WHITE PAPER · ACCESS GOVERNANCE · SOC 2 CC8.1

The Causal Chain Problem

> AI Agents, Non-Repudiation, and the SOC 2 CC8.1 Gap — A Framework for Governing Autonomous AI Agents in Compliance Environments

AUTHOR
Steve Weltman, CISSP

ORGANIZATION
Aletheia Security Consulting

PUBLISHED
June 2026

ABSTRACT

Every access control framework in existence — SOC 2 Type II, ISO 27001, HIPAA Security Rule, CMMC 2.0 — was written with a foundational assumption: a human being is the atomic unit of accountability. A human requests access, a human is authorized, a human takes action, and a human's identity appears in the audit log. The causal chain from decision to system change terminates at a person.

Autonomous AI coding agents break that assumption — deliberately. The severing of the causal chain is not a defect to be patched. It is the core value proposition of agentic AI. This paper argues that the current compliance posture of most organizations using AI coding agents represents a material non-repudiation failure that SOC 2 auditors will increasingly be required to address. It defines the specific gap in CC8.1, identifies why common workarounds are insufficient at scale, and proposes a Causal Chain Framework for Agentic Access Governance that satisfies non-repudiation requirements without eliminating agent autonomy.

1. The Foundational Assumption That AI Agents Break

1.1 How SOC 2 CC8.1 Assumes Human Causality

SOC 2 CC8.1, under the Change Management category of the Common Criteria, requires that an organization demonstrate the following for changes to its production environment:

- Changes are authorized by individuals with appropriate authority
- Changes are tested and reviewed prior to implementation
- Changes are documented with sufficient evidence to trace each change to an authorization event
- Deviations are identified and addressed

The operative phrase is **"authorized by individuals."** The AICPA's intent, when this criterion was written, was that a named, accountable human being could be identified as the authorizing party for any given production change. That person could be held accountable. Their authorization could be verified. The causal chain — human decision, human authorization, human action — was assumed to be intact.

This assumption is load-bearing throughout the entire SOC 2 control framework. CC6.1 (logical access controls), CC6.2 (user registration), CC6.3 (access removal), CC7.2 (incident identification and response) — all of them trace accountability to human actors. The audit trail is, by design, a trail of human decisions and human actions.

This paper focuses on CC8.1 because change management is where the gap is most acute and most auditable. But the underlying problem — human causality assumed throughout a framework being applied to non-human actors — is systemic.

The design assumption that produces this gap is precise: every elevated privilege and change management control in these frameworks was built on a **1:1 model**. One person, one ticket, one approval, one change. The manual approval chain is the control — and it works at human scale, when the unit of work is a deliberate, bounded action by an accountable individual.

Agentic AI does not operate at human scale. An agent making thousands of changes in a day under the same 1:1 approval assumption would require thousands of discrete approvals. No organization requires this. No framework has acknowledged that it cannot. The frameworks were not designed for a non-human actor with autonomous decision-making — that is not an indictment of the frameworks. It is simply the reality that compliance programs are now required to navigate.

1.2 What Agentic AI Actually Does

An AI coding agent, operating through a tool protocol such as MCP (Model Context Protocol), takes the following sequence of actions in a typical production engagement:

1. A human engineer issues a goal: "Update the authentication library to address the CVE flagged in last week's scan."
2. The agent reads the codebase, identifies affected dependencies, and plans a remediation approach.
3. The agent modifies files, updates configuration, runs tests, and — if configured for autonomous deployment — pushes changes to production.
4. Step 3 may involve dozens to hundreds of individual actions: file reads, file writes, shell commands, API calls, database queries — all at machine speed.

The human participated in step 1. Steps 2 through 4 happened without human involvement in each individual action. In a well-instrumented environment, the agent's actions appear in application logs, git history, and deployment records — but attributed to the engineer who invoked the agent, because the agent operated using that engineer's credentials.

The audit trail, as seen by a SOC 2 auditor, shows: "Jane authorized and implemented 47 production changes on Tuesday afternoon." Jane may have been in a meeting during that entire period.

