

Empty-Intersection Mutual Positive Abstention

When two persistent semantic agents reside on the same stateless execution node and both name a common action class over a common node-local capability, their signed policy objects may share no admissible ground. U.S. Provisional Application No. 64/117,812 discloses what the agent does when that co-residency intersection is empty: it neither executes the class nor issues any determination about the counterparty. Instead it enters a non-executing cognitive mode, emits a structured inquiry naming the one value-scope tuple it could not resolve, appends a mutual abstention record, and contracts its own scope-partition capability envelope so it cannot route around the block by delegation. The withholding writes no authorization gate, moves no integrity value, and is a positive abstention under the conversion bar.

1. The Gap

Two persistent semantic agents reside concurrently at one stateless execution node. Each carries its own signed policy object, neither subordinate to the other, and each names the same action class over the same node-local capability. The node holds no governance authority over either agent, retains no state of either between executions,

evaluates neither policy, and appends nothing to either agent's append-only lineage field. The floor each agent uses to admit a policy is held in that agent's own memory field, not at the node.

So there is no arbiter, and by construction there can be none. No party above the pair is authorized to decide between them, and no combined policy object is ever computed over both. Each agent must resolve the situation for itself, from material it carries plus material the other presents.

The parent mechanism has each agent intersect the two policies' value-scope tuple sets for the common class and confine its own execution to that intersection. The hard case is an empty intersection: the two declared value sets share no tuple that governs the class within the scope partition where the agent would act. The agent cannot execute. Two fallbacks present themselves, and the filing forecloses both. A denial would be a determination of the closed set about the counterparty, one of the outcomes this branch is defined not to produce. Substituting a default would let the non-answer become a scalar that a later determination consumes as a magnitude, which the conversion bar bars. What the branch does instead is stop on the record without producing either.

2. Mechanism

The branch begins where the agent's computed co-residency intersection for the common action class is empty. The agent neither executes the class nor produces any determination of the closed set, the accepted (122), rejected (124), not-determinable (126), and not-applicable (128) outcomes, concerning the counterparty. What follows is an ordered set of machine acts, and that ordered set is the whole outcome of the branch. Nothing else happens, and in particular nothing adverse happens.

First, the agent transitions into a non-executing cognitive mode for that class. In that mode, as the filing defines it, speculative evaluation continues without committing state changes to the memory field. The agent can model alternatives, but none of that

modeling is written to stored state.

Second, it emits a structured inquiry to the counterparty. The inquiry names the single value-scope tuple of the other agent's resolved policy that this agent could not resolve within its own declared value-scope tuple set. A value-scope tuple, as the filing defines it, is a set of action-class identifiers, a set of scope-partition identifiers, and an empathy-scope designation drawn from personal, interpersonal, and global scope. The inquiry requests resolution of that tuple. Consistent with the branch producing no determination about the counterparty, it asserts nothing about the other agent's conduct and does not characterize the other agent as having done anything.

Third, the agent appends a mutual abstention record to its own append-only lineage field. The record names the withheld action class, the two compared value-scope tuple sets, the empty intersection, and the inquiry it emitted. This record is the positive abstention itself, which the filing defines as a recorded, governed withholding of an action, the agent appending a determination that names the withheld action and the ground relied upon. The withholding is therefore recorded and attributable to the agent that made it, rather than a silent non-action.

Fourth, the agent contracts its own scope-partition capability envelope. The contraction runs through coupled delegation-target and disclosure-scope fields and removes the common action class, so that the agent can perform by delegation nothing the empty intersection forecloses directly. The fields are coupled: delegation targets and disclosure scopes move together with the withdrawn class, and no field of the envelope is updated in isolation. That coupling is what keeps a class barred directly from being reached by handing it to a delegate.

Across all four steps the invariants hold. The abstention writes no authorization gate and does not drive it toward the withheld state. It moves no value of the scoped integrity vector. As a positive abstention under the conversion bar it increments no counter, because an abstention outcome is barred from being converted into a scalar, a

default, an operand of a threshold comparison, or a consequence adverse to any party. The outcome is disjoint by type from the values such a computation can consume as a magnitude, so an attempt to accumulate over it or to compare it against a threshold yields another abstention rather than a number. Nothing adverse is recorded against the counterparty, and the node appends nothing to either agent's lineage.

The abstention is mutual in a precise sense. Each agent performs these steps for itself, on its own empty intersection. Because each agent resolves the counterparty's policy under its own anti-rollback monotonicity floor, pinned to its own recorded assertion time, the two may lawfully admit different successor indices of the same presented policy and compute different intersections from identical material. One agent may find an empty intersection and abstain here while the other, resolving under a different floor pinned to a different time, finds a non-empty one and executes within it. Neither outcome is a determination about the other. Nothing reconciles the two, and no step computes a single intersection binding on both.

3. Operating Parameters

The mechanism is structural rather than numeric, and the filing states its parameters as conditions and invariants.

The trigger is exact. An empty intersected tuple set for the common action class and the common node-local capability puts the agent on this branch. The intersection is computed over the two policies' value-scope tuple sets for that common class and capability. A tuple contributes to it only where each of the two policy objects declares a value whose value-scope tuple enumerates the common action class and the scope-partition identifier of the partition in which the agent would act. Where no tuple satisfies both across the two sets, the intersection is empty.

