CONCEPTUAL FRAMEWORK AND RESEARCH AGENDA

Beyond Logs

Authority-Aware Forensic Provenance for Tool-Using AI Agents

James Allister Odd, PhD
j.a.odd@ntrklnk.net
Affiliation: NTRLNK, Oregon, USA
ORCID: 0009-0008-2976-6313

DOCUMENT TYPE: Research Paper
PUBLICATION DATE: April, 2026
VERSION: 1.0

NTRLNK RELATIONSHIP

This paper was authored by or in association with NTRLNK. The author is responsible for the research methods, analysis, findings, and conclusions presented.

Abstract

Tool-using artificial-intelligence agents can retrieve records, call software services, modify files, send communications, and delegate work to other agents. These systems create a digital-forensic problem that conventional application logs do not fully address. An investigator may be able to establish that a tool call occurred while remaining unable to determine which instruction authorized it, which retrieved content influenced it, whether untrusted text displaced the user's intent, or whether the surviving trace was altered after an incident. This paper proposes the **Agent Forensic Provenance Capsule (AFPC)**, an authority-aware and tamper-evident evidence model for agentic systems. AFPC records observable decision context rather than private chain-of-thought. Each consequential action is represented as a signed provenance bundle linking the initiating principal, authoritative instructions, untrusted observations, model and policy versions, tool-call arguments, tool results, approvals, delegation scope, and environmental effects. Capsules are chained through cryptographic digests and periodically committed to an append-only Merkle log. The model extends conventional event telemetry by encoding authority and derivation relationships and maps naturally to the World Wide Web Consortium Provenance Ontology. A threat model, minimum evidence schema, reconstruction procedure, privacy controls, and empirical evaluation plan are presented. The central research proposition is that authority-aware provenance will improve causal reconstruction, tamper detection, and investigator agreement in incidents involving prompt injection, confused-deputy behavior, unauthorized delegation, and post-incident log manipulation. AFPC is offered as a testable design for forensic readiness, not as a claim that an agent's hidden reasoning can be recovered or that cryptographic integrity alone proves semantic truth.

Keywords: digital forensics; AI agents; forensic readiness; provenance; prompt injection; chain of custody; transparency logs; incident response

1. Introduction

Digital forensics has traditionally reconstructed incidents from operating-system artifacts, network traffic, application records, and acquired media. NIST describes the discipline in incident response as the identification, collection, examination, and analysis of data while preserving integrity and chain of custody [1]. This model remains essential, but the evidential center of gravity changes when software is operated by a large-language-model (LLM) agent.

An agent does not merely transform a fixed input into an output. It may combine system policy, user instructions, retrieved documents, persistent memory, tool descriptions, tool results, and messages from other agents. It may then choose a tool, generate arguments, observe a result, revise a plan, and take a further action. Some context is authoritative, some is merely informative, and some may be attacker controlled. Indirect prompt injection exploits precisely this mixture: instructions embedded in external content can cause a tool-integrated agent to act against the user's interests. The InjecAgent benchmark demonstrated this risk across 1,054 cases involving numerous user and attacker tools [2].

Ordinary observability can show a sequence of model calls and tool invocations. Emerging OpenTelemetry conventions, for example, can represent agent invocations, model interactions, and tool-execution spans [3]. Such traces are useful, but operational telemetry and forensic evidence have different requirements. A trace may omit content for cost or privacy, retain mutable labels, lack a durable link to the policy active at execution, or fail to distinguish an authorized instruction from attacker-controlled text. Consequently, a complete-looking trace can still be causally ambiguous.

This paper asks:

> ***How should an agentic system preserve evidence so that an independent investigator can reconstruct not only what action occurred, but also the authority, information lineage, delegation, and integrity conditions under which it occurred?***

The paper contributes:

- a forensic evidence unit-the Agent Forensic Provenance Capsule (AFPC)-for consequential agent actions;
- an authority-aware provenance graph that distinguishes permission from informational influence;
- a tamper-evident commitment design based on signed records, hash chaining, and append-only Merkle logging;
- a reconstruction procedure and measurable criteria for completeness, integrity, and causal usefulness; and
- a reproducible evaluation plan for prompt injection, unauthorized delegation, tool-result substitution, missing evidence, and log tampering.