1.3 The Non-Repudiation Failure

Non-repudiation is the security property that ensures a party cannot deny having performed an action. It requires that the record of an action be bound to the identity of the actor in a way that is both accurate and verifiable. When an AI agent operates using human credentials, non-repudiation fails in two directions simultaneously:

- **Forward non-repudiation failure:** Agent actions are attributed to a human who did not take them. The human cannot truthfully confirm those actions as their own deliberate choices.
- **Reverse non-repudiation failure:** Agent actions cannot be traced back to any accountability primitive. If something goes wrong, there is no accountable party. "The agent did it" is not an audit finding — it is the absence of one.

This is not a theoretical concern. It is the current state of most organizations using AI coding agents in production today. The agents are running. The credentials are shared. The audit logs are ambiguous.

This is not a logging deficiency. The system is working as designed — the agent operated using human credentials, and the log correctly records what those credentials did. The false attestation problem is structural. The log is not inaccurate. **The log is a lie with a clean format.**

1.4 Three Compounding Failures

The non-repudiation gap described in 1.3 does not manifest as a single failure. It manifests as three distinct and compounding failures that reinforce each other.

Identity Laundering. Agent actions are recorded as human actions. The SOC 2 report will attest that change management controls operated effectively — controls that presuppose human-initiated, human-approved changes. If agents are making changes under human identities with no per-action approval chain, the attestation is accurate about the controls and false about who operated them. This is not a gap that better documentation closes. It is a false evidentiary record produced by the control architecture itself.

Approval Scope Explosion. The 1:1 approval model does not scale to agent output volumes. Pre-authorization functions as a compensating control only when scope is genuinely narrow and agent outputs are predictable from the authorization. Agentic AI by definition operates in spaces where specific outputs are not fully predictable from inputs — that is the value proposition. The authorization covers a direction, not a changeset. At machine scale, the compensating control is structurally insufficient.

The Branch Approval Fiction. Agents appending to branches that "must be approved by a person" appears to restore the causal chain. At scale, it does not. When an agent produces hundreds of commits in a session, the human reviewer cannot meaningfully evaluate what they are approving. The review gate exists in the system. It does not function as a control in practice. An approval that cannot be meaningfully exercised is not an approval. It is a liability transfer.

2. Why Current Workarounds Are Insufficient

2.1 The Service Account Treatment

The most common current response — from both organizations and auditors — is to treat AI agents as service accounts, a known pattern with established controls. This treatment fails for one critical reason: **scope indeterminacy.**

A traditional service account performs a defined, bounded function. An AI coding agent's scope is indeterminate by design. Restricting it to a narrow service-account-style permission set either renders it unable to complete complex tasks, or requires granting such broad permissions that the "service account" label provides no meaningful scope restriction.

More critically, the service account treatment does not address the causal chain problem. Even a properly scoped service account assigned to an AI agent still obscures which human decision authorized which specific change in a way that is non-repudiable.

2.2 Human-in-the-Loop Mandates

A second response is to require human approval at every step — no agent action without explicit human sign-off. This restores the causal chain by reinstating the human at each link. It also eliminates the primary value of agentic AI.

At any meaningful scale of autonomous operation, human-in-the-loop mandates become rubber-stamping. When an agent executes 200 production changes per day and each requires human approval, the approval becomes a checkbox. Humans approve without reviewing.

> **CRITICAL DISTINCTION**
>
> This is the Branch Approval Fiction in practice: the approval exists on paper, the auditor can sample it, and it demonstrates nothing. Checkbox approval of actions the human did not evaluate is arguably worse than transparent documentation that an agent acted autonomously — because it creates a false record of human review. The goal of a governance framework is not to maximize human involvement. It is to ensure that human authorization is **meaningful where it exists** and **accurately documented where it does not.**

2.3 Policy Controls Alone

A written policy stating that AI agents are not permitted to make production changes without human review is a necessary component of any governance framework — but it does not satisfy CC8.1's evidence requirements for the same reason that a policy requiring seatbelts does not demonstrate that seatbelts were worn. The auditor needs evidence that the control operated effectively, not just that a policy existed.

For AI agent governance specifically, policy controls face a structural challenge: the agents themselves generate the evidence. If the agent's actions are not instrumented at the access layer — before the action occurs — the evidence artifacts are produced after the fact by the same system that took the action. That is not auditable non-repudiation; it is self-attestation.

