

ServiceNow AI Agents: When the Correct Outcome Is No Action

An enterprise agent reaches a step where the record it needs is silent, the policy in force cannot be resolved, or the counterparty whose acceptance the action requires is not recorded in the tenant at all. Something has to come out of that step. A pending provisional application discloses an architecture in which the outcome of such a step is a recorded abstention: an entry naming the unavailable input, held in a type disjoint from magnitude, carried across an agent boundary by attestation, and adverse to no party. This article sets that structure alongside the agent layer ServiceNow publicly describes.

A step with nothing in it

A procurement agent is asked to approve a vendor exception. The approval requires sign-off from a category owner. No category owner is recorded for that spend category, because the person left and the role was never reassigned. The agent reaches the step and there is nothing there.

Several things can come out of that step. The agent can execute anyway, on a default. It can deny, which resolves the matter against the requester for a reason that has nothing to do with the requester. Or it can fail, which monitoring will eventually read as unreliability. Each of those converts a missing input into a consequence for somebody.

The same shape recurs across agentic enterprise work. A record is silent as to whether a thing happened. A policy version cannot be resolved for the time the assertion was made. A peer agent has stopped acting in a class of actions while its counterparties keep dispatching. The honest answer in each case is that the determination cannot be made, and the architectural question is whether that answer has somewhere of its own to go or has to be expressed as a boolean, a null, a retry count, or a health score.

ServiceNow AI Agents, as publicly described

ServiceNow, as publicly described, is an enterprise workflow platform organized around records and around structured processes that move those records through defined states. Its public positioning centers on IT service management and extends to other domains where work arrives as a request and has to be routed, approved, fulfilled, and closed against an auditable record.

ServiceNow AI Agents, as publicly described, extend that pattern to agentic execution. The company describes tooling for building AI agents, for orchestrating multiple agents across a workflow, and for having those agents act on platform data and platform actions rather than in a separate system. The stated design intent is that agents operate inside the same workflow, approval, and record-keeping structures that already govern human work on the platform, so that an agent's activity carries the same routing, approvals, and audit expectations as a human fulfiller's.

That is a sound architectural commitment. An agent acting inside the system of record inherits its accountability, along with the platform's existing controls: role-based access, approval steps, workflow state, and audit history.

Nothing below audits that product or characterizes its internals. What follows describes what a different, separately disclosed architecture requires of itself, so that a reader on the ServiceNow side can compare the two structurally.

Abstention as a first-class outcome, and the bar on converting it

In the filed architecture, no determination stage converts unavailability of a required input into an adverse consequence for any party. Three elements carry that invariant: a degradation map (500), a conversion bar (502), and a non-execution attestation (504).

The degradation map enumerates, for each determination stage, the input required, the abstention outcome produced when that input is unavailable, and the consequence foreclosed. Where the append-only lineage field (104) is silent as to an asserted conduct, the admission evaluator (120) produces the not-determinable determination (126), and the rejected determination (124) is foreclosed, since under the filing a rejection issues only upon an affirmative contradicting entry. Where the signed policy object (112) in force at the recorded assertion time cannot be resolved, no determination is produced at all: the conduct evaluation artifact (116) is appended as pending and a structured inquiry requesting the policy object is generated. Where no acceptance determination is received within the window declared in the signed policy object (112), the outcome is recorded as not-determinable, and the consequence foreclosed is resolution of the matter against the non-responding party.

One exception to the invariant is recited. Where the reason-type of an edge cannot be resolved, the outcome is not an abstention but a modification bounded to the non-zero minimum declared in the signed policy object (112), the consequence foreclosed at that path being a magnitude of zero rather than an adverse magnitude.

Structurally, the conversion bar carries the weight. An abstention outcome is represented in a form disjoint from the domain of values a consuming determination can consume as a magnitude, realized by a type discipline in which an outcome entry and a magnitude are values of disjoint types and no total function maps the former to the latter. A threshold comparison of such an entry emits an outcome of the recorded abstention class and does not emit a Boolean. An accumulation over a set containing one emits an abstention outcome rather than a sum over the remaining members. The filing is explicit that the foreclosure is affirmative, barred by the structure of the values

and not by an absence of defined behavior, and describes an embodiment in which the type discipline is enforced statically, whereby a determination attempting the conversion is not constructible.

Abstention propagates rather than resolving. A determination consuming one emits an outcome of its own recorded class, with no stage replacing, resolving, or defaulting it, so the identity of the originally unavailable input stays recoverable from the append-only lineage field (104) by following the chain. It also crosses the agent boundary. Where an agent has written its authorization gate (300) to the withheld state (310) for an enumerated set of action classes and transitioned into the non-executing cognitive mode (302), it emits a non-execution attestation (504) carrying that enumeration, an abstention-class type marker whose declared type is the abstention type (506), an attested epoch field, and an express designation that the attestation is not a determination and is not a denial of any dispatch.

