Nick Clark · Published July 24, 2026 · PDF

NIST AI Risk Management Framework: Where Abstention Fits in Agent-to-Agent Conduct

An agent in one organization stops performing a class of actions, and its counterparty records the gap as failure: timeouts, a falling health metric, a vendor marked unreliable. Deliberate withholding and a crashed process arrive looking identical, and the agent that announces a withholding is charged what the agent that goes dark is charged. The NIST AI Risk Management Framework, as publicly described, gives organizations a shared structure for governing, characterizing, measuring, and managing AI risk, and it is deliberately neutral about implementation. U.S. Provisional Application No. 64/117,812 works at the layer that neutrality leaves open, disclosing a degradation map, a conversion bar that forecloses turning an abstention into a magnitude or a Boolean, and a credentialed non-execution attestation that, upon verification by the receiving agent, carries a withholding between agents without it becoming a fault of either.

When a Counterparty Goes Quiet, the Record Calls It a Fault

An agent operated by one organization dispatches work to an agent operated by another. At some point the second stops performing a class of those actions, and nothing announces it. The first agent's instrumentation records what instrumentation

records: requests that did not complete, a climbing error rate, a vendor health indicator below whatever line someone drew.

Later, when risk documentation for the deployment is assembled, that period appears as counterparty unreliability. It may not have been. The second agent may have withheld deliberately, under its own governance. The record cannot tell the two apart, because the data path never carried a value meaning "declined" that was distinguishable from a value meaning "broke."

Two problems follow. The record is wrong about conduct, which is what risk documentation exists to capture. And the incentive runs backward: the agent that announces a withholding watches its counterparty's health metric fall and its non-responses accumulate into a number that later reads as fault, while the agent that goes dark is charged the same.

Underneath sits a representation problem. Where a pipeline carries no type for absence, "no answer" tends to become a number on its way downstream: a held sample, a default constant, an operand in a threshold comparison that yields a Boolean. From there the pipeline computes on a value never observed.

The event is bilateral while the accounting is not. Each organization keeps a register scoped to its own systems, describing the other party from outside. Neither is the record of what passed between them.

What the Framework Sets Out to Do, in Its Own Terms

The NIST AI Risk Management Framework is, as publicly described, a voluntary framework published by the U.S. National Institute of Standards and Technology to help organizations that design, develop, deploy, or use AI systems identify and manage the risks those systems carry. Its published materials present it as intended for use across sectors and across the AI lifecycle.

Its published description organizes the work into a small set of core functions, commonly named Govern, Map, Measure, and Manage. Alongside these it sets out characteristics associated with trustworthy AI, including validity and reliability, safety, security, accountability and transparency, explainability, privacy, and the management of harmful bias.

The framework is deliberately non-prescriptive. As documented publicly, it describes outcomes and practices rather than mandating a particular implementation, technology, or control set, and it anticipates profiles that tailor it to a given use case, sector, or technology type. That neutrality is a design decision and a considerable strength: one framework serves many sectors at once, and it gives engineering and legal teams shared vocabulary.

Being exact here matters, because the point is not a shortcoming. A framework that told implementers which data type to use for a missing input would be a specification, not a risk management framework. It asks an organization to account for how its AI systems behave and to document the decisions shaping that behavior. It does not dictate how a runtime represents the absence of an input, or what one organization's agent should write into its records about another's.

The Filed Construct: an Outcome That Cannot Become a Number

Chapter 5 of U.S. Provisional Application No. 64/117,812, "Adverse-to-No-Party Abstention and the Conversion Bar," addresses that representation problem. Its named elements are a degradation map (500), a conversion bar (502), a non-execution attestation (504), an abstention type (506), and an abstention propagation chain (508).

The invariant comes first. Where an input required by a determination is unavailable, incomplete, or unresolvable, the determination emits an outcome entry of a recorded abstention class identifying the unavailable input, and emits no outcome adverse to the semantic agent (100), to an asserting party (118), or to a counterparty. It holds at every

stage requiring an input, whatever the cause. One exception is recited: at the reason-type-unresolved path, an unresolvable reason-type instead produces a modification bounded to the non-zero minimum declared in the signed policy object (112), foreclosing a magnitude of zero. The degradation map enumerates, for each determination stage, the input required, the abstention outcome produced upon unavailability, and the consequence foreclosed; at its non-response path, where no acceptance determination arrives within the window declared in that policy object, what is foreclosed is resolution of the matter against the non-responding party.