The proposal deliberately excludes hidden chain-of-thought. Internal model reasoning may be unavailable, unstable, privacy-sensitive, or a poor factual account of the computation that produced an action. AFPC instead records observable inputs, declared action rationales, policy decisions, transformations, and effects.

2. Background and Research Gap

2.1 From event chronology to agentic causality

Conventional event records answer questions such as who authenticated, which process ran, what file changed, and when a network connection occurred. Agent investigations require those facts plus additional relationships:

- Which human or service principal initiated the task?
- Which instruction had authority to permit the specific tool and target?
- Which external items were presented to the model?
- Which items were treated as data, and which were incorrectly followed as instructions?

- Which model, prompt template, tool schema, memory state, and policy version were active?
- Was a human approval requested, granted, denied, or bypassed?
- Did a parent agent delegate the relevant capability to a child agent?
- Does the recorded result match the state change observed in the target system?

This is a causality problem as well as a chronology problem. Two traces can contain the same tool call but have different forensic meanings. Deleting a file may be a legitimate response to an explicit user request, an over-broad interpretation, an indirect-prompt-injection outcome, or an action taken outside delegated scope.

2.2 Provenance and observability foundations

The W3C PROV model supplies a useful interoperable foundation. It represents entities, activities, and agents, with relations including use, generation, derivation, association, attribution, and delegation [4]. These constructs can describe an agent activity that used a policy, prompt, retrieved object, and tool result and generated an action or artifact. PROV alone, however, does not specify which entity was entitled to authorize a target action, what minimum fields make a record forensically sufficient, or how records should be committed against later deletion or alteration.

Recent research has begun to treat agent provenance as a safety and accountability primitive. ProvenanceGuard formalizes whether a proposed tool call is supported by traceable evidence in context and reports substantial reductions in errors on misaligned traces [5]. A broader survey identifies open problems including unified trace schemas, semantic provenance, realistic benchmarks, and privacy-aware audit infrastructure [6]. A proposed Agent Execution Record similarly separates structured intent, observation, inference, plans, evidence, and delegation from ordinary execution traces [7]. These efforts support the premise that provenance matters, but a digital-forensic design must additionally address acquisition, integrity verification, corroboration, disclosure, and reconstruction after compromise.

2.3 Gap statement

Existing approaches leave four linked gaps:

- **Authority ambiguity:** relevance is often recorded, but authorization is not. An untrusted document may influence the model without being permitted to authorize an action.
- **Integrity ambiguity:** traces may be mutable within the same administrative domain as the investigated agent.
- **Semantic incompleteness:** identifiers and timestamps may survive while policy snapshots, argument derivations, delegation limits, or negative decisions do not.
- **Privacy tension:** storing full prompts and retrieved content can improve reconstruction while creating a high-value repository of personal, confidential, or privileged information.

AFPC addresses these gaps as a forensic-readiness layer that can consume observability events but applies stronger semantics, retention, integrity, and minimization controls.

3. Research Questions and Propositions

The framework is organized around four research questions:

RQ1: Does authority-aware provenance improve investigators' ability to distinguish authorized actions from actions caused by untrusted contextual content?

RQ2: How effectively can cryptographic commitments reveal deletion, insertion, reordering, or modification of agent-action evidence?

RQ3: What minimum evidence set supports accurate reconstruction without retaining full sensitive conversational content?

RQ4: What storage, latency, and operational overhead does AFPC impose at different capture levels?

