Agent Attribution for Healthcare Audit

A FHIR R5 AuditEvent Profile for Auditing AI-Mediated Access to Protected Health Information

Tanmaya Kumar

Behavioral Health Open Source · bh-healthcare.org

July 2026 · Technical Report BHOS-TR-2026-03

Repository: github.com/bh-healthcare/bh-audit-schema

ABSTRACT

Healthcare audit standards, including FHIR R5 AuditEvent, model a single actor per event. That model holds only while the identity that authenticated, the entity that acted, and the person whose intent is carried out are the same human. When an AI agent operates clinical software those identities separate, and one actor field collapses them back together, so the record can no longer name the accountable human apart from the acting agent. bh-audit-schema v2.0 makes agent-mediated action attributable through a three-role model, authenticating, acting, and authorizing, with validation that rejects an unattributed agent action by construction. It ships as a JSON Schema producer contract and as a FHIR R5 AuditEvent profile. HL7's `auditevent-OnBehalfOf` extension does not close the attribution gap so much as relocate it: applied to an agent action it moves the `requestor` flag onto the agent, so the most standard access-review query in FHIR returns the machine instead of the accountable clinician. That shape is already visible in published healthcare AI governance work. A schema cannot detect a non-cooperating agent, and the specification states the limit and defines emission tiers around it. Telemetry from a production audit deployment shows why the guarantee belongs in the schema and not in guidance: data classification is an optional emitter argument, the two highest-volume services never pass it, and 99.8% of events are therefore unclassified.

---

1. The problem

When an AI agent operates clinical software, the audit log can no longer say who did what.

Every audit standard in use today was designed around a single actor per event. That model is correct as long as three identities coalesce: the identity that authenticated, the entity that performed the action, and the person whose intent the action

represents. For a clinician clicking through a chart, those three are the same human, and one actor field captures all of it.

Agentic AI fractures that assumption. An agent can drive a graphical interface inside a clinician's authenticated session, call an API under its own service credential, or run scheduled work with no human present. In each case the three identities separate. One field is left holding three answers, and nothing in the record says which one it took.

The consequence shows up in the most ordinary query in any HIPAA access review: show me everything user Y did. Once agents are in the environment that query quietly returns the wrong answer. It cannot separate what Y did personally from what an agent did while holding Y's credentials, and nothing in the record marks the difference. The query does not fail. It returns a clean list that is wrong, which is worse.

For behavioral health data, which carries 42 CFR Part 2 protections and elevated stigma risk, an audit trail that cannot answer which entity actually touched this record, under whose credentials, and on whose behalf, is the difference between an auditable system and an unauditable one. This is not a future problem. Healthcare engineering, clinical operations, and revenue teams are already deploying agent-driven automation against existing web-based systems, and the audit infrastructure that should govern it does not yet exist in standard form.

2. Background and related work

Two FHIR resources describe who did what. Provenance records how a resource came to be, on the generation side, and AuditEvent records access and use. FHIR's delegation primitive, `Provenance.agent.onBehalfOf`, lives on the generation side. AuditEvent, the usage record, has no native delegation element. Agent access auditing is a usage problem, so the primitive is missing exactly where agentic deployments need it.

HL7 has recognized part of this. The published `auditevent-OnBehalfOf` extension for `AuditEvent.agent` adds a delegation reference to the usage record, which confirms the community sees delegation as a real gap in base AuditEvent.

Three adjacent efforts are worth situating against. IHE Basic Audit Log Patterns (BALP) profiles AuditEvent for common exchange patterns; it refines participation properties, where the profile in this paper refines participation semantics, so the two are complementary rather than competing. OAuth 2.0 Token Exchange (RFC 8693) defines an `act` claim for delegated authority at the credential layer; the profile's field semantics are deliberately compatible with it, so that when on-behalf-of token issuance for agents becomes common, the audit layer is already shaped to record it. Separately, a set of practitioner-led governance frameworks for healthcare

AI has begun publishing FHIR AuditEvent emission models of its own, and those published models turn out to be the most direct evidence available about how the standard behaves under agent traffic.

This work also builds on a prior foundation. bh-audit-schema v1 is a PHI-safe audit event standard for behavioral health systems, with control mappings to the HIPAA Security Rule, 42 CFR Part 2, and SOC 2, running in a production behavioral health deployment. The v2.0 work described here is the agent-attribution layer added to that base.

