Nick Clark · Published July 29, 2026 · PDF

Zendesk AI Agents: Escalation, Audit, and Agent Conduct

A customer tells a support bot that it promised a refund and then went back on it. That assertion is not about an answer being wrong. It is about the agent's conduct, and it has to land somewhere. Customer service platforms are built to route such an assertion into work: a human queue, a satisfaction signal, a case history. U.S. Provisional Application No. 64/117,812 discloses a different handling path, in which a signed conduct evaluation artifact arrives from a party other than the principal and the agent resolves it against its own append-only record and the signed policy object in force, producing exactly one determination from a closed set of four. This article compares that architecture with Zendesk AI agents, as publicly described.

The complaint that is about the agent itself

Support automation has gotten good at the first-order job: read an inbound message, check an order or a subscription, then resolve the request or hand it to a person. When the answer is wrong, the correction is straightforward in kind. The answer gets fixed and the case gets reopened.

A different kind of message behaves differently. The customer is not disputing an answer but asserting something about what the agent did: that it committed to a credit and then reversed, or that it made a representation outside what it was authorized to make. The subject of the assertion is the agent's conduct.

Conventional handling routes this upward. The conversation escalates, a person reviews the transcript, and the outcome is recorded. That is sensible operational design, and for most consumer support it is the right one.

The filed architecture starts from a different requirement: that the agent whose conduct is asserted be the component producing the determination, that the determination be drawn from a closed set, and that it write the agent's own authority to take further action of the class at issue. That matters most as agents take consequential actions such as issuing credits, changing entitlements, or canceling service, where the question stops being "was the reply good" and becomes "should this agent still be permitted to act in this class."

What Zendesk AI agents are built to do

Zendesk, as publicly described, is a customer service platform organized around ticketing. Conversations arriving by email, chat, or messaging become tracked work items with an owner, a status, and a history. Around that core, the platform is publicly described as offering administrator-configured automation, a help center and knowledge base, and service reporting.

Zendesk AI agents, as publicly described, form the autonomous layer over that model. They engage customers directly, aim to resolve common requests without a person, draw on the organization's own knowledge content, and pass a conversation to a human agent when it calls for one. Administrators are described as configuring their scope.

This is a well-fitted design for its purpose, worth stating plainly before any contrast. Support organizations need containment, human oversight at the boundary, a durable case history, and administrator control over what automation may say and do. Escalation to a person is not a fallback in such an architecture but a deliberate control, and for the large majority of consumer support a qualified person is a better answer than any purely computational determination.

What follows is not an audit of that product and asserts nothing about how it is built internally. It sets out what the filed architecture requires, so a practitioner can compare those requirements against whatever their own deployment already provides.

Resolving the assertion against the agent's own record

In the disclosed embodiment, the architecture operates within or upon a persistent semantic agent (100), an identity-bearing computational object 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) authored by the principal and covered by that principal's signature. The agent also holds a counterparty identity record (114) per counterparty.

A complaint about conduct enters as a conduct evaluation artifact (116) originating from an asserting party (118) other than the principal, 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. It arrives over the agent's ordinary interaction channel, the same channel over which it receives dispatches, responses, and governed observations. The filing states that no dedicated evaluation channel, no out-of-band reporting interface, and no centralized intake service is required.

The artifact is structured rather than narrative. It carries an asserting-party identifier, a recorded assertion time, a conduct descriptor comprising an action-class identifier, a scope-partition identifier, and an affected-party class, a signature of the asserting party (118), and an attested state of that party's assertion-cost counter (400) together with an epoch identifier of that party's dynamic agent hash chain. The descriptor identifies conduct by reference to structural elements recorded in the append-only lineage field (104), and not by natural-language characterization. Per the filing, the architecture infers no state, intent, or affect of the asserting party from the artifact's content, and applies no language model or sentiment classifier to it.