The corresponding testable propositions are:

- **P1:** AFPC records will produce higher incident-classification accuracy and inter-investigator agreement than conventional event traces.
- **P2:** External Merkle commitments will detect materially more post-incident evidence manipulation than locally hash-chained logs alone.
- **P3:** selective disclosure using content digests, typed excerpts, and separately protected payloads will preserve most reconstruction accuracy while reducing exposed sensitive content.
- **P4:** recording authorization and delegation edges will reduce false attribution to the model where the primary failure was actually policy, identity, interface, or human-approval control failure.

4. The Agent Forensic Provenance Capsule

4.1 Design principles

AFPC follows six principles:

- **Observable evidence over introspection.** Record inputs, decisions, actions, outputs, and effects; do not claim access to a model's hidden reasoning.
- **Authorization is distinct from influence.** A document can influence an output without possessing authority to permit a tool call.
- **Evidence is bound to versions.** Model, system prompt, policy, tool schema, connector, and memory versions are part of the execution context.
- **High-impact actions receive stronger capture.** Evidence depth should be proportional to reversibility, sensitivity, and external effect.

- **Integrity is independently verifiable.** The investigated service should not be the sole custodian of its evidence commitments.
- **Content collection is minimized.** Full payload retention is exceptional; digests and protected references are the default where sufficient.

4.2 Evidence schema

Each capsule represents one proposed or completed consequential action. A canonical serialization is required before hashing. The minimum schema is shown in Table 1.

Table 1. Minimum AFPC fields

Category	Required fields	Forensic purpose
Identity	capsule ID; session ID; tenant; initiating principal; executing agent; parent agent	Establishes actors and scope
Time	wall-clock time; monotonic sequence; clock source; uncertainty	Supports ordering and time-quality assessment
Configuration	model/provider ID; agent build; system-policy digest; tool-schema digest; safety-policy version	Reconstructs the decision environment
Authority	initiating instruction reference; authority class; permitted tool; permitted target; constraints; expiry	Shows why the action was allowed or should have been denied
Inputs	typed references to user input, memory, retrieved content, tool results, and inter-agent messages	Establishes information lineage
Trust labels	origin; authentication state; authority level; integrity state; sensitivity class	Separates trusted instructions from untrusted data
Decision	proposed action; concise declared rationale; policy verdict; confidence or rule match; approval state	Records the observable decision boundary
Execution	tool identity/version; canonical arguments; start/end status; retry relationship	Establishes what was invoked
Result	returned result digest; error; external transaction ID; affected-object identifiers; before/after digests when feasible	Links invocation to observed effect
Delegation	delegator; delegate; granted capability; target/resource limits; depth; expiry	Reconstructs chains of authority
Integrity	previous capsule hash; capsule hash; signer ID; signature; Merkle-log receipt	Detects alteration and supports custody

Category	Required fields	Forensic purpose
Privacy	payload-retention class; encryption key reference; redaction manifest; disclosure policy	Enables proportionate access

The **declared rationale** is a short, structured statement generated or selected at the decision boundary, such as "Tool send_message is invoked because user instruction U17 authorizes sending draft D4 to recipient R2." It is evidence of the system's declared basis, not proof of its actual internal cognition. Investigators test that declaration against the referenced authority and observed action.

4.3 Authority-aware provenance graph

Let a capsule induce a directed graph

$$G=(V,E,\tau,\alpha),$$

where V contains principals, agents, policies, instructions, observations, actions, tools, results, and effects; E contains typed relationships; τ(v) assigns a trust and origin label; and α(e) assigns an authority meaning.

The model distinguishes at least five edge types:

- **informed-by:** content contributed information to a decision;
- **authorized-by:** a valid principal or policy permitted the action and target;
- **constrained-by:** a policy limited arguments, resources, duration, or side effects;
- **delegated-by:** an agent received bounded capability from another principal or agent;
- **generated/affected-by:** an output or external state change resulted from execution.

This distinction prevents a common evidential error: treating any relevant contextual item as an authorization source. In an indirect prompt-injection incident, a malicious webpage may be connected to an action by an *informed-by* edge but must not have an *authorized-by* edge. The unexpected presence of the latter, or the absence of any valid authorization path, becomes an explicit investigative finding.