Enforcement falls to the conversion bar (502), which forecloses a consuming determination from converting an abstention outcome into a scalar value, a default value, an operand of a threshold comparison (510), or a consequence adverse to any party. The foreclosure is affirmative rather than an absence of defined behavior: an outcome entry and a magnitude are values of disjoint types, and no total function maps the former to the latter. A threshold comparison over an outcome entry emits an outcome of the recorded abstention class and not a Boolean; an accumulation over a set containing one emits an abstention outcome and not a sum. In an embodiment the type discipline is enforced statically upon the agent's instructions before they execute, so a determination attempting the conversion is not constructible.

Across the organizational boundary, one construct does the work. An agent that has written its authorization gate (300) to the withheld state (310) for an enumerated set of action classes, and has transitioned into the non-executing cognitive mode (302), emits a non-execution attestation. Its payload carries the withheld action-class enumeration, an enumerated evidentiary basis, an attested epoch drawn from the agent's own hash chain rather than a host clock, an abstention-class type marker carrying a type designation and not a magnitude, and an express non-determination designation stating that it is neither a determination nor a denial of any dispatch.

On receipt, and upon verifying the authority credential and continuity hash, the attested epoch as a valid successor, and the marker as resolving within the closed enumeration declared in its own signed policy object (112), the receiving agent writes a carried abstention entry as a value of the abstention type (506), that designation adopted as received rather than re-derived or defaulted. Four negative limitations then hold: no conversion to a scalar, a default, or a threshold operand, so no availability score, health quantity, suspicion level, or failure rate takes the entry as an input; no counter of the disclosing agent is incremented; nothing adverse is appended to its counterparty identity record (114); and subsequent non-response within a withheld class is recorded as a not-determinable outcome naming that class as the unavailable input, however often it recurs. The receiving agent's dispatch-authority predicate thereafter fails for those classes toward that party alone, and stays failed until the entry is released; that failure is recorded as a positive abstention rather than a denial, with no fault of either agent.

Different Layers of the Same Concern

The two converge at the category level: each is addressed to the legibility of an AI system's behavior in the record a later reader consults.

The divergence is architectural, along two axes. The first is layer. The framework, as publicly described, states outcomes an organization should achieve and leaves the means to implementers. The filing runs the other way, fixing a representation and foreclosing a conversion at the level of types, to the point that in an embodiment the offending code is not constructible. One is a management posture that survives any technology stack; the other lives inside a particular runtime, is not a framework, and says nothing about accountability structures or which risks matter.

Unit of account is the second axis. Risk management frameworks are organized around an organization accounting for the systems it develops, deploys, or uses, third-party relationships included. The filing's unit is narrower: a bilateral record held between two

agents that may sit in different organizations entirely, where the object crossing between them carries an abstention's type rather than a status code or a health signal. Choosing that representation is the kind of implementation decision a non-prescriptive framework deliberately leaves open, which is why the two sit alongside each other rather than compete.

A third difference looks like a conflict and is not. The conversion bar forecloses turning an abstention outcome into a scalar, a default, or an operand of a threshold comparison, yet measurement is not thereby forbidden. A further disclosed embodiment has the agent maintain, in the counterparty identity record (114), an abstention-attribution count: abstention outcomes produced within a declared interval at stages whose unavailable input was a response, a record, or a receipt required of that counterparty. The count is recited as a measure and not an adverse fact, incrementing no refusal counter (304), causing no write of the authorization gate (300), and not admitted as an operand of a threshold comparison producing a consequence adverse to the counterparty. Where it exceeds a bound declared in the signed policy object (112), the agent emits a structured inquiry to its principal naming the counterparty and the persistently unavailable input. Counting is permitted; charging the counterparty with the count is not.

Running Both in One Deployment

An organization crosswalking its AI governance to the framework need change none of that work to adopt what Chapter 5 discloses. Governance structure, characterization of intended use and affected parties, monitoring plans, and accountability assignments remain the organization's.

