Futurum
RESEARCH

Who Can Prove What the Agent Did?

Export Alone Does Not Prove an Agent Record Is Complete, Intact, or Authorized: Five Checks for SRE, Security, Governance, and Operations

Analyst(s): Vikram Rathnam
Publication Date: November 23, 2026
Document #: AIOVR202611

What You Need to Know

- **Export Alone Does Not Establish Completeness, Integrity, or Accountability:** A team can hold an agent's telemetry in its own tools and still have to reconstruct who authorized an action, what happened, and how far to trust what survived.
- **The Record Must Tie Each Action to Its Authority:** That authority is a person's approval or a policy approved in advance. The record must show that it covered this action on this target at that time.
- **Five Checks Show What a Record Establishes and What It Leaves Open:** Export, Connect, Missing, Altered, and After Exit address different evidence needs. Passing one does not establish that the others pass.
- **Integrity Checks Exist for Infrastructure Logs but Not in the Agent Documentation We Reviewed:** The reviewed pages describe parts of the record but do not provide independently verifiable evidence across the full workflow of a production agent.
- **Independent Proof Matters Most at Four Moments:** They are an incident investigation, a disputed authorization, an audit, and a platform exit. The evidence has to be captured before any of them arrive.

The Futurum View

Futurum's view is that teams responsible for production agents need records that enable them to reconstruct an action and verify their findings, even when those records span multiple products. Export is the starting point for five checks used by site reliability engineering (SRE), security, governance, and operations teams. An export shows what a platform captured. On its own, it can't prove what the agent did.

Teams Can Have Agent Telemetry and Still Not Know What Happened

At KubeCon + CloudNativeCon North America, 13 vendors briefed us, and we asked as many of them as we could whether the team running an agent can rebuild what it did in production from the records the vendor provides, and tell if anything is missing or was changed. The answers were mostly about getting data out or following an agent's steps: traces sent via OpenTelemetry, audit logs forwarded to a security tool, and an audit trail of the agent's actions. That answers the first half of the question. In most briefings, the second half either didn't come up, went unanswered, or was described as future work.

The second half is the operational problem. Take a hypothetical case. At 2 a.m., an on-call engineer approves an agent's request to scale down a node pool that is driving up cloud costs. The agent drains and deletes the pool, including nodes running another team's batch job. The trace records every call the agent made. The approval is a chat message that says "scale down," not "delete." The following week, someone edited the policy that had let the agent delete nodes, and no one kept the version in effect that night. Within an hour, the SRE rebuilds the sequence. Security can't show that the trace wasn't sampled or edited, and the auditor can't show that the approval covered deletion. Two engineers spend three days piecing the rest together from chat history and memory, and the postmortem's answer to "was this authorized?" is "we think so."

Agents already act outside their approved scope in many organizations, and fewer than half of organizations name audit logging among their controls for agents. In Futurum Research's 2H 2026 Software Lifecycle Engineering Global Enterprise Decision Maker Survey (n=839, fielded June 2026, published July 2026), 75% of organizations had a production incident in the previous 12 months in which AI-generated code, AI agent actions, or AI tooling was a contributing factor. Among those organizations, 36% reported AI agents executing unintended or out-of-scope actions. When asked which controls they have in place for agents acting in their software development environment, 45% named audit logging of agent actions.

Mitch Ashley's Seven Principles of Observability-Native made Open Standards-Based Interoperability its fourth principle, relying on open standards to enable agent telemetry portability, and most vendor answers we heard rested on OpenTelemetry. Portability is necessary. Futurum's position is that agent deployment is paced by what an organization can observe, control, and prove, and portability serves the first of those. The proving is verification debt in operational form, where records pile up faster than anyone checks them.

The Record Has to Tie the Action to Its Authority

We went into KubeCon assuming agents had moved from suggesting changes to making them. Futurum's own survey had already said otherwise. When asked for the dominant mode of AI use in their software engineering organization, 47% of organizations named individual developer assistance, such as code completion and chat. Supervised agents that carry out

©2026 The Futurum Group, LLC. All rights reserved.

discrete tasks with human approval were the dominant mode at 18%. Supervised, semi-autonomous, and fully autonomous agents together accounted for less than 40%, and end-to-end autonomous agents for under 6%. The KubeCon briefings matched the data.

That correction changes what the record shows. Human approval does not eliminate the need for an action record, because the team still has to link what was approved to what the agent actually executed. Whether a person approves a change or a previously approved policy permits it, the record must show the authority for the action. Mitch Ashley's second principle, Complete Decision Cycle Capture, already includes the authorization that enabled an outcome in the record. In the same survey question, 32% of organizations named human approval gates for irreversible actions. As humans move from being in the loop to leading it, more of that authority will be written into policy, and teams need to preserve the policy decision and the authority under which it was made.

Naming the actor is not enough. A record that shows the agent and the person it acted for establishes who acted, and it leaves open whether they were allowed to. To close that, the record must show that the authorization covered this action on this target under the policy version and permissions in effect at the time. A signature on an expense report works the same way. It shows who approved the expense and leaves open whether they had the authority to approve that amount.

