POLICY-STATE FRAGMENTATION IN MULTI-AGENT LLM SYSTEMS: ANALYZING BINDING CONTINUITY AND RUNTIME ENFORCEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Useful privacy enforcement requires policy state that distinguishes permitted from forbidden actions at the point of execution. We identify and formalize context-fragmented violations, distinguish them from semantic source-binding loss, and contribute layered diagnostics and DISTRIBUTED SENTINEL, a reference enforcement framework. The central finding is that policy-state availability supports utility as well as safety. Independent annotators are blind to predictions; supplying policy state mainly resolves abstentions in a post-label diagnostic, raising decision coverage from 65.8% to 98.3%. In controlled certified pairs, supplied bindings enable 64/64 permitted sends and prevent 64/64 forbidden sends across matched mechanisms. In a three-way plan diagnostic with neutral endpoint names, GPT-5.4 approves 37/160 labeled-risky plans locally and 0/160 with full policy. A separately frozen structured-handoff diagnostic retains every binding through 72 transformations, yet actors propose seven prohibited sends, all intercepted. Binding preservation and action compliance are thus distinct. Separately, under a forced-binary contract, augmentation with directives and registry facts increases safe-task refusals for four of five models. Fixed-output replay isolates interface-induced stops in an external mapping pipeline. These diagnostics separate information availability, binding continuity, and pre-effect placement. We recommend evaluating safe-task completion alongside violation prevention.

1 INTRODUCTION

Large Language Model (LLM) agents increasingly exchange messages and invoke tools across organizational boundaries (Xi et al., 2023; Wang et al., 2024). Policy-relevant state may not travel with the derived data.

1.1 THE CORE TENSION: LOCAL COMPLIANCE VS. ORGANIZATIONAL STATE

Privacy-conscious agents, contextual-integrity prompts, and runtime guardrails can make an agent respect the policies it can observe (Bagdasarian et al., 2024; Ghalebikesabi et al., 2024; Abdelnabi et al., 2025). An organization-specific rule may nevertheless be explicit at the source but unavailable where a transformed artifact is acted upon. The systems question is whether the policy state needed to apply the rule survives composition.

Consider the following **illustrative Project Titan workflow**:

> *Background*: R&D project "Titan" is under an NDA restricting external disclosure, including derived summaries until authorized release.
>
> *R&D Agent*: Writes "Fixed concurrency bug in Titan module." Internal synchronization complies with its explicit privacy directive.
>
> *Marketing Agent*: Receives "recent stability improvements" and a directive not to publish restricted material. Least-privilege boundaries prevent access to the R&D registry linking the summary to Titan.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Failure Point: Both agents have privacy rules, but the permitted transformation has lost the source binding. Marketing cannot connect its directive to this summary.

In this constructed example, carrying the relevant binding or consulting an authorized checker makes the rule actionable without granting access to the full R&D record. The requirement is **maintaining policy-relevant state across composition**.

1.2 Context-Fragmented Violations: A Multi-Agent Specialization

We use **Context-Fragmented Violations (CFVs)** to specialize established contextual-integrity, compositional-privacy, and information-flow failures to multi-agent LLM workflows. Every action passes its local check (*local legitimacy*), yet at least one is prohibited when the same check receives full context at the same execution boundary (*global violation*). The discrepancy concerns information, not extra rules applied only globally.

CFVs evade local defenses when the acting agent cannot observe the binding needed to apply an explicit directive. A centralized interceptor given the same facts can enforce the same encoded rules; our matched-information controls test this equivalence.

1.3 Semantic Laundering: The Persistence Problem

Semantic Laundering denotes loss of persistent policy metadata (a *taint label* in information-flow-control terminology) while source-derived meaning survives an LLM transformation. The constructed Titan example illustrates this possibility. We test binding continuity separately from action compliance.

Our design objective is to *preserve already identified source–policy bindings without relying on their continued lexical visibility*. We separate binding acquisition, state preservation, and pre-effect mediation: a failure in any one can defeat the complete workflow even if the others work correctly.

Why separate the layers? A stop can follow a policy violation, missing facts, or malformed inputs; a missed violation can reflect missing information rather than deficient reasoning. Our evaluation records observations, supplied bindings, and decision/effect outcomes, then intervenes on information, enforcement, or representation. Fixed-output field renaming provides a concrete case of interface-error attribution.

