Nick Clark · Published July 23, 2026 · PDF

The Derived-Artifact Provenance Chain and Chain-Head Identifier

When a governed agent registers another agent's inference artifact as a tool, that artifact is usually derived: a base model published by one party, then retrained by others before it reached the consuming agent. The derived-artifact provenance chain, disclosed in U.S. Provisional Application No. 64/117,812, gives the consuming agent a tamper-evident, locally verifiable record of that derivation history. It is an ordered sequence whose first entry names the base publisher over a signed digest of the base artifact and whose every later entry records one retraining event, each retraining agent signing over both the artifact it produced and a digest of the preceding entry. A chain-head identifier digests the head and its predecessor, committing to the entire chain so that any altered entry yields a different identifier.

1. The Gap

A semantic agent that registers another agent's inference artifact as a tool inherits everything the artifact carries. In practice the artifact is rarely original. A base publisher released a model, and one or more agents thereafter retrained it against corpora they held, each redistributing the result onward. By the time the artifact reaches the consuming agent and begins producing determinations the agent relies on, its behavior is the accumulated product of retraining events the consuming agent never witnessed and cannot inspect.

The account of that history, where one exists at all, is a model card, a registry entry, or a line in a spreadsheet. It is human-authored, editable after the fact, held by whoever hosts it, and never checked at the moment of use. Nothing binds the claimed history to the bytes actually registered, and nothing stops a party from rewriting an earlier step. An agent that must answer for its own conduct cannot answer for a tool whose derivation it takes on faith.

The gap the filed disclosure addresses is a record the consuming agent can verify itself, from material it already holds, every time it touches the artifact, with no registry to query and no upstream party to contact.

2. Mechanism

The consuming persistent semantic agent (100) that registers an inference artifact obtained from a source agent maintains, against that endpoint, a derived-artifact provenance chain: an ordered sequence of entries held with the endpoint record.

The first entry names the base publisher. It carries the publisher's signature over a digest of the base artifact, together with the digest so signed. This anchors the chain to the artifact as it was originally published, under the publisher's own key.

Each subsequent entry records one retraining event by one retraining agent, and a plurality of retraining agents is permitted across successive entries. A retraining-agent entry carries five elements: the retraining agent's identity primitive; its epoch identifier at the retraining event; a reference to the entry that the same agent appended to its own append-only lineage field (104) recording that event; a digest of the artifact as retrained; and that agent's signature over the foregoing elements together with a digest of the immediately preceding entry of the chain.

The final element is what makes the sequence a chain rather than a list. Each retraining agent signs, under its own key, over its own entry, which carries a digest of the artifact it produced, together with a digest of the entry immediately before it. An entry therefore commits to its predecessor, the predecessor to its own predecessor, and so on down to the base. Altering any earlier entry changes that entry's digest, which invalidates the signature of every entry after it. The history is tamper-evident from the inside: no party can quietly revise a step it has already signed over, and no downstream party can insert or reorder a step without breaking a signature.

The reference to the retraining agent's own append-only lineage entry, the third of the five elements, does a second kind of work. The chain does not merely assert that an agent retrained the artifact; it points to the entry that agent recorded in its own history for the same event. The claim in the chain is corroborable against the agent's independent record rather than resting on the chain alone.

The chain head is the last entry, being the retraining-agent entry of the most recent retraining event, or the base-publisher entry where no retraining event is recorded. Its chain-head identifier is a digest computed over the head and its predecessor, and it commits to the whole of the chain. Because each entry already binds its predecessor's digest, a digest over the head transitively fixes every entry beneath it, so a chain differing at any entry yields a differing chain-head identifier. Two artifacts identical in their current bytes but different in how they were derived carry different chain-head identifiers, so provenance is part of the artifact's identity as the consuming agent holds it.

The consuming agent verifies the chain at each lifecycle operation upon the endpoint, including registration, deregistration, substitution, and each dispatch of an inference request. The verification proceeds in order. First it tests the base entry's signature against its digest. Then, from the first entry through the chain head, each entry's signature is checked against that entry's elements and the digest of the preceding entry. For any retraining agent that resolves to a counterparty identity record (114) the

consuming agent holds, the epoch identifier presented must be a valid successor of the epoch recorded there. The head's artifact digest must equal a digest over the artifact as registered. Last, the chain-head identifier is recomputed. On any failed step the operation is not performed, and the attempt, the failing element, and the entry index are appended to the append-only lineage field (104) as an outcome that is non-adverse to every party named in the chain.

