DIGITAL FORENSICS • AI SECURITY • INVESTIGATIVE METHODOLOGY

When Software Acts

Counterfactual Reconstruction for Investigating Autonomous 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: August, 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

Autonomous artificial-intelligence agents complicate a foundational investigative question: who or what caused the consequential act? Unlike conventional software, an agent may interpret an open-ended goal, retrieve external content, maintain mutable memory, select tools, delegate to other agents, revise a plan, and act through accounts carrying human authority. The resulting event is neither adequately explained by a user's initial prompt nor by a final model-generated rationale. It emerges from a distributed and partly stochastic system in which instructions, data, identity, authorization, model behavior, orchestration code, tool responses, and human interventions interact over time. Conventional digital forensics remains necessary, but its familiar emphasis on endpoints, files, network events, and chronological reconstruction does not by itself resolve agentic causation.

This conceptual article proposes **Counterfactual Agent Reconstruction (CAR)**, an investigation methodology for incidents involving tool-using and multi-agent AI systems. CAR separates preservation from interpretation and organizes evidence across ten planes: mandate, identity, authority, perception, memory, model and configuration, planning, tool execution, environment response, and human governance. Its central innovation is a disciplined transition from observable action reconstruction to controlled counterfactual testing. Investigators first build a signed, time-normalized delegation-and-action graph from primary records. They then replay the incident in an isolated environment, varying one material factor at a time to test whether the harmful outcome depends on a malicious external instruction, poisoned memory, a particular model or policy version, excessive privilege, tool ambiguity, inter-agent message, or missing approval gate. Replay does not reproduce intent and cannot convert probability into historical certainty; it is used to discriminate among causal hypotheses and measure system sensitivity.

The article defines evidentiary confidence levels, addresses non-determinism and missing chain-of-thought, distinguishes authorization from influence, and proposes reporting rules that prevent an agent's self-explanation from being treated as ground truth. It also examines legal admissibility, privacy, privilege, vendor dependency, anti-forensics, and the risk of investigative overreach. Finally, it proposes an Agentic Forensic Readiness Standard, a synthetic benchmark, and a research agenda for validating the method. The paper argues that investigators must move from asking only “what executed?” to asking “what delegation, observation, state change, and control failure made this action possible—and what minimal change would have prevented it?”

Keywords: AI agents; agentic systems; digital forensics; incident investigation; causal inference; counterfactual replay; prompt injection; provenance; multi-agent systems; attribution; audit logging

1. Introduction

Digital investigation has traditionally reconstructed human and software activity from artifacts that were assumed to be relatively stable: a file was created, a credential was used, a process executed, a packet crossed a boundary, or a user clicked a control. Autonomous AI agents disturb that stability. An agent can receive a broad objective, choose intermediate steps, gather new information, invoke tools, communicate with other agents, and continue acting without a person approving each transition. NIST describes agents as systems capable of planning and taking autonomous actions that affect real-world systems or environments, and in 2026 launched an AI Agent Standards Initiative focused on interoperable, secure, and trustworthy adoption [1,2].

The investigative consequence is larger than the arrival of another software category. Agentic systems collapse distinctions on which attribution often depends. Data may function as instruction. A document retrieved for analysis may contain an indirect prompt that redirects the agent. An email that appears to have been sent by an employee may have been composed, addressed, and transmitted by an agent using the employee's delegated token. A destructive command may originate in a model-generated tool call, but the conditions that made it effective may lie in excessive permissions, an ambiguous tool schema, a poisoned memory entry, an orchestration retry, or a human approval interface that concealed the true scope.

NIST's 2026 summary of public responses on agent security reports broad agreement that fundamental cybersecurity practices remain relevant but require adaptation for agent systems [3]. The challenge is therefore not to discard digital forensics. It is to extend it. Disk, memory, cloud, identity, application, network, and log evidence still matter. Yet an investigator must also reconstruct the agent's mandate, observations, evolving state, available action space, inter-agent delegation, and governance controls.

This article proposes Counterfactual Agent Reconstruction (CAR). CAR is designed for criminal, civil, regulatory, workplace, national-security, and internal investigations in which an AI agent may have caused, amplified, concealed, or merely recorded a consequential event. It is not a standard of law and does not assume that every autonomous action is an “AI incident.” Its purpose is methodological: to help an investigator identify what is known, what is inferred, what can be experimentally tested, and what remains irreducibly uncertain.

The principal thesis is that agentic investigation requires two distinct reconstructions. The first is **historical reconstruction**: a source-grounded account of what the deployed system observed and did. The second is **counterfactual reconstruction**: controlled testing of which material conditions were necessary, sufficient, or strongly contributory to the outcome. Neither reconstruction alone is adequate. Logs without testing can describe an opaque sequence without explaining why it changed direction. Replay without trustworthy historical evidence can produce an impressive simulation of an event that never occurred.

2. What Makes an Agentic Incident Different?

2.1 From deterministic execution to bounded autonomy

Traditional programs may be complex, but investigators can often relate execution to code paths, configuration, and inputs. A modern agent is commonly assembled from a model, system instructions, orchestration logic, memory, retrieval, tool definitions, credentials, policies, and external services. It operates within an action space rather than a single predetermined sequence. OpenAI's early governance framework described agentic systems as systems that pursue complex goals with limited direct supervision and emphasized constrained action spaces, activity legibility, monitoring, attributability, and interruptibility [4].

Autonomy is not binary. An agent may only recommend actions, may execute reversible actions automatically, or may possess broad authority to transact, deploy code, change access, or communicate externally. The investigator must measure actual autonomy at the time of the incident, not accept product labels such as “assistant,” “copilot,” or “fully autonomous.”