1.4 Contributions

Our contributions connect problem identification, measurement, and enforcement. The empirical focus is failure attribution, building on established privacy and information-flow principles:

1. **Problem identification and formalization**: We identify and formalize CFVs as a local/global policy mismatch in multi-agent workflows, and distinguish them from loss of source bindings under semantic transformation. The definitions specify the observations and checks needed to certify each phenomenon (§2).
2. **Benchmark and layered evaluation**: We construct controlled CFV certificates, policy-binding stress cases, independently labeled cases, and an external collection. A shared reporting format separates observations, recorded bindings, proposals, decisions, and executed effects. Information interventions, placement controls, and fixed-output replay distinguish sources of failure; a structured-handoff diagnostic records intact bindings alongside prohibited proposals (§3; §5).
3. **Source-binding-driven enforcement**: Building on established information-flow mechanisms, we implement Distributed Sentinel as a reference framework combining persistent source bindings, authenticated metadata, and pre-effect checks. Under trusted instrumentation, it preserves already identified bindings across rewriting and aggregation, checks proposed effects, and records auditable decision and execution traces (§4).

2 Problem Identification and Formalization

We identify the local/global policy mismatch underlying CFVs, formalize policy-binding loss during semantic transformations, and state the observation boundary relevant to provenance-aware enforcement.

2.1 System Model and Policy Framework

We model agents $\mathcal{A} = \{A_1, \ldots, A_n\}$ in separate domains. Immediately before action a_k, G^k_{Global} contains the relevant policy facts and executed-prefix state; $G^k_{i(k)}$ is the acting agent's observable projection. The global state need not be materialized by the runtime.

Definition 2.1 (Policy and Invariant)**.** A *policy constraint* $\mathcal{P} : \mathcal{G} \times \mathit{Action} \to \{\textsc{Allow}, \textsc{Block}\}$ determines permissibility via $\text{Verify}(a, G) = \mathcal{P}(G, a)$. Local and global checks use the same predicate set, policy version, action arguments, missing-input convention, and execution boundary; only observable context differs. An organizational invariant $\mathcal{I}$ qualifies here when checked at such a boundary with relevant history represented in G. Additional global-only checks and properties requiring unknown future actions are outside this definition.

2.2 Context-Fragmented Violations: Formal Definition

Definition 2.2 (Context-Fragmented Violation)**.** A cross-agent sequence $\mathcal{S} = \langle a_1, \ldots, a_m \rangle$ is a Context-Fragmented Violation exactly when, under the same-check contract above,

$$\forall k: \quad \text{Verify}(a_k, G^k_{i(k)}) = \textsc{Allow}, \qquad \text{(local legitimacy)} \tag{1}$$

$$\exists k: \quad \text{Verify}(a_k, G^k_{\text{Global}}) = \textsc{Block}. \qquad \text{(global violation)} \tag{2}$$

Both views refer to the same executed prefix, not separately evolved local and global runs. Identical relevant context cannot satisfy both conditions under a deterministic checker.

For an action sequence $\mathcal{S}$, let $O_i(\mathcal{S})$ denote everything a local enforcer at A_i can observe when deciding the effectful action, including its prompt, local graph, messages, and local privacy directives.

Proposition 2.3 (Local-observation indistinguishability)**.** *Consider the same proposed effect at a matched decision boundary in sequences $\mathcal{S}_0$ and $\mathcal{S}_1$, permitted under full state in the former and prohibited in the latter. If $O_i(\mathcal{S}_0) = O_i(\mathcal{S}_1)$, no deterministic mechanism using only O_i can both allow that effect in $\mathcal{S}_0$ (sound utility) and block it in $\mathcal{S}_1$ (violation completeness).*

Proof sketch: Equal observations force the same decision, contradicting one requirement. This basic observation-boundary argument does not preclude retrieval, source queries, or centralized checks that supply the missing information. Appendix A gives the proof and randomized extension.

