Nick Clark · Published July 29, 2026 · PDF

Agentforce and the Accusation an Agent Must Answer

A customer tells a service agent that it misled them about a refund window. The complaint is not about a wrong answer; it is about conduct, and it arrives from someone who does not own the agent. Salesforce Agentforce, as publicly described, is a platform for building autonomous agents inside the Salesforce environment. U.S. Provisional Application No. 64/117,812 addresses a narrower question that can sit underneath any such platform: how an agent resolves a signed conduct evaluation artifact from a party other than its principal against its own append-only record and the signed policy object in force, producing exactly one determination from a closed set of four.

The complaint that is not about the answer

A customer writes back to the service agent that handled a return: you told me the window was open, I acted on that, and it was not. Nothing in that message is a request. It is an assertion about how the agent behaved, addressed to the agent, from a party with no authority over it.

Accountability machinery around autonomous service agents is usually organized around three jobs: keeping the agent from producing a bad output, recording what happened for a reviewer, and escalating a case that exceeds the agent's scope. Those are

the right targets for service quality, and they aim at a different object than the one the customer just raised, which is whether the conduct asserted occurred and what follows for the agent if it did.

The awkward part is who answers. Have a language model read the complaint and the outcome turns on the customer's phrasing. Put a central service in the loop and someone has to operate it and define who may file against whom. The question underneath both is structural: what is the determination computed from, what values does it run against, and what gets written down when the agent's own record has nothing to say.

What Agentforce sets out to do

Salesforce Agentforce is, as publicly described, a platform for building and deploying autonomous AI agents inside the Salesforce environment. Public materials present grounding as central: an agent answering about an order is described as working from the business's own customer records rather than a generic model prior. Builders are described as configuring the topics an agent may handle and the actions it may invoke, with a path for handing a conversation to a person when it falls outside those bounds. Salesforce also publicly describes a trust layer addressing how customer data is handled in the model interaction path.

For that purpose, this is a well-aimed architecture. The risks those controls are publicly positioned against are ones a deployment has to survive: policy invented on the spot, actions taken outside the intended scope, and customer data leaving where it belongs. Grounding, scoped topics and actions, human handoff, and a data trust boundary all land where a business administrator can reach them, which is configuration rather than model internals. For a CRM, serving the enterprise's relationship with its customers is the right framing.

The architecture described below starts from a different input, defined by the filing rather than by any product: an artifact signed by someone other than the principal, asserting conduct by the agent itself.

Answering on the record

The filed disclosure begins with a persistent semantic agent (100) carrying a memory field (102), an append-only lineage field (104) recording executed actions and determinations, a scoped integrity vector (106), a self-esteem aggregate (108), and a policy reference field (110) resolving to a signed policy object (112). In accordance with an embodiment, that policy object is authored by a principal to which the agent is bound, covered by that principal's signature, and modified by the agent only by admitting a successor.

Into that agent arrives a conduct evaluation artifact (116), received over the ordinary interaction channel, originating from an asserting party (118) other than the principal, and asserting an evaluation of conduct performed by the semantic agent (100) itself, as distinguished from an evaluation of an output, a task result, or a third party. The artifact carries an asserting-party identifier resolving to a counterparty identity record (114); a recorded assertion time, being the time by reference to which the policy object in force is identified; a conduct descriptor comprising an action-class identifier, a scope-partition identifier, and an affected-party class; a signature verifiable against an identity primitive recorded in that record; and an attested state of the asserting party's assertion-cost counter (400) with an epoch identifier of that party's dynamic agent hash chain.

Two properties of the intake do real work. The conduct descriptor identifies conduct by reference to structural elements recorded in the append-only lineage field (104), and not by natural-language characterization. And the architecture performs no inference of the state, intent, or affect of the asserting party from the content of the artifact, and no

classification of that content by a language model or a sentiment classifier; the artifact is consumed by reference to its enumerated fields alone. No dedicated evaluation channel or centralized intake service is required.

Admissibility runs first. The signature is verified against the identity primitive recorded in the counterparty identity record (114) of the asserting party. Where the agent holds no such record, it is verified against a provisional identity primitive constituted from the presented identity material, and the artifact is held in the pre-settlement inert state until a counterparty identity record is promoted upon a matched-pair settlement. Second, the attested assertion-cost counter state is checked: an artifact whose attested state is absent, whose attested epoch identifier is not a valid successor of the epoch recorded for that party, or whose attested counter state falls below a state that party previously attested is appended to the lineage field and not admitted. A non-admitted artifact produces no determination, moves no value of the scoped integrity vector (106), and increments no counter.

An admitted artifact passes to the admission evaluator (120), which produces exactly one determination from a closed set: accepted (122), rejected (124), not-determinable (126), or not-applicable (128). No scalar confidence, no probability, and no graded weight is produced in place of a determination. The determination is produced by the semantic agent over its own lineage field, against the declared value set of the signed policy object in force at the recorded assertion time, and not by an external authority, an arbitrator, a registry, or a scoring service.

The evaluator first runs a value-scope test on each declared value: action-class membership, scope-partition membership, and mapping of the affected-party class to the empathy-scope designation. A declared value is implicated where all three parts are satisfied and not otherwise, and where the descriptor implicates none, the not-applicable determination (128) issues and the procedure terminates. Otherwise the evaluator retrieves lineage entries bearing on the identified conduct. The affected-party

class is not an input to that retrieval, whereby an asserting party is incapable of narrowing the entries retrieved against its own assertion by the affected-party class it declares.