2.2 Instructions and evidence share a channel

Indirect prompt injection exploits the agent's exposure to untrusted content. NIST describes agent hijacking as a condition in which malicious instructions embedded in data ingested by an agent cause unintended harmful action [5]. AgentDojo demonstrates the problem in a dynamic benchmark containing realistic tasks, tools, and injection attacks [6]. The forensic novelty is that the same retrieved page, message, attachment, or database record can be both evidence considered by the agent and an instruction that changes its behavior.

An investigator therefore cannot classify an artifact solely by where it entered the system. The relevant question is how the agent processed it. Was the content placed in a privileged instruction channel, a tool result, retrieval context, long-term memory, or another agent's message? Did the orchestrator label its trust level? Did a parser collapse structured data into free text? Did the model have a policy for resolving conflicting instructions? These routing facts may be more causally important than the artifact's visible wording.

2.3 State is distributed and partially ephemeral

Agent state can span a conversation window, scratchpad, vector store, task queue, browser session, local files, orchestration database, tool-service logs, and identity provider. Some state may be summarized or compacted; some may be overwritten after each turn. An agent may use a fresh model invocation for every step while an external controller maintains continuity. A multi-agent system may distribute fragments of the incident across several services owned by different vendors.

The absence of a single “agent memory” means that acquisition must be architectural. Imaging one endpoint may preserve almost none of the decisive state. Investigators need service maps, retention rules, version histories, and access to the systems that assembled each model context.

2.4 Model behavior is stochastic and version-sensitive

Even when inputs appear identical, sampling, routing, hidden system changes, safety layers, hardware, and model updates can change outputs. A provider may silently update a hosted model behind a stable name. The system prompt may be generated dynamically. Retrieval results may depend on an index that has since changed. Tool availability or rate limits may alter the plan.

This does not make investigation impossible, but it changes the meaning of reproduction. A replay that produces the same action once is not proof that the historical system reasoned the same way. A replay that fails to reproduce the action once is not proof that the historical action was fabricated. Agentic reconstruction requires repeated trials, version pinning where possible, and explicit reporting of reproduction rates and environmental differences.

2.5 Delegation obscures agency

An agent often acts using authority derived from a human, service account, application, or upstream agent. NIST's 2026 concept paper on software and AI-agent identity highlights identification, authorization, auditing, non-repudiation, ephemeral task-dependent identity, and prompt-injection controls as emerging design questions [7]. The investigator must distinguish at least four relationships:

1. **requested by:** who expressed the objective;
2. **authorized by:** whose credential, policy, or approval permitted the act;
3. **selected by:** which agent or controller chose the act;
4. **executed by:** which tool, service, or process changed the external environment.

These relationships may point to different actors. Collapsing them into a single “user” field destroys causally important information.

2.6 The agent's explanation is evidence, not ground truth

An agent may generate a rationale, summary, or post hoc explanation. That output can help locate hypotheses, but it is not a transparent readout of hidden cognition. The explanation may be incomplete, optimized for plausibility, reconstructed after tool execution, or influenced by the question asked. Investigators should preserve it as a contemporaneous statement generated by the system while testing it against primary records. A confident narrative is not a substitute for provenance.

3. The Investigative Object: An Evidence System, Not a Chat Transcript

The chat transcript is often the most visible artifact and one of the least complete. It may omit system instructions, tool parameters, failed attempts, hidden policy interventions, retrieved content, authentication exchanges, background jobs, memory writes, and messages exchanged among agents. The correct unit of analysis is the **agentic evidence system**: the set of human, model, software, identity, data, and environmental components that collectively produced the action.

CAR organizes this system into ten evidence planes.

3.1 Mandate plane

The mandate plane records the objective assigned to the system and the boundaries communicated with it. Relevant evidence includes the user's original instruction, task template, system and developer instructions, organizational policy, workflow definition, time limits, cost limits, prohibited actions, and escalation conditions. Investigators should preserve the exact text and structured representations delivered to the agent, not merely the human-facing display.

3.2 Identity plane

The identity plane identifies human users, agents, service principals, sessions, devices, tenants, and downstream services. It includes identity-provider records, agent instance identifiers, session IDs, workload identities, certificate subjects, token issuers, and mappings between product-level labels and actual principals.

3.3 Authority plane

The authority plane describes what each principal could do. Evidence includes token scopes, role assignments, consent grants, delegated permissions, policy decisions, temporary elevation, approval records, secrets access, sandbox boundaries, and tool-specific restrictions. Effective privilege must be reconstructed at the incident time; current configuration may differ.

3.4 Perception plane

The perception plane contains what the agent received: user messages, retrieved documents, search results, sensor data, screenshots, tool results, API responses, other-agent messages, and environmental observations. Each item should retain source, trust label, retrieval query, rank, transformation, truncation, and placement within model context.

3.5 Memory plane

The memory plane includes conversational context, summaries, scratch state, vector-store entries, persistent preferences, episodic memory, task queues, and caches. Investigators should identify who or what wrote each memory, when it was retrieved, whether it was editable by untrusted content, and whether later compaction altered meaning.

3.6 Model-and-configuration plane

This plane records provider, model identifier, model snapshot where available, sampling parameters, safety settings, routing logic, prompt templates, orchestration version, tool schemas, retrieval configuration, and feature flags. Container images, package manifests, source commits, deployment records, and signed build attestations may establish lineage.

3.7 Planning plane

The planning plane contains observable representations of intended intermediate steps: task graphs, planner outputs, queued actions, model-request envelopes, policy checks, and generated tool-call proposals. Hidden chain-of-thought should not be assumed to exist or be collectible. CAR focuses on operationally material representations that crossed a component boundary or caused a state change.