Public documentation covers parts of this. Docker's audit record reference links approval requests to their resolution and identifies the policy rule and enforcement mode. Its audit log page covers governed developer sandboxes, so this is an example of record design rather than proof of production-wide coverage. Google Cloud's Agent Identity documentation says logs identify both the agent and the user when the agent acts on the user's behalf. That establishes an attribution capability; the page does not establish a record of human approval or a historical snapshot of permissions. Shared accounts and delegated sign-in can obscure attribution when identity context is not preserved.

©2026 The Futurum Group, LLC. All rights reserved.

Five Checks and What Each One Establishes

Apply the five checks to a defined workflow and configuration. A pass in Table 1 applies only to that tested scope.

Table 1: Five Checks for an Agent Record

Check	The Team's Question	A Pass Establishes	A Pass Doesn't Establish
Export	Can we get the agent's records out in a format our own tools read?	The team holds a readable copy.	That the copy is complete or unchanged.
Connect	Can we follow one action from request through authority to result, naming the agent and the person?	The tested action can be reconstructed.	That all actions are captured or authorized.
Missing	Did the records we expected arrive?	Missing expected records can be identified.	That all relevant actions were captured.
Altered	Can we show that a stored record wasn't changed or deleted after a known checkpoint?	The checked records match that checkpoint.	That the original records were true or complete.
After Exit	Can we read and check the records for our full audit period without the product?	The records stay readable and checkable after exit.	That the original capture was complete.

Source: Futurum Research, November 2026

Missing depends on knowing what to expect. An action may be absent because instrumentation, logging settings, or sampling omitted it. A record that was captured may also be lost in delivery. Detecting that loss requires evidence of what should have arrived, such as a manifest, a list of the records a system says it sent. Comparing the agent's records with the audit log of the system it changed can expose more gaps, but either source may omit activity. Integrity checks have a starting point, too. A signature or a protected checkpoint, meaning a saved reference the team trusts, can detect later changes without proving the original record was accurate.

Record both the result and its evidence base. A vendor statement, product documentation, and an observed test support different levels of confidence. Mark an untested check as untested. For each test, retain the workflow, configuration, time window, expected records, and result so another team can repeat it. The five checks are also available as a one-page test sheet with fields for the test scope, result, evidence basis, and gap owner.

What We Reviewed and What It Did Not Establish

Of the seven documentation pages we reviewed, only the Amazon Web Services (AWS) CloudTrail page documents a way for the operating team to check that a stored record is

©2026 The Futurum Group, LLC. All rights reserved.

unchanged, and that page covers infrastructure logs. None of the seven documents how to check agent approvals and policy decisions the same way, or how to verify the whole agent workflow together.

The evidence comes from briefings with 13 vendors at KubeCon and seven public documentation pages from six vendors, reviewed in November 2026: Docker's audit record reference and audit log page, DuploCloud's HelpDesk audit trail, Pulumi's audit logs, Google Cloud's Agent Identity overview, Splunk's OpenTelemetry page for Agent Observability, and AWS's CloudTrail log file integrity validation. The focus is on agents acting on production infrastructure; the developer sandbox and platform audit logs provide examples of individual capabilities. Coding-session repository provenance, model reasoning text as proof, and vendor rankings are outside the scope. By the whole agent workflow, we mean the request, the authority behind it, the policy decision, the execution, and the outcome. Other documentation or customer implementations may provide additional evidence.

The pages document export and parts of attribution and linkage. Docker describes local files, cloud-based export, and forwarding for governed developer sandboxes. Pulumi documents downloadable audit logs and, in its Enterprise edition, continuous delivery to Amazon S3 or Microsoft Sentinel. DuploCloud describes write requests and actor identities for its AI Studio and HelpDesk application programming interface (API), separately from infrastructure audit logs. Its logger sink is always on; additional destinations are off by default, and routine read requests require configuration. Splunk's cited page explains how to send traces to Agent Observability. It does not establish how to export the resulting product record.

The documentation also explains why a page review cannot determine a product-level pass or fail. DuploCloud's page states that a comma-separated values (CSV) export from its console is limited to 200 rows, so a team relying on that export needs to know the limit. Pulumi describes its logs as immutable but does not provide an independent integrity-checking procedure on the cited page.

AWS documents one existing mechanism. CloudTrail hashes delivered log files and supplies digitally signed digest files that customers can validate. This can detect changes or deletions to the covered files after delivery. It does not establish that every relevant action was logged or that those files are connected to approvals and decisions held in other systems. Enabling digest delivery and actually running validation are separate steps. Agent platforms could build on such mechanisms and document how their records connect to the rest of the workflow.

When Independent Proof Is Worth Its Cost

Independent verification has an ongoing cost. Keeping and protecting agent records requires storage, engineering time, and retention management. A team with effective controls inside its existing platform may see little reason to maintain another copy. The relevant test is whether its evidence will remain available and checkable when needed.

Evidence has to be captured before an investigation needs it. Teams should budget for capture and retention as ongoing work, and then test independent verification against four situations.

©2026 The Futurum Group, LLC. All rights reserved.