The remaining three classes are tightly bounded. The rejected determination (124) issues only where a retrieved entry affirmatively contradicts the artifact on a recorded field of that entry, an entry silent as to that field being no contradiction. The not-determinable determination (126) issues where the lineage field contains no bearing entry, an entry that does not resolve the descriptor, or an entry incomplete with respect to it. The accepted determination (122) issues where the descriptor implicates at least one declared value and no retrieved entry affirmatively contradicts the artifact. Absence of evidence resolves to not-determinable and to no other class, whereby a semantic agent is incapable of refusing a conduct evaluation artifact by reason of its own record being silent, incomplete, or unavailable.

Each determination is appended to the lineage field with the artifact, the entries retrieved, the class produced, and the ground of the determination. Consequences are conditional. Where the accepted determination issues, a state modifier modifies the scoped integrity vector (106) and the self-esteem aggregate (108), each modification scaled by the entropy-weighted harm coefficient, and a deviation engine recomputes the deviation likelihood (706) from the modified values; where that quantity satisfies the policy-declared bound, the authorization gate (300) transitions to the withheld state (310) for the affected action class and the agent enters the non-executing cognitive mode (302). Where the rejected determination issues, a refusal counter (304) is incremented and writes the gate upon satisfaction of a threshold. Those bounds and thresholds are declared in the signed policy object.

Where the two designs separate

Both designs constrain what an autonomous agent may do, so the convergence is real. The divergence is in what the constraint is computed from, and on whose side of the conversation the input originates.

Agentforce, as publicly described, places its control surface with the enterprise deploying the agent. The disclosed architecture is not a competitor to that surface. Its recited input is an artifact signed by a party other than the principal, and it specifies how that input resolves against the agent's own record and the policy object in force, rather than against the asserting party's account of events.

Next, the shape of the answer. A closed set of four classes, each appended with its ground alongside the artifact and the retrieved entries, is a different object from a score or a routing decision. A thin record does not exit the procedure: it resolves to not-determinable, recorded like any other class, and repetition against one asserted conduct event increments a refusal counter.

Last, what restores authority. A restoration controller returns the authorization gate from the withheld state toward the granting state only upon a procedure appended to the lineage field. No elapse of time, and no payment, transfer, or consideration by any counterparty, returns it.

Coexisting in a support stack

Nothing in the disclosed architecture displaces a CRM-grounded service platform. An enterprise would still want its agent reasoning over real account data, with configured scope, a handoff path to a person, and a trust boundary around customer data. A conduct admission layer sits behind those, on the agent's side.

The two touch at dispatch. The disclosure recites a dispatch-authority predicate recomputed on each dispatch request from state then carried in the memory field, satisfied only upon a three-stamp conjunction of a policy stamp, a lineage stamp, and an authorization stamp, required in full. Satisfaction of fewer than three produces a deterministic denial that is a valid recorded outcome and not an error, appended naming the stamp that did not resolve, and not converted into a determination concerning any party.

Limits, plainly. This architecture resolves the accusation, not the answer. Its determinations are computed over the agent's own lineage field and the signed policy object in force and over nothing else, and its intake presumes an asserting party able to produce a signed artifact carrying the recited fields.

Disclosure Scope

This article describes subject matter disclosed in U.S. Provisional Application No. 64/117,812, which is pending. Statements here about the architecture describe the disclosure as filed, are not claim constructions, and are not assertions of scope, validity, or enforceability. Features described in accordance with an embodiment are described as such in the filing, and quantities described as policy-declared are declared in the signed policy object; nothing here supplies a value for one.

References to Salesforce Agentforce are to public materials and are used for comparison only; no relationship, endorsement, or infringement is asserted.

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)
- The Corrective Encounter: A Commission-Versus-Omission Compliance Test for Whether an Agent Actually Changed (/articles/conduct-admission/corrective-encounter-commission-omission-compliance-test)
- Continuity-Break Demotion of a Counterparty Identity Record: Reversing First-Encounter Promotion Without a Conduct Determination (/articles/conduct-admission/demotion-persistent-tier-continuity-break)
- Ratchet-Freeze on Uncomplied Correction: Why One Conforming Act Cannot Restore an Agent's Earned Authorization Persistence (/articles/conduct-admission/ratchet-freeze-uncomplied-correction)
- Three-Way Corroboration Classification: Corroborating, Contradicting, or Neither, and the Failed-Corroboration Reject Trigger (/articles/conduct-admission/three-way-corroboration-classification-reject-trigger)
- Untested-Class Authorization Decay: Metering Agent Permission Persistence by Recorded Corrective Encounters (/articles/conduct-admission/untested-class-authorization-decay)

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)
- Sierra AI Agents and the Record Behind a Customer Complaint (/articles/conduct-admission/sierra-ai)
- Decagon: Support Agents and Record-Grounded Accountability (/articles/conduct-admission/decagon-ai)
- Intercom Fin: Resolution Rates and the Conduct Record (/articles/conduct-admission/intercom-fin)
- Zendesk AI Agents: Escalation, Audit, and Agent Conduct (/articles/conduct-admission/zendesk-ai-agents)
- **Agentforce and the Accusation an Agent Must Answer (/articles/conduct-admission/salesforce-agentforce-conduct)**

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)