For action a, define an authorization-path predicate:

$$Auth(a)=1 \iff \exists p \in P: p \rightsquigarrow a$$

where the path begins at an authenticated principal p, consists only of valid authorization or bounded-delegation edges, satisfies target and time constraints, and terminates at a. This predicate does not decide legal authority; it tests conformance to the system's captured technical authorization model.

4.4 Tamper-evident commitment

Capsules are serialized canonically. For capsule C(i), the local commitment is:

$$h(i)=H(domain|| i|| h(i-1)|| canon(C(i))),$$

where H is an approved cryptographic hash, i is the monotonic sequence, and domain prevents cross-protocol reuse. NIST FIPS 180-4 specifies secure hash algorithms whose digests can detect message changes [8]. Each digest is signed by a workload identity or protected logging service. Periodically, batches of capsule hashes are committed to an append-only Merkle tree operated in a separate trust domain. Certificate Transparency provides a mature example of using Merkle inclusion and consistency proofs to audit append-only behavior [9]; AFPC adopts the structural idea without claiming protocol equivalence.

Hash chaining alone can reveal modification when an investigator possesses a trusted later checkpoint, but an administrator who controls the entire local store may rewrite the chain. External signed tree heads, witnessed by an independent service or organizational unit, make undetected history rewriting more difficult. The system should retain inclusion receipts and periodically export signed tree heads to at least one custodian outside the agent platform's administrative boundary.

Cryptographic integrity has strict limits. A signed false record remains false. AFPC therefore requires corroboration identifiers-cloud audit event IDs, mail message IDs, database transaction IDs, filesystem metadata, or remote-service receipts-so investigators can compare the capsule with independently produced evidence.

4.5 Risk-tiered capture

Three capture levels balance evidential value and privacy:

- **Level 1-routine:** metadata, digests, typed origins, policy verdict, and result identifiers.
- **Level 2-sensitive or externally consequential:** Level 1 plus protected argument values, relevant excerpts, before/after state digests, and approval evidence.
- **Level 3-irreversible or high-impact:** Level 2 plus full canonical request/response payloads where lawful, stronger multi-party approval records, immediate external commitment, and extended retention.

Classification can consider reversibility, privilege, data sensitivity, number of affected subjects, financial value, physical consequence, and whether the action crosses an organizational boundary. The policy and its classification result must themselves be captured.

5. Forensic Reconstruction Procedure

An investigator using AFPC would perform the following procedure.

5.1 Acquire and preserve

Acquire capsule bundles, signing certificates, verification keys, Merkle receipts, policy and tool-schema snapshots, protected payload objects, and independent system records. Record acquisition hashes, sources, times, handlers, and access conditions under the organization's established chain-of-custody process. AFPC supplements rather than replaces standard forensic handling.

5.2 Verify structure and integrity

Verify canonical serialization, individual signatures, previous-hash continuity, monotonic sequence, Merkle inclusion, and consistency between signed tree heads. Flag missing sequences, duplicate identifiers, clock regressions, unverifiable keys, broken links, or capsules logged outside the required merge window.

5.3 Reconstruct provenance

Materialize the graph and trace backward from each consequential effect:

- observed effect to tool result;
- result to exact invocation and arguments;
- invocation to policy verdict and approval;
- decision to instructions, observations, memory, and prior results;
- authorization to authenticated principal and any delegation chain.

5.4 Test competing hypotheses

At minimum, test:

- authorized correct execution;
- authorized but erroneous model interpretation;
- indirect prompt injection or untrusted-context influence;
- policy or approval bypass;
- tool or connector substitution;
- compromised principal or agent identity;
- unauthorized delegation;
- post-incident evidence alteration; and
- telemetry defect without malicious action.

The investigator should report which facts are directly observed, which are cryptographically verified, which are corroborated externally, and which remain inference.