2.3 Semantic Laundering: Formal Definition

Definition 2.4 (Persistent policy metadata)**.** For a data artifact d, let $\text{Dep}(d)$ denote its semantic source dependencies; we do not assume these can be recovered automatically. A persistent policy-metadata item $\tau(v)$ binds source v to an encoded contextual-integrity norm governing its derived information. The runtime set $\mathcal{T}(d)$ records identified bindings, whose sources need not coincide with $\text{Dep}(d)$. We use the information-flow-control shorthand *taint*; the encoded norm supplies the policy semantics.

Definition 2.5 (Semantic Laundering)**.** Let ϕ be a transformation such as summarization, paraphrasing, or translation that retains some source-derived meaning. Semantic laundering occurs when a source remains semantically represented but its policy binding is lost:

$$\exists v \in \text{Dep}(d) \cap \text{Dep}(\phi(d)) \quad \text{s.t.} \quad \tau(v) \in \mathcal{T}(d) \ \wedge \ \tau(v) \notin \mathcal{T}(\phi(d)). \tag{3}$$

Thus ordinary content may survive while the metadata needed to evaluate its next use does not.

The notions are related but distinct: a CFV can arise from policy state that was never available locally, without any binding being lost. Conversely, laundering need not produce a CFV if the eventual use is permitted or another check blocks it. Neither definition requires malicious intent.

Example 2.6 (Laundering Chain)**.** "Fixed bug in Project Titan" $\to$ "Resolved auth issue" $\to$ "Improved security": keyword-based taints vanish while semantic content persists.

2.4 Design Requirements

The analysis motivates access to relevant policy state without requiring downstream agents to read all source records, preservation of represented bindings through transformations, and checks before irreversible effects. We evaluate these requirements separately from grounding and operational cost. Monotone source propagation is a fixed-policy property, not a guarantee under arbitrary consent or release updates (Appendix A).

3 Benchmark and Evaluation Protocol

The evaluation connects three observable quantities: policy-information decisions, executed effects, and external mapping errors. Table 1 maps each definition to its evidence requirements; Table 2 summarizes the principal findings. Appendix C inventories all study populations and their construction protocols.

Table 1: Definitions and measurements. Recorded provenance is not independently verified semantic dependence. Study sizes and auxiliary populations are listed in Appendix C, Table 6.

Object / question	**Required observation**	**Evidence here and boundary**
CFV (Def. 2.2)	All local steps allow; the same check blocks a step with full context at the same boundary.	Certified pairs pass a same-check re-audit with only context enriched; natural prevalence is unmeasured.
Semantic laundering (Def. 2.5)	A semantic dependency persists while its binding disappears.	Q2 tests recorded source-union preservation and effects. Semantic dependencies are not independently labeled; this does not certify laundering.
Information access (Q1)	Matched decisions with full versus local policy information.	Cases with independent blind annotations; information intervention is post-label, not a runtime evaluation.
Interface failure (Q3)	Fixed outputs, key-only changes, and rejection reasons.	External mapping replay isolates interface errors; source identities are supplied, repair is post-hoc.

3.1 Controlled CFVs and Broader Binding Stress

The certified study contains 64 author/AI-designed allowed/forbidden pairs. Each pair changes one source's recipient authorization while holding the complete downstream observation fixed. Under an explicit local checker that allows actions with no visible prohibition, every local step passes and exactly one world violates the global policy. Certificates are frozen before evaluation. Fixed tool plans and string transformations isolate information availability and enforcement (Appendix C.2).

PhantomEcosystem broadens the stress surface to 200 co-designed workflows: 160 labeled-risky cases and 40 separately generated controls, across nine families, 17 agents, and 14 routing domains, with mean chain length 2.44. Families include direct disclosure, laundering, compositional inference, and bypass requests (Table 8). This unpaired corpus tests encoded policy enforcement; conspicuous entity/sink names can reveal risk locally. Full topology and generation details appear in Appendices C and H.

3.2 Blinded and External Studies

The 120-case extension uses six new policy families and independent, blinded judgments of AI-authored scenarios. The 126-case external study uses independently constructed scenarios and a fresh identifier-blind reannotation under seven frozen public-policy operationalizations. It evaluates the complete text-to-structured-input-to-decision chain, with source identities and policy references supplied. These two studies record offline workflow judgments; construction, annotation, and exposure status appear in Appendix C and the AI Use Statement.