3.8 Execution plane

The execution plane records tool calls, parameters, credentials used, code execution, API requests, database operations, messages sent, files changed, transactions initiated, and retry behavior. Primary tool and service logs should be preferred over agent-generated summaries.

3.9 Environment-response plane

The environment may accept, reject, modify, queue, or partially apply an action. Evidence includes API response codes, resulting state, downstream automation, asynchronous callbacks, rate limits, transaction receipts, and compensating actions. A generated tool call that failed is not equivalent to a completed act.

3.10 Human-governance plane

This plane records approvals, denials, overrides, alerts, monitoring decisions, incident response, and opportunities to intervene. The interface shown to the approver matters. A nominal approval is weak evidence of informed authorization if the interface concealed parameters, aggregated several actions, or described a consequential act as routine.

4. Counterfactual Agent Reconstruction

CAR proceeds through ten stages. The sequence is iterative: new evidence may require renewed preservation or a revised hypothesis. The methodology deliberately separates factual collection, causal analysis, and responsibility assessment.

4.1 Stage 1: Stabilize without erasing

The first objective is to prevent further harm while preserving volatile state. Possible actions include revoking a specific agent token, pausing a workflow, disabling external tool execution, snapshotting containers, isolating queues, preserving cloud logs, and capturing current memory stores. A broad shutdown may destroy ephemeral evidence or trigger cleanup jobs. Containment should therefore be scoped and logged.

NIST SP 800-61 Rev. 3 emphasizes recording investigative actions and preserving the integrity and provenance of incident records and collected data [8]. For agentic incidents, the responder's own actions may alter memory, context, or scheduling. Every containment step should identify actor, authority, time, target, expected effect, and observed result.

4.2 Stage 2: Define the consequential act

“The agent went rogue” is not an incident definition. Investigators should identify the external state change or prohibited decision in falsifiable terms: a message was transmitted to a specified recipient; credentials were disclosed; a cloud resource was deleted; a payment was initiated; a record was altered; code was deployed; or a restricted dataset was accessed.

The definition should separate attempted, authorized, executed, and completed acts. It should also specify the relevant decision window. Agent sessions may run for hours, but the causal window may begin when a poisoned document entered memory days earlier.

4.3 Stage 3: Preserve across all evidence planes

Preservation should follow the architecture, not organizational ownership. The team should issue targeted preservation requests to model providers, cloud platforms, identity services, tool vendors, collaboration systems, and upstream data owners where legally authorized. Acquisition should include raw exports, schemas, audit metadata, version information, and integrity values.

The collection manifest should identify every expected source, whether it was obtained, its retention horizon, the responsible custodian, and known gaps. A screenshot of a vendor dashboard is a lead, not a complete acquisition. Where only a proprietary export is available, investigators should preserve the viewer, documentation, and version needed for future interpretation.

4.4 Stage 4: Normalize time and establish event identity

Agentic systems generate overlapping identifiers: conversation, run, trace, span, task, tool call, job, token, transaction, and agent instance. Investigators should create a crosswalk rather than choose one identifier as canonical. Clock sources, time zones, synchronization quality, queue delays, and client-versus-server timestamps must be recorded.

The output is a time-normalized event ledger in which each entry includes source, original timestamp, normalized timestamp, identifier set, actor or component, input reference, action, output reference, and confidence. Raw timestamps must remain recoverable.

4.5 Stage 5: Build the delegation-and-action graph

The graph represents people, agents, services, credentials, data objects, actions, and approvals as nodes connected by typed edges. Recommended edge types include requested-by, authorized-by, instantiated-by, observed, retrieved, wrote-memory, proposed, approved, invoked, executed, generated, modified, delegated-to, blocked-by, and derived-from.

The W3C PROV model provides durable concepts—Entity, Activity, Agent, generation, use, derivation, association, and delegation—that can support interoperable representation [9]. CAR extends the practical vocabulary to distinguish authorization from causal influence and model selection from external execution.

The graph should expose breaks. An action without an authorization edge indicates either missing evidence or a control failure. A rationale without a source edge indicates an unsupported assertion. Two apparently independent agents that consumed the same poisoned memory should not be treated as corroborating one another.

4.6 Stage 6: Reconstruct the observable action trace

The investigator now tells the narrowest account supported by primary records. Each statement is classified as:

5. **observed:** directly recorded by a primary source;
6. **corroborated:** independently supported by two or more sources;
7. **inferred:** the most plausible explanation connecting observations;
8. **claimed:** asserted by a person, vendor, or agent but not independently established;
9. **unknown:** material evidence is absent, inaccessible, or ambiguous.

The trace should include failed attempts, policy interventions, retries, alternate tool selections, and environment responses. “Agent sent email” should be decomposed into generation, tool invocation, authentication, server acceptance, and delivery where evidence permits.

4.7 Stage 7: Form competing causal hypotheses

CAR rejects a single preferred narrative. At minimum, investigators should consider:

10. the user expressly requested the consequential act;
11. the agent reasonably interpreted an ambiguous mandate to include the act;
12. untrusted content hijacked the agent's goal;
13. persistent memory or retrieval was poisoned;
14. an upstream agent supplied a harmful instruction or false fact;
15. the model or orchestration system generated an erroneous plan without adversarial input;
16. a tool schema, parser, or service misinterpreted an otherwise valid request;
17. excessive authorization transformed a minor error into a consequential act;
18. a human approval or monitoring control failed; or
19. records were altered, selectively retained, or misinterpreted after the event.

Each hypothesis should list supporting evidence, contradictory evidence, expected artifacts, missing tests, and alternative explanations.

4.8 Stage 8: Conduct controlled replay

Replay occurs only in an isolated, authorized environment with synthetic or appropriately protected data. The goal is not theatrical reproduction. It is to test whether the preserved configuration can produce the observed pathway and to identify sensitivity.