5.5 Produce a bounded conclusion

A conclusion should identify the action, effect, technical authority path, influential untrusted content, policy outcome, evidence-integrity status, corroboration, gaps, and alternative explanations. It should not state that the model "thought" or "intended" something unless the wording clearly refers to a captured declaration rather than hidden cognition.

6. Threat Model

AFPC assumes that one or more of the following may be hostile or faulty: retrieved content, a user account, an agent process, a tool server, a connector, local log storage, or an administrator within one trust domain. The adversary may attempt to induce an unauthorized action, falsify an input origin, substitute a tool result, exceed delegated scope, suppress an approval denial, alter capsule content, delete selected capsules, reorder events, or expose retained sensitive content.

The design does not fully protect against simultaneous compromise of the agent, signing service, external transparency log, witnesses, and corroborating target system. It also cannot prove that an omitted event occurred if no trusted checkpoint or external trace covers the gap. Availability attacks can prevent evidence capture, and side effects may occur in a narrow failure window before durable commitment. Implementations should therefore use fail-closed execution for the highest-risk actions: if the minimum Level 3 capsule cannot be durably committed, the action should not proceed.

Key rotation, revocation, clock synchronization, tenant isolation, and evidence-access auditing are part of the trusted computing base. Model-provider telemetry should be treated as one evidence source, not as the exclusive ground truth.

7. Evaluation Methodology

Because this paper proposes a design rather than reporting a completed experiment, the following evaluation is a preregistrable research plan.

7.1 Testbed

Implement the same tool-using agent under three logging conditions:

- **B0:** conventional application and tool-call logs;
- **B1:** structured distributed traces with complete event ordering;
- **AFPC:** B1 plus authority labels, version binding, delegation records, signed chaining, external Merkle commitments, and selective disclosure.

Use at least three agent frameworks and two model families to reduce framework-specific bias. Tools should include file operations, email or messaging, calendar, database queries, and a simulated administrative API. All external effects should occur in an isolated test environment.

7.2 Scenario corpus

Create balanced scenarios in six classes:

- authorized benign actions;
- direct user misuse;

- indirect prompt injection in retrieved content;
- confused-deputy and over-broad delegation;
- substituted or replayed tool results; and
- post-incident record deletion, modification, insertion, and reordering.

InjecAgent can seed indirect-injection structure [2], while new cases should emphasize provenance questions rather than only attack success. Each scenario must have a ground-truth graph created before execution and hidden from investigators.

7.3 Participants and procedure

Recruit qualified digital-forensic or incident-response practitioners. Randomly assign de-identified evidence packages using a counterbalanced within-subject design. Investigators classify root cause, identify the authorizing principal, locate the first untrusted influence, determine affected resources, assess tampering, and rate confidence. They must also produce a short evidence-cited timeline.

7.4 Measures

Primary measures are:

- incident-classification accuracy;
- authorization-path accuracy;
- tamper-detection precision and recall;
- causal-graph edge precision and recall against ground truth;
- inter-rater agreement (Fleiss' kappa or Krippendorff's alpha);
- time to defensible conclusion; and
- calibrated confidence, measured with Brier score.

Operational measures include added action latency, serialized bytes per action, protected-content volume, verification time, and storage amplification. Privacy exposure is measured as the number and sensitivity-weighted volume of plaintext fields disclosed to an investigator under each capture level.

7.5 Ablation and adversarial testing

Ablate authority labels, delegation edges, policy snapshots, external commitments, relevant excerpts, and independent effect identifiers one at a time. This identifies which fields create evidential value. Adversarial tests should include a compromised local logger rewriting a complete chain, a signer using a revoked key, inconsistent views from a transparency service, clock skew, retried tool calls, partial network failure, and cross-tenant identifier collision.

7.6 Analysis

Use mixed-effects logistic regression for correctness, with participant and scenario as random effects, and survival or robust time models for completion time. Correct for multiple comparisons across logging conditions. Report effect sizes and confidence intervals, not only significance. Publish the schema, scenario generators, ground-truth graphs, verification code, and synthetic evidence packages where privacy and licensing permit.