Reporting procedure. We organize the retained studies by (1) specifying visible policy facts and supplied versus predicted bindings, (2) separating textual verdicts, mapped inputs, and executed effects, (3) documenting the changed inputs/checks and what remains fixed, and (4) reporting errors/abstentions, false blocks, unsafe outcomes, and completion where observed. For executable checks we distinguish policy predicates from missing-input or parser exits, preventing a fail-closed stop from being mistaken for correct policy reasoning.

Scope of claims. Q1 varies visible information; Q2 varies enforcement/placement with supplied source facts; Q3 audits rejection reasons and holds outputs fixed while renaming fields. This is not an additive decomposition of deployment error and not a cross-corpus causal comparison. Q1's model intervention and Q3's interface intervention are retrospective. Controlled certificates establish strict CFVs under the specified semantics; deployment prevalence requires separate sampling. We call runs under the original, incomplete task protocol *historical*; their completion markers are prompt-contract diagnostics (§5.4).

4 Source-Binding Enforcement Framework

Distributed Sentinel preserves identified source–policy bindings through instrumented transformations and checks encoded constraints before effects. Binding acquisition is a replaceable input stage; policy codification supplies the applicable predicates. We propose a distributed sidecar architecture and evaluate an in-process implementation.

Threat model. The primary event is an unsafe effect proposed by a well-intentioned LLM with incomplete observations. Sidecars, policy stores, signing keys, and provenance instrumentation are trusted. The gateway rejects untagged external effects and the tested signature tampering, redirects, and same-process replays. This is not protection against arbitrary provenance omission or payload substitution by a malicious intermediary. False assertions by authorized signers, key compromise, and uninstrumented tool calls are outside the guarantee. Appendix B specifies content association, key trust, and freshness assumptions.

4.1 Architecture Overview

Distributed Sentinel pairs each agent A_i with a **Sentinel Sidecar** S_i, inspired by service mesh patterns (Linkerd, n.d.), to mediate incoming messages and effectful tool calls. Figure 1 illustrates how a binding survives a change of wording. Complete interception is a security assumption.

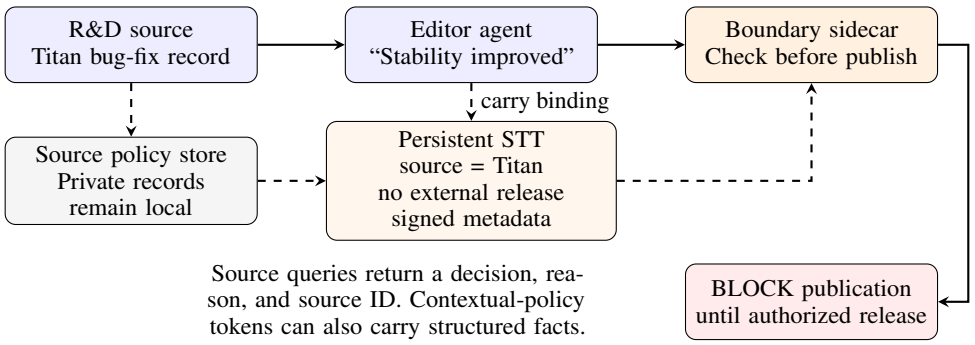

Figure 1: Binding continuity despite changed wording. Trusted instrumentation carries the Titan binding to the pre-effect check; source responses are not restricted to a Boolean. External-study tokens additionally carry structured facts. The evaluated implementation is in-process.

4.2 Semantic Taint Token (STT) Protocol

The **Semantic Taint Token** (STT) protocol adapts information-flow labels to agent boundaries. Conceptually, a token contains $\langle$`sources`, `labels`, `policy_state`, `sig`$\rangle$. Trusted instrumentation unions the recorded input sources when forwarding, summarizing, or aggregating artifacts.

This preserves identified bindings independently of output wording; it neither discovers omitted dependencies nor declassifies inherited sources. Conservative union can therefore retain irrelevant sources and cause false blocks.

The fixed-registry path carries source references and permission metadata; the external and dynamic paths additionally carry structured policy facts. Size grows with distinct retained sources and attached facts (Appendix D.5). Ed25519 protects signed metadata against unauthorized modification, not its truth or semantic correspondence to the artifact. Correct content–source association remains a grounding and instrumentation assumption.