The replay package should pin or record model version, system instructions, orchestration build, tool schemas, memory snapshot, retrieved content, permissions, sampling settings, and simulated environmental responses. When the historical model is unavailable, the substitution must be explicit and conclusions narrowed.

Investigators should run sufficient repetitions to characterize variation. Results should report the proportion of trials producing the consequential action, important intermediate divergences, and confidence intervals where meaningful. Single-run screenshots are inadequate.

4.9 Stage 9: Perform counterfactual perturbation

Counterfactual testing changes one material factor while holding others as stable as practicable. Examples include:

20. remove the suspected injection while preserving benign document content;
21. retain the injection but mark the source untrusted or isolate it from instructions;
22. replace poisoned memory with the last known clean snapshot;

23. reduce tool permissions to the minimum needed for the stated task;
24. require an informed human approval before the consequential call;
25. substitute an earlier model or policy version;
26. correct an ambiguous tool description or parameter default;
27. remove an upstream agent message;
28. disable retries or parallel execution; or
29. provide the missing contextual fact that should have prevented the act.

The investigator asks: does the outcome disappear, become less frequent, change form, or move to another pathway? A factor is not declared "the cause" merely because its removal prevents the event. The result supports a causal contribution claim bounded by the replay's fidelity.

4.10 Stage 10: Assess responsibility and report uncertainty

Technical causation and normative responsibility are separate. CAR reports which components and decisions materially contributed, which controls failed, and what evidence supports those findings. Legal, disciplinary, or policy responsibility depends on governing law, duty, foreseeability, authority, knowledge, and institutional role.

The final report should lead with the consequential act, confidence level, decisive evidence, material gaps, tested alternatives, reproduction statistics, and minimal preventive changes. It should separately identify user conduct, developer or deployer design, identity and permission governance, vendor behavior, human oversight, and adversarial intervention.

5. The Counterfactual Test Matrix

The following matrix operationalizes Stage 9. It is illustrative rather than exhaustive.

Variable under test	Historical evidence needed	Counterfactual change	Interpretive value
External instruction	exact retrieved content, context placement, retrieval metadata	remove or neutralize suspected injection	tests dependence on untrusted instruction
Memory state	snapshot, write history, retrieval result	restore clean memory	tests persistence or memory poisoning
Authorization	token scopes, policy decision, tool restrictions	apply least privilege	tests whether privilege amplified error
Model/configuration	model snapshot, prompts, sampling, safety settings	substitute recorded alternative	tests version or policy sensitivity
Tool semantics	schema, defaults, parser, service response	clarify parameters or require explicit values	tests interface-induced action

Variable under test	Historical evidence needed	Counterfactual change	Interpretive value
Human control	approval screen, alert, timing, response	insert informed approval or interrupt	tests preventability through oversight
Multi-agent message	sender identity, content, trust, routing	remove or independently verify message	tests delegated misinformation or hijack
Orchestration	retries, concurrency, routing, error handling	disable retry or serialize execution	tests cascade and race conditions

The matrix prevents replay from becoming an unstructured demonstration. Each test has a historical anchor, a specific intervention, an expected discriminating result, and a limit on interpretation.

6. Evidence Confidence and Causal Language

6.1 Four confidence tiers

CAR proposes four confidence tiers for material findings:

30. **Established:** supported by intact primary records whose provenance and interpretation are independently corroborated.
31. **Strongly supported:** supported by consistent records with no material contradiction, but with a limited provenance or retention gap.
32. **Plausible:** consistent with available evidence and replay, but dependent on one or more unverified assumptions.
33. **Indeterminate:** competing explanations remain materially viable or decisive evidence is unavailable.

Confidence applies to a proposition, not an entire report. An investigator may establish that a tool call occurred while remaining unable to determine whether an external injection or endogenous model error selected it.

6.2 Causation vocabulary

Reports should distinguish:

34. **necessary in tested conditions:** removing the factor prevented the outcome in the specified replay set;
35. **risk-increasing:** the factor increased outcome frequency or severity;
36. **enabling:** the factor made the act technically possible but did not select it;
37. **triggering:** the factor immediately preceded and redirected the action pathway;
38. **amplifying:** the factor expanded scale, speed, persistence, or impact;

39. **correlated:** the factor varied with the outcome without a demonstrated causal mechanism; and
40. **unresolved:** evidence cannot distinguish among material explanations.

This language is intentionally narrower than “the AI decided” or “the user caused.” It gives decision-makers a defensible account of what the investigation actually shows.

7. Investigating Prompt Injection and Agent Hijacking

Prompt injection should be investigated as a cross-boundary control failure, not merely as suspicious text. The key questions are where the instruction originated, how it entered context, what authority it acquired, whether it persisted, and which action it influenced.

NIST's evaluation work and AgentDojo show that agent hijacking remains difficult even for capable models [5,6]. The OWASP Top 10 for Agentic Applications 2026 identifies risks including goal hijacking, tool misuse, identity and privilege abuse, memory poisoning, insecure inter-agent communication, and cascading failures [10]. MITRE ATLAS now includes an Agentic AI platform, and its 2026 OpenClaw investigation described prompt smuggling through public-facing content, configuration modification, host escape, and credential harvesting as agent-specific investigative concerns [11,12].

An injection investigation should preserve the original bytes and rendered form, including hidden text, markup, metadata, OCR layers, alternate content, redirects, and scripts. Investigators should reconstruct transformations: sanitization, parsing, summarization, chunking, embedding, retrieval, and placement. A benign-looking model context may be derived from a malicious original, while a malicious-looking stored document may never have reached the model.