Admissibility runs as an ordered procedure. First the signature is verified against an identity primitive in the counterparty identity record (114). Then the attested counter state is verified, and an artifact whose attested state is absent, whose epoch identifier is not a valid successor of the recorded epoch, or whose counter state is below a state previously attested by that party, 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 an admission evaluator (120), which produces exactly one determination from a closed set: an accepted determination (122), a rejected determination (124), a not-determinable determination (126), or a not-applicable determination (128). No scalar confidence, probability, or graded weight is produced in place of a determination, and the determination is not produced by an external authority, an arbitrator, a registry, or a scoring service. The evaluator first runs a value-scope test against the declared value set of the signed policy object (112) in force at the recorded assertion time. Where the descriptor implicates no declared value, the result is the not-applicable determination (128). Where at least one is implicated, entries bearing on the conduct are retrieved from the lineage field. The rejected determination (124) issues only where a retrieved entry affirmatively contradicts the artifact on a

recorded field, and not where the entry is silent as to that field. Absence of evidence resolves to the not-determinable determination (126) and to no other class, so an agent cannot refuse an artifact by reason of its own record being silent or incomplete.

Consequence attaches to the agent rather than to a case record, and it is conditional at every step. Each determination is appended to the lineage field together with the artifact, the entries retrieved, and the ground of the determination. Where the accepted determination (122) issues, a state modifier modifies the scoped integrity vector (106) and the self-esteem aggregate (108), and a deviation engine thereupon recomputes the deviation likelihood (706) from the modified values. Where that quantity satisfies the policy-declared bound, and only then, 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), in which speculative evaluation continues without committing state changes. An escalation emitter emits an escalation record to the principal upon that transition. A restoration controller returns the gate 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.

Authority is rechecked rather than granted once. A dispatch-authority predicate is recomputed on each dispatch request from state then carried in the memory field (102), and is satisfied only upon a three-stamp conjunction of a policy stamp, a lineage stamp, and an authorization stamp, required in full. Per the filing, it is not computed from a previously issued authorization token, a cached predicate result, or a session grant, so a write of the authorization gate (300) takes effect at the next dispatch request without revocation infrastructure.

Category overlap, architectural distance

Both designs address the same question, what happens when an agent's conduct is contested, at different layers.

A service platform addresses it organizationally. The complaint becomes work, a person takes ownership, the resolution is recorded, and administrators adjust configuration if a pattern emerges. That layer carries the customer relationship and the commercial remedy, neither of which the filed architecture touches.

The filed architecture addresses it computationally, inside the agent's own carried state. Three properties are load-bearing requirements of that architecture rather than options within it: determinations are produced by the accused agent over its own append-only record, the output set is closed with no confidence score in place of a determination, and restoration of withheld authority is barred both to the elapse of time and to consideration from any counterparty.

Positioning is therefore complementary rather than competitive. Ticketing answers what the organization did about a customer's problem. Record-grounded conduct admission answers whether a given agent remains authorized to act in a given class.

Coexistence, and what the filing does not address

In a realistic deployment the two layers sit in series. The customer-facing surface, the transcript, the human handoff, and the reporting stay where they are, and a conduct layer sits underneath, governing which actions the agent may dispatch.

Getting there imposes preconditions a practitioner should price before committing. Because conduct descriptors are structured, the action taxonomy has to be declared before any complaint can be resolved against it. Because artifacts must be signed and must carry an attested assertion-cost counter state and a valid successor epoch identifier, the asserting party must hold identity material resolvable to a counterparty identity record (114) and must maintain a dynamic agent hash chain.

The limits deserve equally plain statement. The architecture does not decide whether an assertion is true in the world. It resolves the assertion against the agent's own record, and a silent record yields the not-determinable determination (126) rather than a finding either way. An unstructured, free-text complaint falls outside its intake by design. Nor does it supply the declared value set, the scope partitions, or the bounds that gate consequence; those are authored by the principal in the signed policy object (112), and the filing describes several quantities as policy-declared without stating values. And the withheld state (310) is a restriction on the agent, not a remedy to the customer: it stops action of an affected class and notifies the principal, while any commercial resolution remains a matter for the service organization.

So the two are not substitutes. One governs the case. The other governs the agent.

Disclosure Scope

The architecture described here is disclosed in U.S. Provisional Application No. 64/117,812, a pending application. Statements about it are drawn from that filing and use its own mechanism names and outcome words. Nothing here is a claim construction or a legal opinion.

References to Zendesk AI agents are to public materials and are used for comparison only; no relationship, endorsement, or infringement is asserted. Product descriptions reflect publicly documented purpose and category at a qualitative level, may not reflect current capabilities, and say nothing about internal implementation. Readers should consult that vendor's current documentation.

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)