Nick Clark · Published July 24, 2026 · PDF

The EU AI Act and Inter-Agent Conduct: Where Record-Keeping and Oversight Duties Stop

A high-risk AI deployment can satisfy its logging and human oversight duties in full and still have no answer for the moment one company's agent tells another company's agent that it caused harm. The record exists, nobody is reading it at runtime, and the accused system keeps dispatching. Chapter 1 of U.S. Provisional Application No. 64/117,812 discloses an architecture in which the accused agent admits that assertion over its ordinary interaction channel, verifies the signature and the asserting party's attested assertion-cost counter, tests the asserted conduct against its own append-only lineage field, and returns exactly one determination from a closed set of four. An accepted determination can move the persistent state that gates the agent's own next dispatch.

When the complaint arrives from another company's agent

A fulfillment agent operated by one company receives a message from a scheduling agent operated by another: you committed to a delivery window in a scope you were not authorized for, and my side absorbed the cost. The sender is not a regulator, not an auditor, and not the principal standing behind the receiving agent. It arrives on the same interface the two agents use for ordinary work, with no operator watching.

The default answer is that it goes into a log for later review, and the gap that leaves widens with autonomy: the accused agent keeps dispatching in the interval, each dispatch made by a system told it caused harm and holding no procedure for what that means to its own authority to act.

Records are not the scarce thing; traces and audit logs capture plenty. The difficulty is that a record answers a human question, asked later, about what happened, while a dispatch decision answers a machine question, asked now, about what may happen next. A counterparty's assertion belongs to the second and gets filed under the first.

A failure mode waits on the other side of that problem: an agent able to dismiss an assertion because its own record does not mention the conduct would turn accountability into a formality and poor record-keeping into a defense.

The regulation on its own terms

The EU AI Act is the European Union's horizontal regulation for artificial intelligence. As publicly described, it takes a risk-based approach, so obligations scale with the risk a system presents rather than applying uniformly. Public summaries describe practices prohibited outright, a high-risk tier carrying the heaviest substantive obligations, transparency duties for certain systems that interact with people or generate content, and a separate track for general-purpose AI models.

For systems in the high-risk tier, the obligations documented publicly form a coherent product-safety package: a risk management process across the system's lifecycle, governance of the data used to build it, technical documentation prepared and maintained, automatic recording of events over the operational lifetime with those records retained, information to deployers sufficient for proper use, human oversight designed in so that natural persons can understand the system's operation and intervene or stop it, and requirements addressing accuracy, resilience, and

cybersecurity. Public descriptions add post-market monitoring, reporting of serious incidents to authorities, and a conformity assessment step before a high-risk system reaches the market.

Two structural features matter for what follows. Duties attach to identifiable legal persons: the regulation, as publicly described, distinguishes providers from deployers and further addresses importers and distributors. And oversight is human by design, the recorded events existing so that people can reconstruct behavior.

That is a demanding and internally coherent design, and nothing here suggests a defect in it. What matters is the kind of instrument it is. As publicly described it is a market-access and accountability regime addressed to the organizations that build and deploy AI systems, and a regime of that kind works by allocating duties among legal persons. A runtime protocol governing how two independently operated agents speak to each other about conduct belongs to a different category, one such a regime does not set out to specify.

Intake, admissibility, and one determination from four

Chapter 1 of U.S. Provisional Application No. 64/117,812 describes a runtime layer in that second category. The unit is a persistent semantic agent carrying a memory field, an append-only lineage field recording executed actions and determinations, a scoped integrity vector with personal, interpersonal, and global components, a self-esteem aggregate, and a policy reference field resolving to a signed policy object. It holds a counterparty identity record per counterparty, carrying that party's identity primitive and its previously attested states. The policy object is authored and signed by a principal to which the agent is bound; the agent does not author it and modifies it only by admitting a successor.

A conduct evaluation intake receives, at runtime and over the agent's ordinary interaction channel, a conduct evaluation artifact from an asserting party other than the principal, asserting an evaluation of conduct performed by the agent itself rather than of an output, a task result, or a third party. No dedicated evaluation channel, no out-of-band reporting interface, and no centralized intake service is required.

The artifact carries enumerated fields and is consumed by reference to those alone: an asserting-party identifier; a recorded assertion time, which fixes the policy object in force; a conduct descriptor of action-class identifier, scope-partition identifier, and affected-party class; a signature verifiable against the identity primitive recorded for that party; and an attested state of that party's assertion-cost counter together with an epoch identifier of its dynamic agent hash chain. The descriptor names conduct by structural elements recorded in the lineage field rather than by natural-language characterization, and no language model or sentiment classifier is run over artifact content.