Testing should include semantic neutralization rather than deletion alone. Removing an entire document can change the task enough to create a false result. A stronger test preserves factual content while removing adversarial instructions, then compares action traces across repeated trials.

8. Multi-Agent Causation

Multi-agent systems introduce delegation loops, shared memory, role ambiguity, and false corroboration. A planner may delegate to a researcher, which delegates to a browser agent, which writes a summary consumed by an executor. An evaluator agent may approve the result using the same poisoned source, creating the appearance of independent review.

Investigators should model each agent instance as a distinct principal even when instances share a base model. The graph should record role instructions, identity, permissions, accessible memory, messages, and tool boundaries. A stable product name is insufficient because several transient instances may act under it.

Particular attention should be given to:

41. message authenticity and whether sender identity was cryptographically or merely textually asserted;
42. whether downstream agents treated upstream outputs as instruction, evidence, or both;
43. shared-memory write permissions and contamination paths;
44. task decomposition that removed safety-relevant context;
45. recursive delegation that escaped original limits;
46. consensus procedures that reused correlated evidence; and
47. termination, timeout, retry, and conflict-resolution rules.

A multi-agent conclusion should not be described as independently verified unless the agents used independent sources and failure modes. Model diversity alone does not create evidentiary independence.

9. Anti-Forensics and Investigative Failure Modes

Agentic systems create new opportunities for anti-forensics as well as ordinary record loss.

9.1 Selective observability

A vendor may log successful tool calls but omit failed plans, denied requests, retrieved context, safety interventions, or model-routing changes. Such logs can be accurate yet systematically incomplete. Investigators should compare advertised observability with actual telemetry and identify categories that could not be recorded.

9.2 Mutable memory and self-cleanup

Agents may summarize, overwrite, or delete working state to control cost. Cleanup can occur automatically at task completion or incident containment. Memory timestamps and version histories should be preserved, and investigators should determine whether an agent had tools capable of altering its own records or monitoring configuration.

9.3 Rationale laundering

An agent or operator may generate a clean retrospective narrative from incomplete logs. A later summary should be labeled derivative and traced to its source set. It should never replace raw records.

9.4 Trace spoofing and identity confusion

Untrusted content may insert text that resembles a system message, tool output, agent name, or approval. Logs may also accept client-supplied trace identifiers. Investigators should determine which fields were generated by trusted infrastructure and which were supplied by the agent or user.

9.5 Replay overclaim

The most subtle failure is treating successful replay as historical proof. Replay demonstrates system capability under specified conditions. It does not reveal subjective intent, guarantee identical hidden computation, or prove that an unavailable input was present. CAR requires explicit separation between historical findings and experimental findings.

10. Legal and Evidentiary Considerations

10.1 Authentication

In U.S. federal proceedings, the Federal Rules of Evidence provide the general framework for authentication, expert testimony, hearsay, and self-authentication of certain electronic evidence [13]. Rules 901 and 902(13)-(14) may be relevant to electronic records and copied data, but agentic evidence complicates foundation. A witness may need to explain not only how a log was stored, but what component generated each field, whether values were client supplied, how identity was resolved, and what transformations occurred.

The federal evidence rulemaking process is actively examining AI-generated and AI-processed evidence, including machine-generated opinions and deepfakes [14]. Investigators should not assume that a technically valid log is automatically admissible or that an AI output is a business record, expert opinion, or non-hearsay machine result. Counsel must determine the theory under governing law.

10.2 Best evidence and completeness

A human-readable transcript may be only a rendering of structured events. The native export, schema, linked tool records, and configuration may be needed to evaluate completeness. Where a provider offers only a PDF transcript, investigators should document requested but unavailable native fields.

10.3 Expert scope

Fact investigators can establish collection, identity, timelines, permissions, and observed actions. Specialized expertise may be required to interpret model behavior, statistical replay, cryptographic attestations, distributed tracing, or tool implementation. Experts should state validation limits and avoid anthropomorphic claims about belief or intent.

10.4 Discovery, privilege, and confidentiality

Agent logs may contain privileged communications, trade secrets, personal data, credentials, or unrelated third-party content. Collection should be proportionate and legally authorized. Teams should use filtering, access control, protective orders, and minimization without destroying the ability to test context. The investigative need for a full model context does not create unlimited authority to retain sensitive material.

11. Provenance, Integrity, and Interoperability

No single provenance standard presently captures the entire agentic incident. Existing building blocks can nevertheless improve readiness.

The C2PA 2.2 specification provides signed manifests and tamper-evident provenance for digital content, while cautioning that valid provenance is a trust signal rather than a value judgment about truth [15]. W3C PROV represents entities, activities, agents, derivation, association, and delegation [9]. The IETF Supply Chain Integrity, Transparency, and Trust work develops interoperable transparency building blocks for digital supply chains [16]. Software attestations, signed builds, and immutable audit stores can establish model-wrapper and tool lineage.

CAR recommends an **Agent Action Envelope** for each consequential tool call. The envelope should include:

48. globally unique action and trace identifiers;
49. agent and workload identity;
50. initiating mandate reference;
51. model and orchestration version references;
52. input and memory bundle hashes;
53. proposed tool, parameters, and declared purpose;
54. authorization decision and credential scope;
55. human approval or policy-gate result;
56. execution service and environment response;
57. resulting artifact or state reference;
58. time source and sequence information; and
59. signatures or integrity protections appropriate to risk.

The envelope should not contain hidden chain-of-thought. It should preserve operational facts needed for accountability while minimizing unnecessary sensitive content.

12. Agentic Forensic Readiness

Investigation quality is largely determined before an incident. Organizations deploying consequential agents should adopt an Agentic Forensic Readiness Standard with the following minimum controls.