3. The Causal Chain Framework

The following framework addresses the non-repudiation gap without mandating human review of every agent action. Its core principle: **preserve human accountability at the authorization level while providing accurate agent attribution at the action level.**

PRINCIPLE 1

Categorical Pre-Authorization

Rather than requiring human approval for each individual agent action, the framework requires human authorization of a category of change. A human authorizer approves a scope — describing the purpose, target systems, and time window — not each atomic operation. The human can evaluate the category; they are releasing the agent to act within a legible boundary.

Categorical authorization events must be: recorded at the time of authorization (not reconstructed after), bound to a specific human identity with verified authentication, scoped to a defined change class and target system set, and cryptographically signed so they cannot be altered retroactively.

PRINCIPLE 2

Agent Identity Separation

The agent must operate with its own identity, not a delegated copy of the human's identity. When an agent uses human credentials, every action is attributed to the human — this is the root of the non-repudiation failure.

Short-lived credential mechanisms (JIT certificates, signed tokens, temporary role bindings) already exist in most modern PAM and identity platforms. What is required is that these mechanisms be applied specifically to AI agent identities, with the agent's credential bound to the categorical authorization event that enabled it. The credential's metadata should encode the agent's unique identifier, the authorization event ID and timestamp, the authorizing human's identity, permitted scope, and a hard expiry enforced at the issuing authority level.

PRINCIPLE 3

Dual-Record Non-Repudiation

The audit record for any agent action must contain two distinct, cryptographically linked entries:

Event A — Human Authorization: A record of the specific human, authenticated by their identity provider, authorizing a category of agent action. Contains the human's identity, authentication assurance level, approved scope, and timestamp.

Event B — Agent Execution: A record of the specific agent, operating under a JIT credential tied to Event A, performing a specific action. Contains the agent's identity, the credential serial traceable to Event A, the action taken, and timestamp.

Together: this human decided this scope was appropriate; this agent, acting under that authority, took this specific action. The auditor can verify both, and verify the link between them.

PRINCIPLE 4

Pre-Flight Verification at the Access Gate

Before issuing a JIT credential for any agent session, the governance layer should perform automated verification appropriate to the change class:

- **Vulnerability pre-flight:** Check proposed packages against known CVE databases (OSV.dev, NVD, Snyk) at the moment of credential issuance. Sessions introducing known critical CVEs are blocked before they start.
- **Policy pre-flight:** Verify the requested change class is not prohibited under current organizational policy (change freeze windows, release lockouts, incident response blocks).
- **Scope pre-flight:** Confirm the agent's stated target systems are within scope for the authorized change category.

Pre-flight failures are recorded with the same fidelity as successful sessions. What was blocked, and why, is often the most auditor-relevant evidence.

PRINCIPLE 5

Delegation Chains in Multi-Agent Orchestration

Modern AI architectures increasingly involve orchestrators spawning sub-agents. The causal chain must be preserved through delegation: each sub-agent receives a scoped sub-credential issued by the orchestrator's credential, which is in turn traceable to the original human authorization event.

The resulting audit structure is a directed acyclic graph rooted at a single human authorization event, with every agent action at every level of delegation traceable back to that root. The question "which human authorized this production change?" has a single, verifiable, non-repudiable answer regardless of how many layers of agent delegation intervened.

4. Evidence Requirements for Auditors

This section describes what auditors evaluating CC8.1 compliance in environments using AI coding agents should expect to see. These are proposed evidence criteria — we believe CC8.1's non-repudiation requirements, properly applied to agentic AI, require this class of evidence. Authoritative guidance from the AICPA has not yet addressed agentic AI specifically; these criteria are offered as a practitioner framework pending that guidance.

4.1 Minimum Evidence Artifacts

Artifact	Required Content	Verification Method
Human Authorization Record	Human identity; authentication timestamp; MFA confirmation; authorized scope; expiry	Cryptographic signature; traceable to IdP logs
Agent Credential Record	Agent identity; credential serial; issuing authority; scope; expiry; link to authorization record	Cryptographic chain from authorization record
Action Execution Log	Each action; timestamp; agent credential used; target system; change summary	Hash-chained log; each entry references the prior entry
Session Closure Record	Voluntary or forced closure; session duration; summary of changes made	Linked to credential record; completion status
Pre-Flight Verification Record	Checks performed; results (pass/fail); CVEs queried; policy checks; timestamp	Independent log; not modifiable by the evaluated agent