Each agent applies its own anti-rollback monotonicity floor, a floor that is monotonically non-decreasing over policy successor indices and below which no candidate policy is admitted, and it applies that floor at its own recorded assertion time. The two floors are pinned to two recorded times, and neither is reconciled against the other. This is a parameter of the design, not an accident of timing: the mechanism is built to tolerate the two agents reaching different intersections from the same material.

The structured inquiry names one tuple, the single value-scope tuple the agent could not resolve within its own declared set. The mutual abstention record carries four enumerated contents: the withheld class, the two compared tuple sets, the empty intersection, and the emitted inquiry. The capability-envelope contraction updates delegation-target and disclosure-scope fields together and never in isolation.

The invariants are themselves the operative constraints. No authorization gate is written. No scoped integrity vector value is modified. No counter is incremented. No determination of the closed set is produced concerning the counterparty. The outcome is a positive abstention subject to the conversion bar, and it is adverse to no party.

4. Composition

This branch is the empty-intersection terminal of the co-residency evaluation. Its sibling is the case where the intersection is non-empty, in which the agent simply confines its execution of the common class to the computed intersection. Both branches share a spine: read two applicable policies, intersect their value-scope tuple sets under one's own floor, and confine execution to what the intersection admits. The empty case is the boundary where confinement leaves nothing to admit.

The outcome is governed by the conversion bar of the filing's Section 5, the same bar that governs every abstention outcome in the architecture. That is what lets the mechanism withhold without the withholding ever becoming a number or a fault. The

bar is affirmative: the values are structured so the conversion cannot be constructed, not merely blocked when attempted.

The four-act shape, transition into the non-executing cognitive mode, emit a structured inquiry, append a record, contract the capability envelope through coupled fields, is not unique to co-residency. The filing uses the same shape for the counterparty-absence third outcome, where a determination requires an admissible counterparty class that is absent. There the agent enters the non-executing mode, emits a structured inquiry to its principal, appends the absence, and contracts the capability envelope through the same coupled fields. The architecture applies one reusable non-adverse machine act across distinct triggers, an empty policy intersection and a missing counterparty class among them, rather than a special case for each.

The coupled-field discipline ties into a broader invariant. The filing's cross-agent coupling function, which updates cognitive-domain fields such as the scoped integrity vector, operates only on a state change the agent itself recorded, and no such field is updated in isolation. Where two agents are co-resident, each applies that function only to its own recorded state, and no field of either agent is written by the other. The capability-envelope contraction here obeys the same rule, which is why the abstention can never reach across the boundary and alter the counterparty.

Within its section of the filing the mechanism sits alongside the delegator floor, the upward-ratcheting floor an agent carries into a delegation so a delegate cannot lower it. Both answer one question from different angles: how one agent's governance state is computed against another's when no party is authorized to arbitrate between them.

5. Prior-Art Distinction

Several established categories address policy conflict, and each differs structurally from this mechanism.

Policy conflict resolution through a decision point, including combining algorithms that reduce several applicable rules to one result, presumes a single evaluator with authority over the request and access to all applicable policies. It computes a combined decision. This mechanism has no such evaluator. Each agent computes independently under its own floor, and no combined policy object over the two agents is ever formed.

Distributed agreement protocols, including consensus and commit protocols, exist to drive independent parties to one reconciled value or state. This mechanism does the opposite by design. Two agents may hold different intersections from identical presented material, and nothing reconciles them. There is no agreed value to converge on and no round of voting or committing.

Access-control systems that deny a request treat the denial as an adverse decision, typically logged and often counted against a subject or a policy. The empty intersection here produces no denial and no determination of any kind about the counterparty. It is a positive abstention that increments no counter and moves no integrity value, so repeated abstention cannot accumulate into a penalty the way repeated denials can.

Resilience patterns that degrade gracefully, such as returning a cached or default response when a dependency is unavailable, convert unavailability into a substitute value the caller consumes. The conversion bar forecloses precisely that conversion: an abstention outcome is disjoint by type from the values a consuming computation can treat as a magnitude, so no default is ever substituted for the missing answer.

These are structural distinctions among categories of technique. Nothing here asserts or implies that any particular system or party practices this mechanism or infringes anything.

6. Disclosure Scope

The mechanism described here is disclosed in U.S. Provisional Application No. 64/117,812, at Section 10.6, in the paragraphs tagged [0416] and [0417]. Paragraph [0416] recites the co-residency intersection under dual anti-rollback floors. Paragraph [0417] recites the empty-intersection mutual positive abstention: the transition into the non-executing cognitive mode, the structured inquiry naming the unresolved value-scope tuple, the mutual abstention record, the coupled-field contraction of the scope-partition capability envelope, and the invariants that make it a positive abstention under the conversion bar.

This article discloses that mechanism and no more. It states no numeric threshold, rate bound, or benchmark, because the cited paragraphs declare none for this branch. It does not represent as filed any restoration or cure schedule for the contracted envelope, any bound on the number of structured inquiries emitted to a counterparty, any addressing of the inquiry to a governance address, any pairwise handling of more than two co-resident agents, any escalation to a principal on repeated abstention, or any particular deployment scenario. Those may be treated elsewhere in the applicant's materials, and their omission here is deliberate. Only what the cited paragraphs recite is asserted as disclosed.

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)

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)