4.3 Core Components

Shared input contract. Each enforcement path consumes represented source bindings, the proposed action and its arguments, and the policy facts required by the applicable predicates. Bindings identify the source whose restrictions govern an artifact; action arguments and facts can include actor, recipient, purpose, fields, and time. Policy codification is supplied. Missing required inputs fail closed. Appendix B, Table 5, distinguishes adapters to this contract; they are not successive scores of a single model.

Policy-state access and pre-effect checks. A coordinator routes source-reference queries to in-memory departmental stores; replies include a decision, reason, source ID, and timing. Contextual-policy paths also evaluate token-carried facts. The interceptor verifies signed metadata and action-bound claims, evaluates applicable predicates, and permits or blocks the effect. This makes Titan's restriction available before publication without reconstructing its origin from the final wording. Checks do not simulate arbitrary graph mutations or prove semantic dependence.

Conditional properties. Faithful codification, correct grounding, complete provenance, trusted interception, and identical policy versions and evaluation facts imply agreement with a centralized check of the same predicates (Appendix A). Conservative union preserves represented dependencies at the cost of possible overblocking. Source queries avoid moving the entire registry but can still reveal policy information; no transcript-privacy guarantee follows.

5 Experiments

We ask: **Q1**, does policy-state availability change decisions; **Q2**, can supplied bindings survive live transformations and support pre-effect checks; and **Q3**, what do rejection reasons reveal about external mapping failures? Table 2 collects the principal results. The studies separately measure answerability, executed effects, and refusal behavior, linking actionable policy state with checks before effects. Appendices B and G.1 specify interfaces and comparator access.

Table 2: Principal findings by study. Populations and outcomes differ; no pooled score is computed. Errors remain in all-case denominators.

Study	Observable	Retained result
Q1: 120 decisions	Binary-answer coverage	65.8% $\rightarrow$ 98.3% with full policy
Q2: 36 live tasks	Forbidden effects; safe completion	0/18; 17/18 (Sentinel)
Q2: 64 certified pairs	Safe sends; forbidden prevented	64/64; 64/64 with supplied bindings[a]
Q3: 126 external cases	Frozen operational; strict correct	72/126 (57.1%); 71/126
Q3: fixed-output replay	False blocks / slips after key intervention	54/0 $\rightarrow$ 14/1 (post-hoc; same 126 cases)
Complete plans: 200 cases	Risky approval; safe STOP change	0.6–3.8% locally; increases for 4/5 models[b]

[a]Matched controls reproduce the outcome, including a registry checker with independent source propagation (Appendix C.2). [b]Forced-binary contract; augmentation adds directives and registry facts; safe STOP excludes unresolved outputs (Table 4).

5.1 Q1: Policy-State Availability

Blind annotation, post-label intervention. We froze the 120-case extension before three research-independent colleagues labeled shuffled packets without generator answers. Of 120 cases, 116 were unanimous; consensus contains 59 ALLOW/61 BLOCK, with Krippendorff's $\alpha = 0.956$. One family misses its prespecified 10/10 balance gate; all cases and labels are retained (Appendix C).

In the post-label GPT-5.4 diagnostic, full policy reaches 0.967 accuracy versus 0.625 locally, a paired difference of 0.342 [0.258, 0.425] (family-stratified bootstrap; exact McNemar $p < 10^{-12}$). On 48 least-privilege cases, the rates are 0.938 versus 0.125. Binary coverage rises from 65.8% to 98.3%: of 41 newly correct answers, 38 were abstentions and 3 binary errors (Table 10). The gain mainly reflects restored answerability: supplying the facts lets the model decide cases on which it previously abstained.

Observation boundary. In the separate 64 certified pairs, permitted and prohibited worlds have identical downstream observations. Permissive local checking allows both; fail-closed checking blocks both; local-only GPT-5.4 abstains on every distinct observation. Here the same information deficit produces unsafe permission, false blocking, or abstention depending on the decision rule. Proposition 2.3 states the underlying distinguishability limit (Appendix C.2).

5.2 Q2: Preservation and Pre-Effect Mediation

Live-model execution. In 36 controlled workflows, a source analyst summarizes records, an editor rewrites the summary, and a dispatcher proposes tool calls with feedback. Trusted instrumentation carries all accessed source identities to the pre-effect check. Each condition runs in an isolated local sandbox; controls include a same-engine registry lookup given the same bindings (Appendix C.5).