During an incident, the SRE must reconstruct the sequence of events even if the primary platform is unavailable. In a disputed authorization, the team has to show what permitted or denied the action. For an audit, a reviewer has to be able to examine the supporting records. After a platform exit, the organization must still interpret and validate the retained evidence.

Security and governance leads should set the required assurance by the consequences of the agent's actions, the systems it can reach, and the organization's obligations. Teams can begin with high-impact workflows and test how much their existing controls establish.

What Each Team Should Do Now

SRE: Run a Reconstruction Drill. Once a quarter, pick one real agent action and rebuild it from records alone: request, authority, policy decision, execution, and outcome. Measure the time it takes and count the steps that require someone's memory. Assign an owner to each gap the drill exposes.

Security: Test Integrity Verification. Identify the stores holding agent records and their verification mechanisms. Enable and run validation where supported. In a controlled test copy, check whether alteration or deletion is detected. Measure the share of sampled records covered by successful validation.

Governance: Verify the Authority for the Action. Sample agent actions and require the record to show the approval or policy that covered the action and target, its version, and the permissions in effect at the time. Measure the share of sampled actions where it does.

Operations: Test Retention and Exit Access. For high-impact workflows, retain records under agreed access and retention controls. Preserve schemas, policy references, and any verification material needed after exit. Measure the share of sampled workflows whose records can be read and checked without the originating product.

Vendors: Publish Repeatable Verification Procedures. Explain each check's prerequisites, supported scope, and known limits. Provide sample records and a procedure customers can repeat. Measure the share of the five checks with a documented, reproducible test.

What to Watch

- OpenTelemetry's Agent Conventions: The generative AI agent conventions are still under development in the project's dedicated repository. Watch whether instrumentation and receiving tools adopt compatible versions; stable conventions alone would not establish complete capture.
- Completeness of Capture: Watch whether platforms report delivery gaps, sampling decisions, and disabled instrumentation, with a clear account of what those signals can and cannot detect.

©2026 The Futurum Group, LLC. All rights reserved.

- **Agent and Person Named Separately:** Google Cloud documents both identities for delegated actions using Agent Identity. Watch whether other implementations preserve that distinction when agents use delegated credentials or shared accounts.
- **Integrity Validation for Agent Records:** Signed digests are documented for infrastructure logs. Watch for an agent platform that documents the same for approvals and policy decisions.
- **Security and Audit Teams Setting the Bar:** Watch whether teams make independently checkable records a condition of production write access. That would support our expectation that security and audit teams will demand them. Continued reliance on in-product evidence would challenge that expectation without resolving the technical limits of export alone.

Other Insights from Futurum

Observability: AI Is Eating the Budget That Keeps AI Safe

Claude Agent Observability Can't Be the Audit Record on Its Own

The Hard(er) Challenge in Agent Governance Is Authorization

©2026 The Futurum Group, LLC. All rights reserved.

About Us

About the Authors

Vikram Rathnam is Research Director for Software Lifecycle Engineering at The Futurum Group. His coverage focuses on two areas: AI-driven software development and observability. He studies how AI agents change the way software gets built and run, who owns the outcome when an agent acts, and what evidence shows whether the agent did what it was supposed to do. He came to analyst work from the vendor side. Over 24 years, he worked first as an engineer, including at Silicon Labs, and later in product and partner strategy at Cisco, Secureworks, Dell, and CMIT Solutions. He can be reached at vrathnam@futurumgroup.com.

About The Futurum Group

Every day, The Futurum Group's analysts, researchers, and advisors help business leaders worldwide anticipate tectonic shifts in their industries and leverage disruptive innovation. Unlike traditional analysts, The Futurum Group works not only in analysis and research but also takes that insight and knowledge even further, engaging all the way through the go-to-market process.

Futurum Research provides in-depth research and insights on global technology markets using advisory services, custom research reports, strategic consulting engagements, digital events, go-to-market planning, and message testing. It also creates, distributes, and amplifies rich media content that all stakeholders read, watch, and listen to.

See more details on The Futurum Group at futurumgroup.com.

Copyright & Use License

Copyright Notice

Copyright ©2026 by The Futurum Group, LLC. All rights, including that of translation into other languages, are specifically reserved. No part of this publication may be reproduced in any form, stored in a retrieval system, or transmitted by any method or means, electrical, mechanical, photographic, or otherwise, without the express written permission of The Futurum Group futurumgroup.com. United States copyright laws and international treaties protect this publication. Unauthorized distribution or reproduction of this publication, or any portion of it, may result in severe civil and criminal penalties and will be prosecuted to the maximum extent necessary to protect the publisher's rights.

License Notice

This document may be distributed within the licensed organization only.

The following acts are prohibited:

Transmittal to others outside your immediate organization, including partners, resellers, external consultants, etc., in any media format

Posting on a website that is accessible to others outside your immediate organization

The possession or use within an unlicensed organization

Limitation of Liability Notice

The information contained herein has been obtained from sources believed to be reliable. The Futurum Group shall have no liability for errors, omissions, or inadequacies in the information contained herein or for interpretations thereof. The reader assumes sole responsibility for the selection of these materials to achieve their intended results. The opinions expressed herein are subject to change without notice.