Upon a satisfied verification, a receiving agent writes the attestation as a carried abstention entry, adopting the type designation as received rather than re-deriving it. Its dispatch-authority predicate then fails for that agent in those classes, and the filing requires that failure be recorded as a positive abstention rather than as a denial: no determination that the dispatch was impermissible, and no fault of either agent. The predicate is unaffected for classes absent from the enumeration and for dispatches to other counterparties. Release is by named condition, including a verified unmarked execution record in the class, a superseding attestation omitting it, or elapse of the time-to-live field, which elapse the filing states is no evidence that the class has been restored.

Convergence first, then the point of divergence

At the category level the two approaches converge. Both hold that agent activity belongs inside a governed record rather than beside it, and both treat auditability as architectural rather than as a logging afterthought. The disclosed architecture is

complementary on that reading: it concerns the layer at which a determination is computed, not the layer at which work is routed and fulfilled.

Divergence sits at the level of what an outcome is permitted to be. Workflow states such as approved, rejected, pending, failed, or cancelled are legible to operators, and a process can be held open in such a state. The disclosed architecture requires something narrower of itself: that a withholding be a value of a type from which no total function reaches a magnitude, so downstream arithmetic on it is unconstructible rather than merely discouraged. That is a requirement on the type discipline a determination runs in, a different question from the states a workflow exposes, and not one public materials settle either way.

A second point of divergence is directional. Guardrail and orchestration layers are generally described in terms of whether an agent should be permitted to act. The disclosed architecture governs what is recorded when it does not act, and forbids the recorded non-action from feeding an availability score, a health quantity, a suspicion level, or a failure rate of the receiving agent. Admission of an attestation increments no refusal counter (304) and no counter maintained per counterparty and consumed adversely to it, and appends no attribute recording fault, breach, unreliability, degraded standing, or diminished trust to the disclosing agent's counterparty identity record (114). Subsequent non-response in a withheld class is likewise not a determination, however many such non-responses accumulate.

Third, the filing recites a third outcome where a determination requires an admissible counterparty class of which no member is recorded: neither execution of the action nor denial of it. The agent transitions into the non-executing cognitive mode (302) for that determination within that scope partition, emits a structured inquiry naming the absent class and the delegation alternatives that would satisfy it, and contracts the partition capability envelope through coupled fields until the absence is cured by instantiation of an admissible counterparty identity record (114).

Layering the two, and the limits of the disclosure

Composition is the natural reading. A workflow platform remains the system of record and the place where human approvals, routing, and fulfillment live. The disclosed architecture sits at the determination layer, deciding what a step emits when its input is not there and producing an entry a surrounding system can display: the abstention class, the stage that produced it, the unavailable input, and a recorded time. An operator looking at a stalled request sees a named missing input and knows which record or role would cure it. Across agents, the abstention propagation chain (508) traces a downstream abstention back to the action class originally withheld.

Several limits are worth stating plainly. The disclosure does not address whether an agent's substantive answer is correct; it governs what happens when the inputs to that answer are unavailable. It supplies no workflow, routing, fulfillment, or user interface. It does not cure the missing input: an abstention names the gap, and someone still has to instantiate the counterparty identity record (114), publish the policy object, or answer the inquiry. Recording withholdings as first-class outcomes also keeps them visible rather than collapsed into a completed or failed state.

Quantities live in policy rather than in the architecture. Response windows, the emission window, the time-to-live field, the closed enumeration of abstention classes, and the non-zero minimum on the reason-type-unresolved path are each declared in the signed policy object (112). Deploying this means writing that policy, which is governance work rather than configuration.

Disclosure Scope

This article describes subject matter disclosed in U.S. Provisional Application No. 64/117,812, a pending application. Reference numerals correspond to elements as recited in that filing. Descriptions here are summaries for practitioners and are not claim language; the filed specification governs. Features are described in accordance

with one or more embodiments. Quantities described as declared in the signed policy object carry no value in the filing, and outcomes gated on a declared bound or window occur only where that bound or window is satisfied.

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

Positive Abstention and the Conversion Bar (</positive-abstention>)

[All 49 steps → \(/inventive-steps\)](#)

Refusing to act is a first-class, recorded outcome — and can't be laundered into acting.