12.1 Identifiable agents and bounded credentials

Each agent instance should use an identifiable workload principal rather than silently inheriting a human session. Credentials should be scoped by task, tool, resource, duration, and environment. Delegation should remain traceable across agents and services.

12.2 Complete event design

Logging requirements should be defined from investigative questions. NIST log-management guidance treats logs as records needed to identify and investigate incidents, while SP 800-61 Rev. 3 emphasizes integrity and provenance [8,17]. Agent systems should record mandate changes, context assembly, memory reads and writes, tool proposals, policy decisions, approvals, executions, responses, retries, delegation, and shutdown events.

12.3 Versioned context and policy

System instructions, tool schemas, model routes, safety policies, retrieval settings, and memory compaction rules should be versioned. A deployment label such as “production” is not sufficient for later reconstruction.

12.4 Tamper-evident, segregated storage

The agent should not be able to rewrite the authoritative audit record with the same privilege used for task execution. High-risk systems should stream signed or append-only events to a segregated repository, monitor logging failure, and retain the schemas and software needed to interpret records.

12.5 Replay packages

Deployers should maintain a reproducible test harness, sanitized environment fixtures, and version manifests. The ability to rerun a system should be tested before an incident, not improvised after the relevant model and dependencies disappear.

12.6 Informed approval and interruptibility

Approval interfaces should show the actual action, target, scope, irreversible consequences, source of sensitive parameters, and alternatives. Emergency interruption should revoke or suspend authority without deleting evidence. OpenAI's governance paper and NIST's identity work both identify control, approval, and attribution as central agentic concerns [4,7].

13. A Hypothetical Application

Consider a procurement agent authorized to gather quotes, draft recommendations, and prepare purchase orders below a threshold. It reads a supplier webpage containing hidden instructions to ignore competing vendors and transmit an internal budget spreadsheet to an external address. The agent retrieves the page, writes a summary into shared memory, asks a separate finance agent for the spreadsheet, and invokes an email tool using a procurement employee's delegated token. A monitoring agent approves the message because it evaluates only the agent-generated summary.

A conventional investigation may find the sent email and employee token, then attribute the disclosure to the employee's account. CAR produces a different analysis.

First, historical reconstruction shows that the employee requested quote comparison, not disclosure. The supplier page entered as untrusted retrieval data. The browser agent wrote a summary that omitted the hidden instruction's origin. The finance agent released the spreadsheet because inter-agent requests were implicitly trusted. The monitoring agent evaluated a derivative summary rather than the proposed attachment and destination. The email service accepted the message under a broadly scoped delegated token.

Second, counterfactual tests show that neutralizing the hidden instruction prevents disclosure in 48 of 50 controlled runs; retaining the instruction but marking tool output as untrusted prevents it in 50 of 50; narrowing the token blocks transmission in every run; and requiring an approval screen showing the external recipient and attachment also blocks it. Substituting a different model reduces but does not eliminate the behavior.

The defensible conclusion is not simply that “the model was hacked.” The external instruction triggered the harmful pathway under tested conditions; shared-memory trust propagated it; excessive permissions enabled execution; and two oversight controls failed to surface material facts. The employee account authorized the service transaction technically, but the employee did not request or knowingly approve the disclosure in the hypothetical evidence. Responsibility assessment would then consider organizational design, vendor controls, employee conduct, and attacker action under governing law.

14. Validation and Research Design

CAR is a proposed methodology and requires empirical validation. A useful evaluation program should include synthetic incidents, seeded evidence gaps, multiple agent frameworks, different models, and blinded investigators.

14.1 Agentic Forensic Challenge Corpus

The field should develop a privacy-preserving corpus of fully instrumented scenarios covering:

60. indirect prompt injection through web, email, image, and document content;
61. poisoned long-term memory;
62. malicious or mistaken inter-agent delegation;
63. ambiguous tool schemas and unsafe defaults;
64. excessive permission and identity confusion;
65. orchestration retries, race conditions, and cascading failure;
66. model or policy version drift;
67. forged approvals and trace spoofing;
68. selective vendor logging; and
69. benign anomalies that should not be misclassified as attack.

Each case should include a hidden ground truth, raw evidence, realistic missing records, and a controlled replay environment. Benchmarks such as AgentDojo and Agent Security Bench demonstrate the value of dynamic, tool-using environments for security evaluation [6,18], but a forensic benchmark must additionally measure reconstruction accuracy, source traceability, uncertainty calibration, and resistance to hindsight bias.

14.2 Evaluation measures

Proposed measures include:

70. accuracy of consequential-act reconstruction;
71. correct separation of requested, authorized, selected, and executed actions;
72. detection of injected or poisoned influence;
73. false attribution rate;
74. completeness of evidence-plane acquisition;
75. calibration between stated confidence and ground truth;
76. reproducibility of replay results;
77. sensitivity to missing or misleading logs;
78. time and cost per investigation; and
79. agreement among independent teams.

14.3 Research questions

Priority questions are:

80. What minimum telemetry permits reliable reconstruction without collecting hidden reasoning or excessive personal data?

81. How many replay trials are needed to characterize model-dependent variance for different claim types?
82. Which counterfactual interventions best distinguish prompt injection from endogenous planning error?
83. How should replay evidence be communicated to courts, regulators, and nontechnical decision-makers?
84. Can provenance envelopes remain interoperable across model providers, agent frameworks, and tool protocols?
85. How often do nominally independent agents share correlated sources or failure modes?
86. Which approval-interface properties produce genuinely informed human authorization?
87. How should investigators validate hosted systems when providers cannot preserve historical model snapshots?