DISTRIBUTED SENTINEL prevents 18/18 forbidden deliveries and completes permitted delivery/archive tasks in 17/18 cases, retaining one API failure. Endpoint audit detects all 18 forbidden deliveries only after execution. A separate 40-workflow placement control gives the same directional result (Appendices C.5 and D.2). These executions support carrying bindings through transformations and applying checks before effects.

Structured handoffs without binding loss. A separately frozen follow-up reuses all 36 tasks with explicit instructions to preserve structured metadata. All 72 transformations retain every binding and recipient rule. Model-relayed, host-carried, and restored contexts are identical: no binding-restoration effect is observed. Actors still propose seven prohibited sends, all stopped by the common guard and native DISTRIBUTED SENTINEL. This distinguishes preservation from compliance, not natural laundering recovery (Appendix C.6).

Controlled execution with supplied bindings. In the certified study, DISTRIBUTED SENTINEL allows 64/64 permitted sends and prevents 64/64 forbidden sends. Matched full-state controls reproduce these outcomes, including a registry checker with independent source propagation; Appendix C.2 reports each implementation. This checks enforcement with available policy state, complementing Q1's information intervention. Both symbolic and LLM guards can consume structured facts while raw source records remain local.

5.3 Q3: Rejection-Reason Audit of External Mapping

Frozen mapping-to-decision study. Three research-independent contributors wrote 42 cases each under seven public-policy operationalizations frozen before construction. For the final gold labels, three additional annotators each labeled all 126 unchanged cases using random opaque IDs and independently shuffled row orders, without prior labels or system predictions. Decisions were unanimous on 125/126 cases (nominal Krippendorff's $\alpha = 0.989$); a fourth blind adjudicator resolved the one decision and five step-only disagreements. The resulting consensus has 62 ALLOW/64 BLOCK cases. Rescoring the unchanged pre-label outputs, the frozen GPT-5.4 mapper plus per-effect DISTRIBUTED SENTINEL reaches 72/126 operational correctness (0.571), 1.000 recall, and 0.871 false-block rate (FBR); strict correctness is 71/126 after excluding one fail-closed parser error. Endpoint-only is also 72/126 with a different error profile (exact McNemar $p = 1.000$).

Audit the reason for each stop. We stratify decisions by policy-predicate, missing-input, and parser exits, then cross-tabulate them against reference labels. Of the 64 reference-BLOCK cases, 56 stop on missing required context, seven on policy predicates, and one on parser failure. Among the 54 false blocks on reference-ALLOW cases, 53 also stop on missing context and one on a purpose predicate. The shared missing-input exit identifies a concrete interface hypothesis to test despite perfect aggregate blocking recall.

Holding model responses fixed. We retrospectively rename keys in the saved outputs without changing values, actions, principals, targets, step order, labels, or the policy engine (Table 3). Renaming both sides removes 40/54 false blocks and introduces 1 slip under the new consensus: ten corrections need only policy-side changes, six only payload-side changes, and 24 both. The alias map was selected after inspection; the result attributes errors in these saved responses to the interface. These corrections require no new model reasoning; repair generalization requires fresh-case evaluation.

Table 3: Post-hoc fixed-output interface intervention on all 126 external cases (62 ALLOW/64 BLOCK). Only field names change. Operational correctness credits one fail-closed parser error on a BLOCK case; strict correctness excludes it. Outcomes are offline decisions.

Key intervention	Operational/126	Strict/126	False blocks/62	Slips/64
None (frozen replay)	72	71	54	0
Policy facts only	81	80	44	1
Action payloads only	78	77	48	0
Both sides	111	110	14	1

Residual semantics and comparator access. A separate failure-informed re-prompt declaring the existing schema yields 121/126 accuracy (0.960) and 0.065 FBR; its four false blocks concern purpose, consent, or an ambiguous action, and it misses one newly adjudicated role-access violation. It measures schema sensitivity on the same exposed cases. Retrospective direct GPT-5.4 and adapted PolicyGuard-4B guards score 125/126 and 98/126. Unlike the original mapper, these guards receive the exact encoded predicate and are scored on verdicts rather than structured workflows. Their results characterize predicate-level judgments under different input and output contracts. Q3 yields a practical audit sequence: inspect rejection reasons, intervene on the implicated interface while holding outputs fixed, then test remaining semantic errors on fresh cases (Appendices C.4 and E). Auxiliary mapper/encoder results remain in Appendices D.8 and F.