Provisional application

PRIMARY TECHNICAL DISCLOSURE

- [Positive Abstention: Withholding as a First-Class, Non-Convertible Outcome \(/articles/positive-abstention/withholding-as-a-first-class-outcome\)](/articles/positive-abstention/withholding-as-a-first-class-outcome)

SECONDARY TECHNICAL

- [Relay Non-Execution Attestation With Depth and Cycle Bounds \(/articles/positive-abstention/relay-non-execution-attestation\)](/articles/positive-abstention/relay-non-execution-attestation)
- [The Anti-Amplification Bound on Propagated Determinations \(/articles/positive-abstention/anti-amplification-magnitude-bound\)](/articles/positive-abstention/anti-amplification-magnitude-bound)
- [The Attestation Revocation Object \(/articles/positive-abstention/attestation-revocation-object\)](/articles/positive-abstention/attestation-revocation-object)
- [Empty-Intersection Mutual Positive Abstention \(/articles/positive-abstention/empty-intersection-mutual-abstention\)](/articles/positive-abstention/empty-intersection-mutual-abstention)
- [Curing Counterparty Absence by Introduction Protocol \(/articles/positive-abstention/introduction-protocol-cure\)](/articles/positive-abstention/introduction-protocol-cure)
- [Recorded Counterfactual Coupling Test: Detecting When a Counterparty's Silence Moves an Agent's Own Authorization Gate \(/articles/positive-abstention/recorded-counterfactual-coupling-test\)](/articles/positive-abstention/recorded-counterfactual-coupling-test)
- [Self-Execution Intersection Conjunct: Gating Cross-Agent Determination Propagation on the Admitting Agent's Own Per-Partition Execution Record \(/articles/positive-abstention/propagation-admissibility-gate-self-execution-intersection\)](/articles/positive-abstention/propagation-admissibility-gate-self-execution-intersection)

- [Attack-Conditional Cross-Boundary Suspension of Determination Propagation Between Persistent Semantic Agents \(/articles/positive-abstention/propagation-admissibility-gate-attack-conditional-cross\)](/articles/positive-abstention/propagation-admissibility-gate-attack-conditional-cross).
- [Propagation Depth Ceilings for Cross-Agent Determination Records: A Monotonic Anti-Restore-Depth Gate Against Hop-Count Replenishment \(/articles/positive-abstention/propagation-depth-ceiling-anti-restore-depth\)](/articles/positive-abstention/propagation-depth-ceiling-anti-restore-depth).
- [Attestation Staleness Outcome and the Attestation Window Parameter: Bounding the Age of a Carried Withholding Without Recording a Rejection \(/articles/positive-abstention/attestation-staleness-outcome-attestation-window-parameter\)](/articles/positive-abstention/attestation-staleness-outcome-attestation-window-parameter)
- [Evidentiary-Basis Omission for the Receiving Counterparty: Attesting a Withholding Without Naming the Recipient as Its Occasion \(/articles/positive-abstention/evidentiary-basis-omission-receiving-counterparty\)](/articles/positive-abstention/evidentiary-basis-omission-receiving-counterparty)
- [Persistence-Tier Emission Targeting for Non-Execution Attestations: Bounding the Audience of an Agent's Recorded Withholding \(/articles/positive-abstention/non-execution-attestation-emission-targeting-persistence\)](/articles/positive-abstention/non-execution-attestation-emission-targeting-persistence)
- [Restoration Settlement Reference: Releasing a Carried Abstention Entry on a Matched-Pair Settlement Record of Gate Restoration \(/articles/positive-abstention/restoration-settlement-reference-distinct-release-condition\)](/articles/positive-abstention/restoration-settlement-reference-distinct-release-condition)

APPLICATIONS • GENERAL

- [Safety-Critical Autonomy: Recording a Refusal to Act as a First-Class Outcome \(/articles/positive-abstention/safety-critical-autonomy-abstention\)](/articles/positive-abstention/safety-critical-autonomy-abstention).
- [Clinical Decision Support: When the Correct Output Is No Output \(/articles/positive-abstention/clinical-decision-support-abstention\)](/articles/positive-abstention/clinical-decision-support-abstention).

APPLICATIONS • SPECIFIC

- [NIST AI Risk Management Framework: Where Abstention Fits in Agent-to-Agent Conduct \(/articles/positive-abstention/nist-ai-rmf-abstention\)](/articles/positive-abstention/nist-ai-rmf-abstention).
- [ServiceNow AI Agents: When the Correct Outcome Is No Action \(/articles/positive-abstention/servicenow-ai-agents\)](/articles/positive-abstention/servicenow-ai-agents)
- [Copilot Studio: Recording a Withholding as a Real Outcome \(/articles/positive-abstention/copilot-studio-abstention\)](/articles/positive-abstention/copilot-studio-abstention).
- [Bedrock AgentCore: Session Isolation and Governed Abstention \(/articles/positive-abstention/bedrock-agentcore-abstention\)](/articles/positive-abstention/bedrock-agentcore-abstention)

TERMINOLOGY

- [abstention outcome \(/glossary#abstention-outcome\)](/glossary#abstention-outcome)

- [conversion bar \(/glossary#conversion-bar\)](/glossary#conversion-bar)
- [non-execution attestation \(/glossary#non-execution-attestation\)](/glossary#non-execution-attestation)
- [positive abstention \(/glossary#positive-abstention\)](/glossary#positive-abstention)

[Positive Abstention and the Conversion Bar overview → \(/positive-abstention\)](/positive-abstention)