Admissibility runs as an ordered procedure. The signature is verified first, a non-verifying artifact being appended to the lineage field and not admitted; where the agent holds no record for that party, verification runs against a provisional identity primitive and the artifact is held inert pending a matched-pair settlement. The attested counter is verified second: an artifact whose attestation is absent, whose epoch identifier is not a valid successor of the epoch recorded for that party, or whose counter state falls below a state that party previously attested is likewise appended and not admitted. A non-admitted artifact produces no determination, moves no value of the scoped integrity vector, and increments no counter.

An admitted artifact reaches the admission evaluator, which produces exactly one determination from a closed set of four: accepted, rejected, not-determinable, and not-applicable, with no scalar confidence, probability, or graded weight standing in place of

one. It works over the agent's own lineage field against the declared value set of the policy object in force at the recorded assertion time, and not by way of an external authority, an arbitrator, a registry, or a scoring service.

A value-scope test comes first. A declared value is implicated only where the descriptor's action class and scope partition fall within that value's sets of action-class and scope-partition identifiers, and its affected-party class maps to that value's empathy-scope designation. Where no declared value is implicated the result is not-applicable. Otherwise the evaluator retrieves lineage entries recording an action of that action class within that scope partition. The affected-party class is no input to that retrieval, so an asserting party cannot narrow the entries retrieved against its own assertion by the class it declares. A rejected determination issues only where a retrieved entry affirmatively contradicts the artifact on a recorded field; an entry silent as to that field is not a contradiction. Absence of evidence resolves to not-determinable and to no other class, so the agent cannot refuse an artifact because its own record is silent, incomplete, or unavailable.

Every determination is appended together with the artifact, the entries retrieved, the class produced, and the ground of the result. On an accepted determination a state modifier moves the scoped integrity vector and the self-esteem aggregate, each modification scaled by an entropy-weighted harm coefficient, and a deviation engine recomputes a deviation likelihood from the modified values. Where that quantity satisfies the policy-declared bound, the authorization gate transitions to the withheld state for the affected action class and the agent enters a non-executing cognitive mode in which speculative evaluation continues without committing state changes. On a rejected determination a refusal counter is incremented, and that counter writes the authorization gate on satisfaction of a threshold.

Such writes reach the next action, not a later review. Responsive to each dispatch request the agent recomputes a dispatch-authority predicate from state then carried in its memory field, satisfied only upon a three-stamp conjunction, required in full, of a

policy stamp, a lineage stamp, and an authorization stamp, and computed from no previously issued token, no cached result, and no session grant. A write of the gate, an admission of a successor policy object, or an append to the lineage field therefore takes effect at the next dispatch request without revocation infrastructure.

Who reads the record, and when

Convergence deserves naming before contrast. Both designs treat the record as load-bearing rather than incidental, and both separate policy authorship from execution under it: the regulation, as publicly described, allocates distinct duties to the party that builds a system and the party that uses it, and the filed architecture puts authorship of the policy object with the principal.

Divergence is in who reads the record, and when. Under the regulation as documented publicly, recorded events are evidence, retained and read by deployers, overseers, and authorities after the fact. In the filed architecture the lineage field is an operand: the lineage stamp evidences that its most recent entry is committed and is the recorded successor of the entry named by the policy stamp, so an append changes what the agent may do next without a person reading anything.

Appraisal separates along the same line. Public descriptions place it with people and institutions: the natural persons exercising oversight, the conformity assessment step, the authorities receiving incident reports. Chapter 1 places a narrow, non-semantic determination in the accused agent itself, over its own record, then restricts what that determination is permitted to be. Four classes, and no scores.

The two do not compete for a slot: a regime assigning duties to legal persons and a mechanism governing what one agent does with another's assertion before its next dispatch answer different questions about the same events.

Running both, and the duties the architecture leaves untouched

In a deployment subject to high-risk obligations the two records stay separate, because their consumers are separate: the event log the regulation contemplates, as publicly described, is retained for people, and the lineage field for the agent's own predicate. The coupling point available is the escalation record: upon a transition of the authorization gate to the withheld state, an escalation emitter sends the principal a record enumerating the action class, the scope partition, and the entries upon which the transition was computed. The gate is maintained per action class and per scope partition, so a transition reaches the affected pairing rather than the agent as a whole.