Two properties of that verification matter structurally. It runs over material the consuming agent already holds, being the chain in its endpoint record and whatever counterparty identity records it keeps for parties it has met; it queries no registry and contacts no upstream party. And it binds the chain to the registered bytes: the head's artifact digest must equal a digest over the artifact as actually registered, so a chain cannot be verified against an artifact it does not describe.

3. Operating Parameters

The chain is an ordered sequence. Its first entry is the base-publisher entry and carries three elements: the publisher identifier, the publisher's signature, and the digest signed. Each subsequent entry is a retraining-agent entry carrying the five enumerated elements above. None of those elements, for either entry type, is a quality, performance, accuracy, or standing measure of any party. The entries record identity, epoch, cross-reference, digest, and signature, not any evaluation of the parties they name.

A plurality of retraining agents is permitted across successive entries. The mechanism places no numeric bound on entry count; chain length follows the number of recorded retraining events.

The chain-head identifier is a digest over the head entry and its predecessor. The verification cadence is every lifecycle operation upon the endpoint, the filed spec enumerating registration, deregistration, substitution, each dispatch, and each other governed lifecycle operation. The epoch-successor check is conditional: it applies to a

retraining agent only where that agent resolves to a counterparty identity record (114) the consuming agent holds, so an agent the consumer has never met is verified by signature and digest alone.

On each dispatch the consuming agent records a dispatch provenance entry carrying the then-standing chain-head identifier, the endpoint identifier, the scope partition, an identifier of each determination produced in reliance on the response and its action class, the recorded dispatch time, and the agent's own successor epoch identifier at dispatch. These entries support a retrieval keyed by chain-head identifier, computed from the append-only lineage field (104) alone, requiring no registry and no communication with any party.

4. Composition

The chain-head identifier is the joint on which the rest of the section turns. Because each dispatch records the head then standing against the determinations produced from it, the consuming agent can later retrieve, by head identifier alone, every determination that relied on a given state of a given artifact. That retrieval is what makes provenance withdrawal actionable. When a party named in the chain emits a provenance withdrawal as a governed observation naming a chain-head identifier, the consuming agent already holds, in its own lineage, the exact set of determinations to re-mark, without asking anyone. The provenance-withdrawn attribute and the dispatch-authority predicate that follows from it both key off the identifier this mechanism produces.

The epoch-successor check reuses the same successor relation the agent applies to its own identity continuity, so provenance verification is not a separate trust apparatus but an application of the epoch continuity the filing already relies on. The freshness check for a party that ceases to stand behind an artifact without emitting a withdrawal tests the same successor relation against a presented epoch identifier.

The chain also composes with the corpus-contribution mechanism disclosed later in the same section through the shared endpoint record. Where the provenance chain records who retrained an artifact and in what order, a retraining event's recorded contribution set records which settled contributions it incorporated. One captures who performed each retraining; the other captures what that retraining was trained on. Because both live in the endpoint record, an agent occupying both the consuming and the receiving role holds a single record over which provenance and contribution accounting run without either deriving structure from the other.

The dispatch provenance entry threads all of this into the append-only lineage field (104) that already carries the agent's determinations, so a reviewer reconstructs both what the agent decided and which artifact state it relied on from one record.

5. Prior-Art Distinction

Documentation of model origin is not new. A verifiable, locally checked chain of it is the point of distinction.

Human-authored model documentation, including model cards, dataset datasheets, and model-registry metadata, describes an artifact's origin in prose or structured fields. It is mutable after publication, hosted by whoever maintains the registry, and consulted, if at all, by a human rather than checked by the consuming system at each use. It carries no per-hop signature and offers no way to detect a rewritten earlier step.

Software supply-chain attestation frameworks, such as build-provenance and in-toto or SLSA style attestations, do sign steps and do aim at tamper evidence. Their subject is a software build pipeline, and their verification typically leans on a transparency log or a hosted verifier. The mechanism here differs in locus and target: the chain is held per endpoint in the consuming agent's own records, it records retraining of an inference artifact rather than build steps of a package, and its verification runs against material the agent already holds with no log or registry in the path.