15. Governance Recommendations

15.1 For investigators

88. Treat the visible transcript as an index, not the complete record.
89. Acquire identity, authorization, context assembly, memory, and tool-service evidence early.
90. Separate historical findings from experimental replay findings.
91. Use competing hypotheses and report adverse as well as supportive results.
92. Avoid attributing belief, intent, or knowledge to a model without a legally and scientifically defensible basis.
93. Record every investigative intervention that could change agent state.
94. Engage specialists when claims depend on model behavior, distributed systems, statistics, or cryptography.

15.2 For organizations deploying agents

95. Assign unique agent identities and task-bounded authority.
96. Preserve context assembly and tool-call provenance for consequential actions.
97. Keep authoritative logs outside the agent's modification boundary.
98. Version prompts, policies, models, tools, memory schemas, and orchestration.
99. Test prompt-injection containment and cross-agent trust boundaries.
100. Maintain isolated replay capability and incident-ready vendor contracts.
101. Design approval screens around actual consequences, not agent-generated summaries.
102. Establish retention schedules that reflect legal, safety, privacy, and forensic needs.

15.3 For standards bodies and vendors

103. Define interoperable action envelopes and delegation semantics.
104. Distinguish trusted infrastructure fields from agent- or client-supplied fields.
105. Support export of native traces, schemas, policy decisions, and version manifests.
106. Publish known observability gaps and retention limitations.
107. Provide a method to identify hosted model snapshots or document material changes.
108. Develop standard test fixtures for counterfactual replay.
109. Ensure provenance standards signal integrity without overstating truth or responsibility.

15.4 For courts and regulators

110. Require foundations that explain how agentic records were generated and what they omit.
111. Distinguish tool execution records from model-generated narratives.
112. Permit proportionate access to configuration and replay evidence when reliability is contested.
113. Protect confidential and personal data through scoped process rather than denying necessary technical examination.
114. Encourage neutral terminology and quantified uncertainty in expert reporting.

16. Limitations

CAR is conceptual. It has not yet been validated through controlled studies or adversarial field exercises. The method may be expensive for small investigations and may depend on records held by uncooperative vendors or foreign services. Hosted model snapshots, internal safety interventions, and complete context assembly may be unavailable. Replay can be limited by nondeterminism, changed dependencies, legal restrictions, and the impossibility of reproducing the exact external environment.

The methodology does not solve philosophical questions about machine agency or legal questions about personhood, intent, product liability, employment responsibility, or criminal culpability. It deliberately focuses on operational causation and evidence. Jurisdictions differ in search, seizure, discovery, privacy, employment monitoring, privilege, expert evidence, and record-retention rules.

Finally, more logging is not automatically better. Comprehensive prompts and memory may contain sensitive personal, proprietary, or privileged information. Forensic readiness must be risk-based, purpose-limited, access-controlled, and subject to deletion rules. Accountability architecture should not become a system of indiscriminate surveillance.

17. Conclusion

Autonomous AI agents turn software incidents into delegation incidents. A consequential act may emerge from a user's goal, a model's plan, an untrusted document, a poisoned memory, a permissive credential, an ambiguous tool, an upstream agent, a hidden retry, and a superficial human approval—all within one trace. No single artifact can explain that system.

Counterfactual Agent Reconstruction offers a disciplined response. It preserves evidence across ten planes, builds a typed delegation-and-action graph, reconstructs the observable historical pathway, tests competing hypotheses through controlled replay, and reports causal contribution without converting probability into certainty. Its central question is practical: what minimal, evidence-supported change would have altered the outcome?

The method also changes forensic readiness. Agent identity, context provenance, memory lineage, action envelopes, externalized audit logs, versioned policy, and replay packages must become design requirements for consequential agents. A system that can act but cannot later explain—through trustworthy operational records—what mandate, authority, evidence, and control produced the act is not merely difficult to investigate. It is insufficiently governable.

The cutting edge of investigation is therefore not an attempt to read a model's mind. It is the construction of reliable evidence around delegation, observation, state, action, and intervention. When software acts, accountability depends on making that causal system visible.

References

115. National Institute of Standards and Technology. (2026). *Announcing the AI Agent Standards Initiative for Interoperable and Secure Innovation*. https://www.nist.gov/news-events/news/2026/02/announcing-ai-agent-standards-initiative-interoperable-and-secure

116. National Institute of Standards and Technology. (2026). *AI Agent Standards Initiative*. https://www.nist.gov/artificial-intelligence/ai-agent-standards-initiative

117. Riggs, J., Hamin, M., Perry, N., Edelman, B., & Cihon, P. (2026). *Summary Analysis of Responses to the Request for Information Regarding Security Considerations for AI Agents* (NIST AI 800-5). https://www.nist.gov/publications/summary-analysis-responses-request-information-regarding-security-considerations-ai

118. Shavit, Y., Agarwal, S., Brundage, M., et al. (2023). *Practices for Governing Agentic AI Systems*. https://openai.com/index/practices-for-governing-agentic-ai-systems/

119. Center for AI Standards and Innovation. (2025). *Strengthening AI Agent Hijacking Evaluations*. https://www.nist.gov/news-events/news/2025/01/technical-blog-strengthening-ai-agent-hijacking-evaluations

120. Debenedetti, E., Zhang, J., Balunovic, M., Beurer-Kellner, L., Fischer, M., & Tramèr, F. (2024). *AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents*. NeurIPS Datasets and Benchmarks Track. https://papers.neurips.cc/paper_files/paper/2024/file/97091a5177d8dc64b1da8bf3e1f6fb54-Paper-Datasets_and_Benchmarks_Track.pdf