What changes is what those activities read from. The degradation map puts a system's behavior on a missing input into one enumeration rather than scattering it across constants and fallback branches. Where a counterparty relationship is in scope, the

non-execution attestation is something an organization can emit and retain: a credentialed record that a set of action classes stands withheld, expressly designated neither a determination nor a denial of any dispatch.

Several behaviors are conditioned. Emission goes to counterparties recorded as having dispatched within an emission window declared in the signed policy object (112), and to no other party. Admission requires the abstention-class marker to resolve within the closed enumeration declared in the receiving agent's policy object. A carried entry is released as to a class upon a verified unmarked execution record in that class, upon a superseding attestation omitting the class and carrying a successor attested epoch, or upon elapse of the time-to-live field, and that elapse is expressly no evidence the class has been restored.

The architecture's limits deserve plain statement. It does not make an unavailable input available. Nor does it assess whether an AI system is fit for its purpose, measure harm, characterize context, or assign accountability, which are the framework's concerns and not a type discipline's. It establishes conformance with nothing: the filing is pending, and whether a lineage record satisfies any given program is for the parties responsible under it.

Placing the abstention boundary is design work. Propagation is transitive by construction, a determination consuming an abstention outcome emitting one of its own recorded class with no stage replacing, resolving, or defaulting it, so a multi-stage pipeline terminates in an abstention outcome and the unavailable input stays recoverable by following the chain. Where that leaves a determination without an admissible counterparty class, the disclosed responses are the third outcome, a structured inquiry to the principal, and contraction of the partition capability envelope.

The cross-boundary half presupposes infrastructure rather than a message format: authority credentials, continuity histories, counterparty identity records, and a signed policy object declaring the abstention classes. An organization whose counterparties

run none of this gets the within-agent discipline and not the bilateral record. A further disclosed embodiment bounds the obvious abuse, declining to write a further carried entry where entries held from one origin-equivalence class (200) exceed a bound declared in the policy object.

Disclosure Scope

The mechanisms described here are disclosed in U.S. Provisional Application No. 64/117,812, Chapter 5, "Adverse-to-No-Party Abstention and the Conversion Bar," together with further embodiments in that application.

Disclosed: the adverse-to-no-party invariant and its single reason-type-unresolved exception; the degradation map associating each determination stage with a required input, an abstention outcome, and a foreclosed consequence; the abstention outcome as a first-class lineage entry propagating without replacement, resolution, or defaulting; the conversion bar as an affirmative type-level foreclosure, including static enforcement rendering a converting determination not constructible; the non-execution attestation and its payload; admission of a carried abstention entry as a value of the abstention type, adopted as received; the four cross-boundary negative limitations; the dispatch-authority predicate failing as a positive abstention, with its release conditions; the third outcome upon absence of an admissible counterparty class; and, among further embodiments, the abstention-attribution count and the origin-equivalence gate on attestations.

Disclaimed: any claim to risk management practice at large, to AI governance frameworks as such, to documentation, measurement, or monitoring methodologies as such, or to any particular standard, profile, or crosswalk. References to the NIST AI Risk Management Framework are to public materials and are used for comparison only; no relationship, endorsement, or infringement is asserted. This article applies a disclosed architecture to a governance context, asserts no infringement by any party, and states no licensing requirement; the referenced filing is a pending application.

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)

SECONDARY TECHNICAL

- Relay Non-Execution Attestation With Depth and Cycle Bounds (/articles/positive-abstention/relay-non-execution-attestation)
- The Anti-Amplification Bound on Propagated Determinations (/articles/positive-abstention/anti-amplification-magnitude-bound)
- The Attestation Revocation Object (/articles/positive-abstention/attestation-revocation-object)
- Empty-Intersection Mutual Positive Abstention (/articles/positive-abstention/empty-intersection-mutual-abstention)
- Curing Counterparty Absence by Introduction Protocol (/articles/positive-abstention/introduction-protocol-cure)

APPLICATIONS · GENERAL

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

TERMINOLOGY

- abstention outcome (/glossary#abstention-outcome)

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

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