What the architecture does not do deserves its own list. It classifies no system into a risk tier, produces no technical documentation, performs no conformity assessment, files no incident report with any authority, and establishes nothing about which legal person is a provider or a deployer. An accepted determination runs against the agent's own declared value set and lineage record: no finding by any tribunal, no legal liability established, no regulatory duty discharged.

Internal limits matter as well. Verification reaches the signature, the continuity of the attested counter and its epoch, and affirmative contradiction on a recorded field, so a formally admissible artifact can still be substantively wrong, and one the agent cannot resolve yields not-determinable rather than exoneration. A silent record is not a defense, and that choice carries a cost: repetition of not-determinable against one asserted conduct event increments a refusal counter under a later chapter of the same filing. Restoration also takes work: no elapse of time, and no payment, transfer, or consideration by any counterparty, returns the gate from the withheld state, which returns toward granting only upon a procedure appended to the lineage field that a deployment has to build.

One point bears repeating, since it is the easiest thing here to overstate. The consequence of an accepted determination is conditioned rather than automatic: it moves the integrity quantities and triggers recomputation of the deviation likelihood,

and the gate transitions to the withheld state only where that quantity satisfies the policy-declared bound. Not every accepted determination stops the agent, and a deployment that wants a given class of conduct to stop it has to declare the bound that makes it so.

Disclosure Scope

The mechanisms described are disclosed in U.S. Provisional Application No. 64/117,812, principally Chapter 1, at sections 1.2 through 1.9: the persistent semantic agent and its carried fields; the signed policy object and its ordered resolution under an anti-rollback monotonicity constraint; the authorization gate, the escalation emitter, and the restoration controller; the dispatch-authority predicate and the three-stamp conjunction; the conduct evaluation artifact, the dynamic agent hash chain, and the successor-continuity test; the ordered admissibility procedure and the pre-settlement inert state; the closed set of determination classes with its value-scope test and affirmative-contradiction rule; and the recording of each determination with its ground and its consequences for the scoped integrity vector, the deviation likelihood, and the refusal counter.

References to the EU AI Act are to public materials and are used for comparison only; no relationship, endorsement, or infringement is asserted.

This article is published to establish public, timestamped prior art for applying this architecture to inter-agent conduct accountability in regulated deployments. It describes a pending application. Nothing here asserts that any product, service, standard, regulation, or organization infringes anything, and nothing states that a license is required.

Record-Grounded Conduct Admission (/conduct-admission)

All 49 steps → (/inventive-steps)

Conduct judged against the agent's own record, not an outside authority.

Provisional application

PRIMARY TECHNICAL DISCLOSURE

- Record-Grounded Conduct Admission: Governing How an Agent Answers an Accusation (/articles/conduct-admission/record-grounded-conduct-admission)

SECONDARY TECHNICAL

- Value-Scope Resolution: Testing Whether a Declared Value Is Even Implicated (/articles/conduct-admission/value-scope-resolution)
- The Retroactive Narrowing Bar: Why an Agent Cannot Edit Its Values After the Fact (/articles/conduct-admission/retroactive-narrowing-bar)
- The Delegator Floor: A Monotone Upward Ratchet on Delegated Authority (/articles/conduct-admission/delegator-floor-ratchet)
- The Co-Signed Mutual Admission Compact: Two Agents Binding Each Other (/articles/conduct-admission/co-signed-mutual-admission-compact)
- The Inherited Governance Record: Fork- and Clone-Resistant Accountability (/articles/conduct-admission/inherited-governance-record)

APPLICATIONS · GENERAL

- Agent Marketplaces: Handling a Complaint Against an Autonomous Seller (/articles/conduct-admission/agent-marketplace-dispute-handling)
- Clinical AI Agents: Answering an Accusation Without a Central Registry (/articles/conduct-admission/clinical-agent-accountability)

APPLICATIONS · SPECIFIC

- **The EU AI Act and Inter-Agent Conduct: Where Record-Keeping and Oversight Duties Stop (/articles/conduct-admission/eu-ai-act-accountability-records)**

TERMINOLOGY

- append-only lineage field (/glossary#append-only-lineage-field)

- authorization gate (/glossary#authorization-gate)
- conduct evaluation artifact (/glossary#conduct-evaluation-artifact)
- principal (/glossary#principal)
- semantic agent (/glossary#semantic-agent)
- signed policy object (/glossary#signed-policy-object)

Record-Grounded Conduct Admission overview → (/conduct-admission)