforcement, tamper-evident accountability, portable offline verification, and privacy-preserving disclosure, all bound by cryptographic commitments that an adversary with full database access cannot forge.

We welcome the opportunity to discuss these approaches further with CAISI and are available for technical briefings or standards development participation.

Respectfully submitted,

Jack Brennan
Attested Intelligence Holdings LLC
admin@attestedintelligence.com
AttestedIntelligence.com
March 4, 2026

USPTO Patent Application No. 19/433,835
USPTO Trademark Serial No. 99677085
Illinois File No. 17233815