3. The three-role model

At the core of bh-audit-schema v2.0 is a three-role attribution model. Every action is described by an authenticating identity (whose credentials were presented), an acting identity (which agent performed the action), and an authorizing identity (the human whose intent the action represents).

When a human acts directly, all three collapse into one person, and the event carries a single agent the way a conventional audit record does. The common case stays simple, and the attribution machinery only appears when the roles actually separate.

The authenticating identity is not always a person. An agent calling an API under its own service credential authenticates as itself, so the authenticating and acting identities are both machines and only the authorizing identity is human. The model holds across that case and the browser-agent case without a special rule for either, which is the point of separating the three roles instead of adding a delegation pointer to a single actor.

The change is additive at the schema level, and it is deliberately a major version. In a v1 event, a missing delegation block asserted nothing, because delegation did not exist as a concept; an agent may or may not have been involved, and the record made no claim either way. In a v2 event, the absence of a delegation block is itself a claim, that no agent was involved, and what makes the absence meaningful is the version marker. The semantics of `actor` change with it: in v2 it names the authenticating identity only. A version number that changes what silence means is a major version.

The model is enforced rather than recommended. An agent action with no identifiable authorizing human is invalid and cannot be emitted as a conforming event, so attribution is a property guaranteed by schema instead of a convention policed by a reviewer. The authorizing identity is always a human, because an agent cannot bear authorizing accountability, and when an agent spawns sub-agents that identity stays the root human at every depth. An intermediate agent never becomes the authorizer, which is what stops accountability from being laundered through a chain.

Supervision is recorded and never assumed. An event carries whether the delegation was supervised, autonomous, or scheduled, so a reviewer can tell the difference between a human who was watching, a human who authorized and walked away, and work that ran with no human present at all. The three cases carry different risk and the record should not flatten them into "an agent did it."

The standard ships in two forms. A JSON Schema contract (Draft 2020-12) defines the event structure, the three-role model, and conditional validation rules that reject unattributed agent actions, sub-agent chains missing a parent, and human-override events that wrongly carry a delegation. A FHIR R5 AuditEvent profile expresses the same model in HL7's terms. A reference translator converts conforming events into R5 AuditEvent resources, and a validation suite exercises the schema against positive and negative corpora and runs the translator on every commit in continuous integration.

4. The FHIR R5 profile, and why onBehalfOf is not enough

The profile constrains existing R5 elements where they suffice and adds extensions only where no element exists. It uses R5's first-class `patient` element (a genuine improvement over R4 for audit), R5's `authorization` element for purpose of use, and R5's outcome structure with the standard `audit-event-outcome` codes. Attribution roles are carried by three agent slices discriminated on `agent.type`. Supervision state, agent session identifiers, chain depth, and an agent descriptor are carried by a bounded extension set covering only the genuine gaps.

HL7's `auditevent-OnBehalfOf` extension does not close the attribution gap so much as relocate it. One action, recorded three ways, shows it. An AI browser agent, operating inside a clinician's authenticated session, appends an addendum to a patient note. Same event, same clinician, same note. Only the `agent` array differs.

Recorded as a stock R5 AuditEvent. One agent. The clinician is the requestor. The agent that actually wrote the note does not appear anywhere in the record.

```
"agent": [
  {
    "who":       "user_123",
    "role":      [ { "text": "clinician" } ],
    "requestor": true
  }
]
```

Recorded with the `auditevent-OnBehalfOf` extension. The delegation is now visible, and this is a real improvement. But the agent has taken the `who` slot, the clinician has been demoted into an extension, and `requestor` has moved with the agent. A consumer running the most standard access-review query in FHIR, filtering on `requestor = true`, is handed a browser agent instead of the accountable clinician.

```
"agent": [
  {
    "who":       "agent_inst_4471",
    "requestor": true,
    "extension": [
      {
        "url":            "http://hl7.org/fhir/StructureDefinition/auditevent-OnBehalfOf",
        "valueReference": "user_123"
      }
    ]
  }
]
```

There is one `who` slot and three identities competing for it. The record goes from naming the human and losing the machine, to naming the machine and demoting the human. Neither version answers whose credentials were presented, whether anyone was supervising, or which agent session this belonged to.