4.2 What Adequate Evidence Demonstrates

Taken together, adequate evidence for CC8.1 in an agentic environment should allow the auditor to:

1. **Name the authorizing human** for any given production change and verify their authentication
2. **Confirm actions were within authorized scope** — the agent did not exceed what the human approved
3. **Verify the causal link** between the human's decision and agent actions cannot have been forged or retroactively altered
4. **Confirm pre-flight checks were performed** before credentials were issued, with blocks recorded
5. **Trace multi-agent delegation** — in orchestrated scenarios, trace each sub-agent's actions to the root authorization

4.3 Red Flags for Auditors

AUDITOR RED FLAGS — AI AGENT GOVERNANCE GAPS

- **Agent activity attributed to human accounts:** Git commits or deployment records show human usernames during periods when the named human could not have been present
- **No distinct agent identity in access logs:** All agent activity appears under shared service accounts or human accounts with no agent-specific identifier
- **Change tickets with no agent attribution:** Tickets reference AI-assisted changes but do not identify which agent, with which scope, under which authorization
- **"Human review" checkboxes at implausible frequency:** Dozens of approvals in minutes; approvals at 2am; approval patterns consistent with automated clicking rather than human review
- **No pre-flight records:** No evidence of vulnerability or policy checks prior to agent sessions, or pre-flight checks performed by the same system that would benefit from bypassing them

5. Regulatory Trajectory

5.1 EU AI Act

The EU AI Act, with high-risk system provisions taking effect 2025–2027, requires organizations deploying AI systems in specified domains to maintain risk management documentation (Article 9), audit logging of AI system actions (Article 12), and human oversight mechanisms that are effective — not merely nominal (Article 14). AI coding agents making autonomous production changes to software systems will, in many contexts, qualify as high-risk systems under Annex III.

Article 14's human oversight requirement is not satisfied by a policy mandate for human review that is not enforced at the technical layer. The Causal Chain Framework's categorical pre-authorization model is an example of a technical implementation that satisfies Article 14's intent.

5.2 NIST AI Risk Management Framework

NIST's AI RMF (2023) addresses AI governance through the GOVERN, MAP, MEASURE, and MANAGE functions. The most directly applicable provisions for agentic access governance are:

NIST AI RMF Function	Provision	Framework Mapping
GOVERN 1.1		

NIST AI RMF Function	Provision	Framework Mapping
	Policies and procedures for AI risk management	Categorical pre-authorization policy with technical enforcement
MAP 1.5	Organizational risk tolerances established	Categorical risk classes define organizational tolerance by change type
MEASURE 2.5	AI system performance monitored for trustworthiness	Pre-flight verification and continuous session monitoring

5.3 The Coming Audit Standard

It is our view that within 24–36 months, SOC 2 auditors will routinely include questions about AI agent access governance in their CC8.1 testing procedures. The questions will not be novel — they will be the same questions auditors already ask about human privileged access, applied to a new class of actor:

- How are AI agents identified and authenticated?
- How is human authorization recorded and linked to agent actions?
- What prevents an agent from acting outside its authorized scope?
- What evidence exists that the causal chain from human decision to system change is intact and non-repudiable?

Organizations that can answer these questions with evidence artifacts will receive clean opinions. Organizations that cannot will receive findings. The companies with the most acute exposure are those currently using AI coding agents under the assumption that service account treatment or policy documentation is sufficient. The gap between that assumption and what auditors will require is the compliance debt accruing right now.

6. Implementation Guidance

6.1 What This Framework Does Not Require

> This framework does not require replacing existing PAM infrastructure, eliminating human-in-the-loop for all operations, or adopting any specific vendor or product. The JIT credential mechanisms required exist in most mature PAM platforms (CyberArk, BeyondTrust, HashiCorp Vault, Teleport, AWS IAM, and others). Implementation is a configuration and integration question, not a forklift replacement.