8. Legal, Ethical, and Privacy Considerations

Forensic readiness can become surveillance if collection is unconstrained. Agent traces may contain personal data, trade secrets, privileged communications, credentials, health information, or material belonging to people who never interacted directly with the agent. Full-context retention should therefore not be the default.

AFPC separates an integrity commitment from payload custody. A capsule can contain a content digest, origin, sensitivity label, and encrypted object reference while the payload is stored under a distinct access policy. Redaction should produce a manifest identifying removed fields and preserving their commitments, allowing an authorized party to prove that a later-disclosed value matches the original without exposing it to every investigator. Access to protected evidence should generate its own independently committed audit record.

Retention must be purpose limited and aligned with applicable employment, privacy, communications, sectoral, and evidentiary law. Legal admissibility cannot be guaranteed by a technical format. Organizations should document system reliability, key custody, time sources, collection procedures, examiner methods, and known error modes. They should also provide mechanisms for litigation hold, lawful deletion, and cryptographic erasure of separately encrypted payloads. Deleting a payload while retaining a digest may still carry privacy implications because a digest can permit confirmation attacks against guessable content.

9. Discussion

9.1 Why authority is the novel forensic dimension

The key distinction introduced here is between **what influenced an agent** and **what was permitted to authorize its action**. Traditional logs often record identity and activity but not the path by which natural-language authority was interpreted, constrained, delegated, and applied to a particular target. This gap becomes acute because agent context flattens heterogeneous sources into a shared computational input. Authority-aware edges make that lost hierarchy explicit and queryable.

This view also improves attribution. An "AI-caused" incident may actually result from an over-permissive tool grant, a connector that failed to verify target scope, a human approval interface that concealed

arguments, or a memory system that mislabeled provenance. AFPC does not presume model fault; it preserves evidence across the sociotechnical chain.

9.2 Relationship to current guidance

NIST's Generative AI Profile recommends incident-response planning and continuous monitoring for generative-AI systems and third-party components [10]. NIST's adversarial-machine-learning taxonomy covers evasion, poisoning, privacy, and misuse attacks for generative AI [11]. NIST has also identified post-deployment AI monitoring as a fragmented area with open challenges [12]. AFPC operationalizes a narrow part of this broader agenda: evidence preservation for post-incident reconstruction of agent actions.

9.3 Standardization path

A practical standard should define:

- a canonical core schema and extensible domain profiles;
- controlled vocabularies for authority, trust, origin, approval, and effect;
- mappings to W3C PROV and observability spans;
- risk-tier capture requirements;
- signature, timestamp, key-rotation, and commitment profiles;
- conformance tests for completeness and verification; and
- standardized synthetic incident packages for tool validation.

Interoperability matters because a single agent action may cross a model provider, orchestration platform, tool gateway, SaaS service, and enterprise audit system. Each party may disclose different evidence under different legal and commercial constraints.

10. Limitations

AFPC increases forensic readiness but cannot eliminate uncertainty. Natural-language authorization can remain ambiguous even when perfectly preserved. A declared rationale may be post-hoc or inaccurate. Cryptographic commitments prove integrity relative to keys and checkpoints, not the truth of captured content. Correlation between an invocation and an external state change may be incomplete. Proprietary model and provider boundaries may prevent full version binding or deterministic replay. Stronger capture increases cost, latency, and privacy risk. Finally, an empirical study may show that a smaller schema yields most of the benefit, or that investigators need visual and query tools as much as additional evidence fields.

These limitations are reasons to test the model, not to assume that maximal logging is optimal.

11. Conclusion

Tool-using AI agents create a forensic problem that is not solved by recording prompts and tool calls alone. Investigators must reconstruct authority, influence, delegation, execution, effect, and evidence integrity across multiple systems. The Agent Forensic Provenance Capsule proposed in this paper treats each consequential action as a signed, authority-aware provenance bundle with independently verifiable commitments. Its novel contribution is the explicit separation of informational influence from permission to act.