Recorded against this profile. Three attribution-typed agent slices, and `requestor` restored to the accountable human. Supervision state, agent session, and chain depth are carried as extensions on the event.

```
"agent": [
  {
    "type":      "authenticating-identity",
    "who":       "user_123",
    "role":      [ { "text": "clinician" } ],
    "requestor": false,
    "networkString": "203.0.113.42"
  },
  {
    "type":      "acting-identity",
    "who":       "agent_inst_4471",
    "requestor": false,
    "extension": [
      {
        "url": "agent-descriptor",
        "extension": [
          { "url": "interface",    "valueCode":   "ui_driving" },
          { "url": "model-family", "valueString": "model_y" },
          { "url": "harness",      "valueString": "browser_agent" }
        ]
      }
    ]
  },
  {
    "type":      "authorizing-identity",
    "who":       "user_123",
    "requestor": true
  }
],
"extension": [
  { "url": "delegation-type",  "valueCode":       "supervised" },
  { "url": "agent-session-id", "valueString":     "agsess_01J9X0002" },
  { "url": "chain-depth",      "valuePositiveInt": 1 }
]
```

Abbreviated for legibility across all three specimens. In the resource, `who` is a Reference carrying an opaque identifier, attribution roles are CodeableConcept codings on `agent.type`, the agent descriptor carries vendor and version fields omitted here, and every extension carries its full canonical URL.

The same human occupies two slices, authenticating and authorizing, and the profile does not collapse them. The redundancy is deliberate: a consumer filtering on the authorizing slice writes one query whether or not the two identities happen to coincide. The connecting IP address attaches to the authenticating slice, which is the credential that opened the connection, so the network evidence still points at the clinician's browser while the acting identity names the machine inside it.

The same action, put to the questions a compliance officer has to answer:

Question	Stock R5	R5 + OnBehalfOf	This profile
Whose credentials were presented?	user_123	not recorded	user_123
Which entity performed the action?	not recorded	agent_inst_4471	agent_inst_4471
Who is accountable (`requestor`)?	user_123	agent_inst_4471	user_123
Was a human supervising?	not recorded	not recorded	supervised
Which agent session?	not recorded	not recorded	agsess_01J9X0002

4.1 The same shape, in the field

This is not a hypothetical failure. NHID-Clinical, a practitioner-led governance framework for healthcare voice AI published in June 2026, emits FHIR R4 AuditEvent bundles and documents its agent structure openly: three agent slices carrying the AI voice agent as the requestor, the payer system as the destination, and the provider organisation in the on-behalf-of position when an NPI is present.

Within the standard as written, that is a defensible mapping, and publishing an audit model in that much detail is more than most deployments do. It is also the exact shape described above, in R4, where `agent.who` and `agent.requestor` behave as they do in R5. An access review filtered on `requestor = true` across those bundles returns the voice agent. Here the accountable human is not demoted into an extension so much as absent from the record entirely, because the on-behalf-of slot holds an organisation and an organisation cannot hold intent.

When an independent effort, working carefully and documenting its choices, lands on the structure this section predicts, the result says something about the standard and nothing about the implementer.

5. What this is not, and what it cannot do

No part of this standard is a model. It employs no machine learning, no natural language processing, and no generative component. It is the interoperability and governance layer that agentic AI systems emit into, and it is built to audit the behavior of large language model driven agents across several interaction surfaces: Model Context Protocol tool calls, UI-driving browser agents that operate a graphical interface as a human would, direct API and SDK agents, and orchestration in which an agent spawns sub-agents.

A schema can represent attribution but cannot detect a non-cooperating agent. To a target system, an agent driving a UI under a human's credentials is indistinguishable from that human. Only agent-side infrastructure knows a delegation exists and can emit the record. The UI-driving case is the sharpest expression of this limit: because the event is structurally identical to genuine human action, the schema can-

not conditionally require a delegation without also rejecting legitimate direct actions. Truthfulness of that case rests on the emitting layer, not on the schema.

Because that is true, a deployment's attribution guarantee is only as strong as its weakest emission path, and claims about it should be tier-specific.