Centralized model registries and experiment-lineage trackers store derivation as metadata in a shared service. They record lineage, but not as a sequence in which each retraining agent signs, under its own key, over both the artifact it produced and the prior entry, and the consuming party does not recompute a self-committing head over the whole history at each operation.

Certificate chains, as in the X.509 family, are structurally the closest analogue: a sequence of signatures each vouching for the next. They attest identity and authority bound to a key, not the derivation history of a retrained artifact, and no link binds a signer to an independent append-only record of the very event it certifies or to a digest of the artifact as that signer left it.

Distributed ledgers reach tamper evidence through a shared, consensus-maintained log. This mechanism reaches it locally: the tamper evidence is intrinsic to the hash-linked signatures, and no ledger, no consensus round, and no coordination among parties is required to verify a chain.

None of this asserts that any system infringes anything. The categories are set out to locate the specific structure the filing describes: a per-hop signed, hash-linked chain of retraining events, each entry cross-referencing the retraining agent's own lineage and the artifact digest it produced, closed by a chain-head identifier that commits to the whole and is recomputed locally at every lifecycle operation.

6. Disclosure Scope

The operative disclosure is U.S. Provisional Application No. 64/117,812. The derived-artifact provenance chain and the chain-head identifier are disclosed in Section 10.5 at paragraph [0395], with the chain verification at each lifecycle operation at [0396] and the carriage of the chain-head identifier into the consuming agent's lineage at [0397].

Disclosed there, and asserted here, is the following: an ordered chain whose first entry names the base publisher over a signed digest of the base artifact; each later entry recording one retraining event with the retraining agent's identity primitive, its epoch identifier at the event, a reference to that agent's own append-only lineage entry for the event, a digest of the artifact as retrained, and the agent's signature over the entry and the preceding entry's digest; a chain head that is the last entry; and a chain-head identifier that is a digest over the head and its predecessor and commits to the whole chain.

What is not claimed here is as important. The disclosure fixes no particular digest function or signature scheme; those are embodiment choices and no specific algorithm is asserted. It places no numeric bound on chain length. No filed entry element is a quality, performance, or standing measure. The companion mechanisms of the same section, being provenance withdrawal and the provenance-withdrawn attribute at [0398], the re-marking and dispatch-authority gate that follow, and the matched-pair-gated corpus contribution at [0403], are disclosed in their own paragraphs and are described here only where they compose with the chain. Certain fuller characterizations in the author's drafting chapter, including describing the identity primitive as a position on the retraining agent's trust slope and the rule that a single agent performing several retraining events contributes one entry per event, extend beyond the filed paragraph and are not asserted as part of this disclosure.

Governed Matched-Pair Settlement

All 49 steps → (/inventive-steps)

(/governed-settlement)

Two agents settle on matched signed observations, by continuity, with no clearer.

Provisional application

PRIMARY TECHNICAL DISCLOSURE

- Matched-Pair Settlement Between Agents: Bilateral Closure Without a Clearer (/articles/governed-settlement/matched-pair-settlement-without-a-clearer)

SECONDARY TECHNICAL

- **The Derived-Artifact Provenance Chain and Chain-Head Identifier (/articles/governed-settlement/derived-artifact-provenance-chain)**
- Provenance Withdrawal and the Provenance-Withdrawn Attribute (/articles/governed-settlement/provenance-withdrawal)
- Matched-Pair-Gated Corpus Contribution (/articles/governed-settlement/matched-pair-gated-corpus-contribution)
- Partition-Scoped Escrow Release Gated on Intersection Non-Emptiness (/articles/governed-settlement/partition-scoped-escrow-release)
- The First-Encounter Settled Pair and Promotion to the Persistent Tier (/articles/governed-settlement/first-encounter-settled-pair)

TERMINOLOGY

- governed observation (/glossary#governed-observation)
- matched pair (/glossary#matched-pair)
- scope partition (/glossary#scope-partition)

Governed Matched-Pair Settlement overview → (/governed-settlement)