6.2 Integration Touchpoints

A complete implementation typically integrates with the following systems — none of which need to be replaced:

System Category	Role in the Framework	Examples
Identity Provider / MFA	Authenticates and records the human authorization event	Okta, Azure AD, Duo
JIT PAM / Secrets Management	Issues and enforces scoped, short-lived agent credentials	CyberArk, BeyondTrust, Vault, Teleport, AWS IAM
Change Management / Ticketing	Links authorization events to change records	Jira, ServiceNow, Linear
ChatOps / Approval Channel	Surfaces approval requests to human authorizers in real time	Slack, Teams, PagerDuty
Monitoring / SIEM	Receives audit ledger records for correlation and alerting	Splunk, Elastic, Grafana Loki

6.3 Phased Implementation

Organizations implementing this framework incrementally should prioritize in this order:

1. **Visibility:** Instrument agent activity at the identity layer. Establish what agents are running and what they are doing. This alone may surface the non-repudiation gap to stakeholders who have not yet recognized it.
2. **Separation:** Establish distinct agent identities. Even without JIT credentials, replacing shared human credentials with dedicated agent service identities resolves attribution ambiguity.

3. **Authorization Records:** Implement categorical pre-authorization with recorded human approval events. Connect approval records to agent action records with a persistent link.
4. **Pre-Flight and Enforcement:** Add pre-flight verification at the credential issuance gate and enforce scope boundaries technically, not just by policy.
5. **Cryptographic Chain:** Harden the audit ledger with cryptographic chaining so records cannot be altered after creation.

Conclusion

The compliance community is at an inflection point. AI coding agents are in production at scale. The frameworks that govern production access were written for human actors. The gap between what agents do and what compliance frameworks assume is growing with every deployment.

The Causal Chain Problem is not a gap that will be closed by waiting for updated AICPA guidance, or by telling auditors that AI agents are "just like service accounts." The non-repudiation failure is structural. It requires a structural response.

The framework described in this paper — categorical pre-authorization, agent identity separation, dual-record non-repudiation, pre-flight verification, and delegation chain tracking — addresses the structural failure without eliminating the autonomy that makes AI agents valuable. It produces evidence artifacts that auditors can evaluate against existing criteria and future guidance alike.

The organizations that implement this framework before their next SOC 2 Type II audit will be ahead of the requirement. The organizations that wait for auditors to formalize the question will be answering it under time pressure, with an audit finding on the table.

The causal chain, once broken, leaves no evidence. The work is to restore it before anyone asks.

Appendix A. CC8.1 Evidence Matrix

CC8.1 Sub-Requirement	Traditional Evidence	Agentic Evidence Equivalent
Changes are authorized	Signed change request; approver identity	Categorical pre-authorization record with human identity and MFA timestamp
Changes are documented	Change ticket; description; impact assessment	Authorization record scope definition + agent session summary
	Test results; QA sign-off	

CC8.1 Sub-Requirement	Traditional Evidence	Agentic Evidence Equivalent
Changes are tested prior to implementation		Pre-flight verification record; agent test execution logs within authorized scope
Changes are reviewed	Code review records; reviewer identity	Scope boundary enforcement (agent could not act outside reviewed scope) + post-session review records
Deviations identified and addressed	Exception reports; incident tickets	Pre-flight block records; scope violation alerts; auto-revocation events

Appendix B. Glossary

Agentic AI

> An AI system that takes sequences of actions to accomplish a goal, using tools to interact with external systems, without human involvement in each individual action.

Causal Chain

> The traceable sequence from a human decision to authorize an action, through the action's execution, to the resulting system change. A complete causal chain is required for non-repudiation.

Categorical Pre-Authorization

> Human approval of a class of change (e.g., "dependency updates to the authentication service") rather than each individual action within that class. The authorized category defines the agent's operational scope.

JIT Credential (Just-in-Time)

> A short-lived, scoped access credential issued at the moment of need and expired automatically. JIT credentials eliminate standing privilege — no persistent access exists when not in active use.

Non-Repudiation

> The security property ensuring that a party cannot deny having taken an action, because the record of that action is bound to their identity in a way that is accurate and verifiable.

Agent Identity Separation

> The practice of issuing AI agents their own distinct credentials rather than allowing them to operate under human credentials. Required for accurate attribution in audit records.

Delegation Chain

> In multi-agent orchestration, the sequence of credential delegations from an orchestrating agent to sub-agents, each linked to the original human authorization event.

Pre-Flight Verification

> Automated checks performed before a JIT credential is issued, confirming that the proposed agent session does not violate vulnerability policies, organizational change controls, or scope restrictions.

Steve Weltman, CISSP
Aletheia Security Consulting
sweltman@aletheiasecurity.com