Tier	Mechanism	Guarantee
Enforced	Agent traffic passes through a gateway or middleware (MCP gateway, API proxy, instrumented SDK) that constructs and emits the attribution event; the agent cannot reach the target system without emission occurring	Attribution guaranteed for all traffic on the controlled path
Instrumented	The agent harness or runtime emits through hooks; cooperative but structural	Attribution guaranteed for actions through the instrumented runtime
Advisory	Rules or instruction files direct agents to self-report	No guarantee; an instruction to self-report is an honor system in a text file

Advisory templates are worth publishing as an adoption on-ramp, labeled for exactly what they are. They are not audit infrastructure. A reference implementation should target the enforced tier first, because the gateway is the strongest control point in current agent deployments.

6. A finding from production

The telemetry is opt-in by design, because a standard should not surveil the systems that adopt it, so nothing here is a population claim about the field. Across the instrumented services reporting into a production audit deployment since audit telemetry began in April 2026, the services split cleanly on data classification, and the split is the finding. Several classify every event correctly. The two highest-volume services set no classification at all, falling back to an unclassified value, which is why 99.8% of events across the deployment are unclassified. Same library, same organization, different integration decisions.

Classification is an optional argument on the emitter library. The field works when it is passed, and an optional argument is one that does not get passed.

A schema cannot force a producer to classify correctly, any more than it can force PHI out of a free-text field. What it can do is make classification a first-class, queryable field, so that unclassified emission is visible and auditable instead of silently absent, and the emission tiers define where a gateway can enforce it. Making the omission visible is the contribution.

The telemetry itself records successful emissions and does not aggregate emission-time rejections, which are logged but never summed. Any count taken from it is a floor on validated volume and not a failure rate, and it should be read that way.

7. Regulatory and privacy implications

The work sits directly on federal healthcare regulation. The strongest legal anchor is HIPAA Security Rule 45 CFR 164.312(d), person or entity authentication, which requires verifying that a party seeking access is who it claims to be. Agent-mediated access under human credentials is precisely the case where the authenticated identity and the acting entity diverge, and the delegation record documents that divergence. The model extends 45 CFR 164.312(b) audit controls by distinguishing the performing entity from the credentialed identity and by recording supervision state as a queryable property.

The Security Rule in force is the 2013 version. The proposed overhaul (RIN 0945-AA22, published for comment January 6, 2025) would rewrite the technical safeguards, and the unified agenda now targets final action in 2027. The argument here does not depend on it. Person or entity authentication and audit controls are obligations under the rule as it stands today, and an agent operating under a clinician's credential is a gap in both of them today.

For behavioral health specifically, 42 CFR Part 2 governs substance use disorder records. The 2024 final rule aligning Part 2 more closely with HIPAA reached its compliance date on February 16, 2026, HIPAA's penalty structure now attaches to Part 2 violations, and enforcement authority sits with the Office for Civil Rights. Part 2 remains the more stringent regime in the areas that matter here, including the prohibition on using Part 2 records in legal proceedings absent consent or a court order. An agent-mediated disclosure of a Part 2 record is still a disclosure, and the question of whose consent authorized it takes a person as its answer. The authorizing identity is the field that holds that person. The UI-driving flag marks the access pattern carrying the weakest technical controls, for heightened review. The delegation record is also an operational evidence layer for governance frameworks including SOC 2 (CC6.1, CC7.2), NIST AI RMF, and ISO/IEC 42001, as supporting evidence and not as a claim of certification.

Regulators have no good way to write a rule about agent access today, because detecting when an agent is acting is not something a target system can do. If agent access is to be governed at all, it has to be governed at the point of emission. An emission requirement is enforceable where a detection mandate is not, and an auditor can test against it.

The privacy model is simple: no bodies. Prompts, model outputs, screenshots, and reasoning traces never enter an audit event; no field accepts them, metadata is con-

strained to scalar values with a bounded property count, and strict structural validation rejects any smuggled field. Identifiers are opaque throughout, route information uses templates and never raw paths, and error messages are required to be sanitized. Deployments that need prompt or context retention store it in a separately access-controlled artifact store and reference it by opaque identifier. The audit log is designed to remain useful without becoming a secondary repository of protected health information, which also reduces breach scope if the audit system itself is compromised.