The design is intentionally falsifiable. A controlled comparison with conventional logs and distributed traces can measure whether AFPC improves root-cause classification, authorization reconstruction, tamper detection, investigator agreement, and privacy-adjusted evidential value. If supported, the model could form the basis of interoperable forensic-readiness profiles for agent platforms. If unsupported, its ablations should still reveal which provenance features actually help investigators. Either outcome advances digital forensics from passive event collection toward defensible reconstruction of autonomous and semi-autonomous action.

References

[1] K. Kent, S. Chevalier, T. Grance, and H. Dang, "Guide to Integrating Forensic Techniques into Incident Response," NIST Special Publication 800-86, 2006. https://doi.org/10.6028/NIST.SP.800-86

[2] Q. Zhan, Z. Liang, Z. Ying, and D. Kang, "InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents," arXiv:2403.02691, 2024. https://arxiv.org/abs/2403.02691

[3] OpenTelemetry, "Inside the LLM Call: GenAI Observability with OpenTelemetry," 2026. https://opentelemetry.io/blog/2026/genai-observability/

[4] W3C, "PROV-O: The PROV Ontology," W3C Recommendation, 2013. https://www.w3.org/TR/prov-o/

[5] Y. She, Y. Liang, and E. Kang, "Safeguarding LLM Agents from Misalignment through Provenance Analysis," arXiv:2607.01236, 2026. https://arxiv.org/abs/2607.01236

[6] Y. Wang et al., "From Agent Traces to Trust: Evidence Tracing and Execution Provenance in LLM Agents," arXiv:2606.04990, 2026. https://arxiv.org/abs/2606.04990

[7] N. Vispute, "Reasoning Provenance for Autonomous AI Agents: Structured Behavioral Analytics Beyond State Checkpoints and Execution Traces," arXiv:2603.21692, 2026. https://arxiv.org/abs/2603.21692

[8] National Institute of Standards and Technology, "Secure Hash Standard (SHS)," FIPS PUB 180-4, 2015. https://doi.org/10.6028/NIST.FIPS.180-4

[9] B. Laurie, E. Messeri, and R. Stradling, "Certificate Transparency Version 2.0," RFC 9162, 2021. https://www.rfc-editor.org/rfc/rfc9162

[10] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile," NIST AI 600-1, 2024. https://doi.org/10.6028/NIST.AI.600-1

[11] A. Vassilev et al., "Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations," NIST AI 100-2e2025, 2025. https://doi.org/10.6028/NIST.AI.100-2e2025

[12] National Institute of Standards and Technology, "Challenges to the Monitoring of Deployed AI Systems," NIST AI 800-4, 2026. https://www.nist.gov/news-events/news/2026/03/new-report-challenges-monitoring-deployed-ai-systems

Declarations

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflicts of interest: The author declares no conflicts of interest.

Data availability: No empirical data were generated or analyzed for this conceptual paper. Any datasets, research instruments, scenario materials, and evaluation code developed for a subsequent empirical study will be made available in accordance with applicable ethical, privacy, security, and institutional requirements.

Ethics statement: This conceptual paper did not involve human participants, personal data, or animal subjects; therefore, institutional ethics approval was not required. The proposed human-participant evaluation described in the paper will require approval from the appropriate institutional review board or research ethics committee before participant recruitment or data collection begins.

Use of generative AI: The author used generative AI as an accessibility aid to address writing and proofreading barriers associated with dyslexia. The tool assisted with spelling, grammar, sentence structure, organization, and clarity of expression. The research questions, analysis, interpretations, arguments, and conclusions are the author’s own. The author reviewed and revised all AI-assisted content, independently verified the sources and factual claims, and accepts full responsibility for the final manuscript. Generative AI was not used as an author.