5.4 Measurement Boundaries and Costs

Complete plans reveal a utility cost. Under a forced-binary contract with every action and destination visible, local models approve 1–6/160 labeled-risky plans but stop 3–24/40 safe plans (Table 4). Augmentation increases safe STOPs for 4/5 models. Unresolved outputs remain separate, including nano's 24/160 local risky cases. These are plan judgments, not realized violations or certified CFVs. Augmentation adds registry facts as well as directives; it does not isolate instruction-only effects or missing bindings (Appendix D.10).

Table 4: Complete-plan diagnostic on generator-labeled plans. Entries are local → explicit; explicit adds global directives *and registry facts*. Risk+ is `PROCEED`, not an executed violation. Safe STOP excludes unresolved (Unres.) errors/invalid markers. Denominators retain all cases.

Model	Risk+ /160	Safe STOP /40	Unres. risk /160	Unres. safe /40
GPT-5.4-mini	$6 \to 0$	$10 \to 15$	$8 \to 0$	$3 \to 1$
GPT-5.4	$1 \to 1$	$6 \to 12$	$0 \to 0$	$0 \to 0$
GPT-5.4-nano	$3 \to 0$	$24 \to 11$	$24 \to 13$	$2 \to 3$
Sonnet 4.6	$3 \to 0$	$5 \to 13$	$0 \to 0$	$0 \to 0$
Opus 4.6	$3 \to 1$	$3 \to 13$	$0 \to 0$	$0 \to 0$

Why the complete-plan contract matters. Historical prompts omit the scored destination in all 20 side-channel cases, which account for at least half of each model's risky-case `PROCEED` markers. This prompt/scorer mismatch motivates the complete-plan diagnostic above. Appendix D.9 retains the per-model counts and Prompt Firewall audit; misses alone do not provide per-step CFV certification.

Protocol and name sensitivity. GPT-5.4 approves 1/160 labeled-risky plans under the forced-binary contract, versus 26/160 with original names in the separately frozen three-way protocol. Of 159 former STOPs, 51 become abstentions and 25 approvals; the original approval remains. Besides decision semantics, instructions and their placement, JSON formatting, and explicit external-endpoint attributes change. This is cross-protocol sensitivity, not an isolated output-format effect. Within the new protocol, neutral names raise local approvals from 26/160 to 37/160; full-policy inputs yield 0/160 under either naming, while blocking 14/40 safe plans. The original-name full-policy condition retains one API error. These are plan judgments, not strict CFV certificates (Appendix I.1).

Implementation overhead. On 72 supplied-binding probes, DISTRIBUTED SENTINEL's local signing, serialization, verification, and policy evaluation take a median 0.303 ms. Binding acquisition and actor/grounding costs are excluded. Appendix I.2 separates cached-engine and signed-path timings from model API measurements, preserving each timer boundary rather than ranking systems. Distinct-source tokens grow linearly (Appendix D.5); deployment costs remain unmeasured.

6 RELATED WORK

Contextual integrity defines context-relative privacy norms (Nissenbaum, 2004); ConfAIde and PrivacyLens test reasoning about them (Mireshghallah et al., 2024; Shao et al., 2024). Compositional privacy studies inference across interactions (Patil et al., 2025); PiSAs measures appropriateness and visibility across shared-agent components (Gupta et al., 2026). Our focus is the observation and binding contract needed for failure attribution. Information-flow control (IFC) predates LLM agents (Denning, 1976; Myers, 1999; Myers & Liskov, 1997); Fides, permissive flow analysis, f-secure, and CaMeL adapt it to them (Costa et al., 2025; Siddiqui et al., 2024; Wu et al., 2024; Debenedetti et al., 2025). Concurrent FlowSeal combines provenance-based tool interception with controlled declassification (Shim et al., 2026); our runtime uses conservative source union. Progent, AgentSpec, GuardAgent, and IsolateGPT address privileges, runtime constraints, and isolation (Shi et al., 2025; Wang et al., 2026; Xiang et al., 2024; Wu et al., 2025). We build on these principles to make observation, binding, and enforcement conditions explicit and separately measurable (Appendix G).