121. National Cybersecurity Center of Excellence. (2026). *Accelerating the Adoption of Software and AI Agent Identity and Authorization: Concept Paper*. https://www.nccoe.nist.gov/sites/default/files/2026-02/accelerating-the-adoption-of-software-and-ai-agent-identity-and-authorization-concept-paper.pdf

122. Nelson, A., Rekhi, S., Souppaya, M., & Scarfone, K. (2025). *Incident Response Recommendations and Considerations for Cybersecurity Risk Management: A CSF 2.0 Community Profile* (NIST SP 800-61 Rev. 3). https://doi.org/10.6028/NIST.SP.800-61r3

123. World Wide Web Consortium. (2013). *PROV-O: The PROV Ontology*. https://www.w3.org/TR/prov-o/

124. OWASP GenAI Security Project. (2025). *OWASP Top 10 for Agentic Applications for 2026*. https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/

125. MITRE. (2026). *MITRE ATLAS*. https://atlas.mitre.org/

126. MITRE. (2026). *MITRE ATLAS OpenClaw Investigation*. https://www.mitre.org/sites/default/files/2026-02/PR-26-00176-1-MITRE-ATLAS-OpenClaw-Investigation.pdf

127. Administrative Office of the U.S. Courts. *Federal Rules of Evidence*. https://www.uscourts.gov/forms-rules/current-rules-practice-procedure/federal-rules-evidence

128. Administrative Office of the U.S. Courts. (2026). *Objection! How the Federal Rules of Evidence Promote Fair Trials*. https://www.uscourts.gov/data-news/judiciary-news/2026/05/05/objection-how-federal-rules-evidence-promote-fair-trials

129. Coalition for Content Provenance and Authenticity. (2025). *C2PA Technical Specification 2.2: Content Credentials*. https://spec.c2pa.org/specifications/specifications/2.2/specs/C2PA_Specification.html

130. IETF Supply Chain Integrity, Transparency, and Trust Working Group. (2025). *An Architecture for Trustworthy and Transparent Digital Supply Chains*. https://datatracker.ietf.org/doc/draft-ietf-scitt-architecture/

131. Kent, K., & Souppaya, M. (2006). *Guide to Computer Security Log Management* (NIST SP 800-92). https://doi.org/10.6028/NIST.SP.800-92

132. Zhang, Z., et al. (2025). *Agent Security Bench: Formalizing and Benchmarking Attacks and Defenses in LLM-based Agents*. ICLR 2025. https://luckfort.github.io/ASBench/

133. Autio, C., Schwartz, R., Dunietz, J., et al. (2024). *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile* (NIST AI 600-1). https://doi.org/10.6028/NIST.AI.600-1

134. National Institute of Standards and Technology. (2025). *Lessons Learned from the Consortium: Tool Use in Agent Systems*. https://www.nist.gov/news-events/news/2025/08/lessons-learned-consortium-tool-use-agent-systems

135. National Institute of Standards and Technology. (2026). *CAISI Issues Request for Information About Securing AI Agent Systems*. https://www.nist.gov/news-events/news/2026/01/caisi-issues-request-information-about-securing-ai-agent-systems

136. Scientific Working Group on Digital Evidence. (2025). *Best Practices for Computer Forensic Examinations* (SWGDE 18-F-001-2.0). https://www.swgde.org/documents/published-complete-listing/18-f-001-swgde-best-practices-for-computer-forensic-examination/

137. National Institute of Standards and Technology. (2022; updated 2026). *Digital Evidence Preservation: Considerations for Evidence Handlers*. https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt/digital-evidence

138. Casey, E. (2011). *Digital Evidence and Computer Crime* (3rd ed.). Academic Press.

139. Carrier, B. (2005). *File System Forensic Analysis*. Addison-Wesley.

140. Ruan, Y., Dong, H., Wang, A., et al. (2024). *Identifying the Risks of LM Agents with an LM-Emulated Sandbox*. ICLR 2024. https://openreview.net/forum?id=GEcwtMk1uA

141. Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). "Not What You've Signed Up For": Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. *ACM Workshop on Artificial Intelligence and Security*. https://doi.org/10.1145/3605764.3623985

142. Scarfone, K., & Souppaya, M. (2023). *Cybersecurity Log Management Planning Guide: Initial Public Draft* (NIST SP 800-92 Rev. 1). https://doi.org/10.6028/NIST.SP.800-92r1.ipd

143. National Institute of Standards and Technology. (2026). *New Concept Paper on Identity and Authority of Software Agents*. https://www.nist.gov/news-events/news/2026/02/new-concept-paper-identity-and-authority-software-agents

144. National Institute of Standards and Technology. (2026). *Concept Note: AI RMF Profile on Trustworthy AI in Critical Infrastructure*. https://www.nist.gov/programs-projects/concept-note-ai-rmf-profile-trustworthy-ai-critical-infrastructure

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 conflict of interest.

Data availability: No empirical data were generated for this conceptual article. Any future benchmark, replay package, schemas, and synthetic evaluation materials should be released where ethically, legally, and operationally permissible.

Ethics statement: No human participants or personal data were used in preparing this conceptual article. Future studies involving investigators, developers, incident records, or human participants would require appropriate institutional ethics review and data-protection controls.

Use of generative AI: The author used generative artificial intelligence as an assistive writing tool to help address difficulties associated with dyslexia, including support with spelling, grammar, sentence structure, organization, and clarity. The author developed the article's arguments, reviewed and revised the generated text, checked the cited sources, and takes full responsibility for the accuracy, originality, analysis, and final content. Generative AI was not used to generate or analyze empirical data.

Scope statement: This article presents a conceptual investigation methodology and general academic analysis. It is not legal advice, a validated forensic standard, or a substitute for jurisdiction-specific authority, qualified expert judgment, or organizational incident-response policy.