Prompts, model outputs, screenshots, and rendered context are the agentic equivalent of request and response bodies, which is why the v1 rule carries into v2 unchanged instead of needing a new one.

8. Status and future work

As a specification, the standard is released, tagged v2.0.0, and the default schema version, which adopters can pin today. It is validated against a corpus of seven positive and thirteen negative examples, and the translator produces structurally valid R5 AuditEvent resources for all seven, checked in continuous integration on every commit. The agent-attribution capability has not yet run against live agent traffic. That first deployment is the near-term work, and until it happens the capability is a prototype with a production base underneath it.

The profile's canonical URLs are identifiers, not published endpoints, and do not currently resolve. Publishing the profile as a formal FHIR Implementation Guide, with dereferenceable StructureDefinitions and CodeSystems, is future work, and it is the kind of artifact that properly comes out of community adoption rather than ahead of it.

Four pieces of work follow from where the specification now stands, roughly in dependency order.

An enforced-tier reference implementation. Gateway middleware sitting in front of MCP tool calls and agent API traffic, constructing and emitting the attribution event so that the agent cannot reach the target system without emission occurring. This is the only tier that produces a guarantee rather than a convention, and it is the tier a compliance officer can be shown.

A first deployment against live agent traffic. Everything in Section 6 comes from human-and-service traffic. The attribution layer has no observed agent behavior behind it yet, and the integration decisions that turn out to matter will not be visible until it does.

Published terminology and an Implementation Guide. Dereferenceable canonicals, a value set for the attribution roles and delegation types, and the profile invariants in publishable form.

Two schema-level gaps worth raising with the FHIR community. R5 outcome codes have no value for a correct authorization denial distinct from operational failure, so a system that correctly refuses an agent looks like a system that broke. The translator works around this today by mapping a denial to the serious-failure code and attaching a `denied` coding in `outcome.detail`, which preserves the distinction for any consumer that knows to look for it and loses it for every consumer that does not. A base-level code would be the durable fix. Separately, base AuditEvent carries no PHI-minimization discipline against agent-observed content, which becomes a live problem the moment agents that see screens start emitting what they saw.

A version of this work was submitted to the HL7 AI Challenge, 2026.

Companion paper: "BH Audit Schema: An Open Standard for PHI-Safe Audit Logging in Behavioral Health Systems," bh-healthcare.org.

Suggested citation: Kumar, T. (2026). Agent Attribution for Healthcare Audit: A FHIR R5 AuditEvent Profile for Auditing AI-Mediated Access to Protected Health Information. Behavioral Health Open Source.

References

1. HL7 International. *FHIR R5 (v5.0.0), AuditEvent resource.* hl7.org/fhir/auditevent.html
2. HL7 International. *AuditEvent OnBehalfOf extension.* `http://hl7.org/fhir/StructureDefinition/auditevent-OnBehalfOf`
3. HL7 International. *FHIR R5 (v5.0.0), Provenance resource.*
4. IHE IT Infrastructure. *Basic Audit Log Patterns (BALP) Implementation Guide.*
5. IETF. *RFC 8693, OAuth 2.0 Token Exchange.* January 2020.
6. U.S. Department of Health and Human Services. *45 CFR 164.312, Technical safeguards (HIPAA Security Rule).*
7. HHS Office for Civil Rights. *Modifications to the HIPAA Security Rule to Strengthen the Cybersecurity of Electronic Protected Health Information.* Notice of Proposed Rulemaking, RIN 0945-AA22, published January 6, 2025. Not finalized as of July 2026.
8. HHS / SAMHSA. *Confidentiality of Substance Use Disorder Patient Records, 42 CFR Part 2, Final Rule.* Effective April 16, 2024; compliance date February 16, 2026. hhs.gov/hipaa/part-2

9. Baynard, B. *NHID-Clinical: A Behavioral Baseline for Healthcare Voice AI, v1.3.* June 2026. CC BY 4.0. nhid-clinical.org
10. NIST. *AI Risk Management Framework (AI RMF 1.0).* NIST AI 100-1, January 2023.
11. ISO/IEC. *ISO/IEC 42001:2023, Information technology, Artificial intelligence, Management system.*
12. Kumar, T. *bh-audit-schema.* github.com/bh-healthcare/bh-audit-schema