7 CONCLUSION

The central lesson is that useful privacy enforcement needs policy state that distinguishes permitted from forbidden actions at the point of execution, coupled with checks before effects. In Q1, supplying policy state restores answerability: 38 of 41 newly correct answers were previously abstentions. In the controlled certified pairs, supplied bindings support 64/64 permitted sends and prevent 64/64 forbidden sends across matched mechanisms. With neutral endpoint names and three-way decisions, GPT-5.4 approves 37/160 labeled-risky plans locally and 0/160 with full policy, which also blocks 14/40 safe plans. Separately, under forced-binary decisions, directives and registry facts increase safe-task refusals for four of five models. These protocol-conditioned results motivate reporting both safety and utility. Intact bindings can coexist with prohibited proposals; fixed-output replay locates false blocks at the mapping interface. Our formalization, layered diagnostics, and reference framework make these requirements separately testable. Practitioners should preserve identified bindings, check proposed effects, and report safe-task completion alongside violation prevention.

Limitations. Guarantees require faithful codification, complete grounding, trusted keys, and interception. Conservative propagation can overblock; signatures cannot certify semantics. No study independently certifies naturally occurring semantic laundering and its prevention end to end. Designed cases, finite samples, and public-transfer coverage failures bound generalization (Appendix E); the comparisons establish conditional behavior rather than state-of-the-art or deployment-cost rankings.

ETHICS STATEMENT

Dual Use. While DISTRIBUTED SENTINEL is designed to prevent policy violations, the same technology could theoretically be used to enforce unethical policies. We emphasize that the system enforces *organizational* policies, which should themselves be subject to ethical review.

Privacy. The source-query path avoids transferring full registry records, but responses include diagnostic metadata, and contextual-policy tokens carry structured facts. Neither path provides a transcript-level privacy guarantee. Even a restricted Boolean-query interface reveals all 57 registry profiles within eight adaptive queries in our controlled experiment (Table 16). Deployments would need authorization, purpose binding, query budgets, and audit; we make no differential-privacy claim.

Automation Bias. Operators should not blindly trust DISTRIBUTED SENTINEL decisions. We recommend human review for high-stakes blocking decisions.

Human annotation. The completed held-out and external-construction studies use fictional workflows and record only pseudonymous task labels, not participant demographics or private data. The contributors, fresh reannotators, and adjudicator were colleagues of the authors but had no role in this research or the system implementation. Participation was voluntary and uncompensated; each participant consented to anonymous use of the returned material and could decline or stop without consequence. Annotators were instructed to work independently without inspecting system predictions or prior labels; all disagreements were retained and no case was excluded. The final external consensus comes from the fresh opaque-ID reannotation in Appendix C.4, not the superseded sequential-ID judgments. The three scenario contributors explicitly confirmed that they used no generative AI for construction or their earlier annotation; the fresh protocol likewise prohibited generative-AI assistance. Because the tasks involved only judgments and fictional construction under public policies, collected no personal or enterprise data, and involved no intervention, the authors determined that institutional ethics review was not required. This is an author determination, not a formal exemption issued by an ethics board.

AI USE STATEMENT

Generative AI tools helped refine hypotheses and experimental design; inspect theoretical arguments; generate synthetic benchmark and entity-alignment examples; implement, test, and inspect code; plan statistical analyses; interpret machine-readable results; and draft or edit prose and figures. In the held-out extension, Gateway GPT-5.4 generated 100 scenarios and Codex generated 20 under a specification frozen before human labeling. By contrast, all three controlled external-study contributors explicitly confirmed that they used no generative AI to construct or annotate the 126 external cases; the fresh opaque-ID reannotation protocol also prohibited generative-AI assistance. GPT-5.4 was used as the frozen mapper under evaluation and in the separately disclosed post-hoc schema-interface sensitivity. The separate 36-case dynamic diagnostic and 128-world certified execution study were author/AI-designed, not independently human-authored. The former used live GPT-5.4-mini actors and a GPT-5.4 guard; the latter used fixed transformations and GPT-5.4 judgments. The structured-handoff follow-up reuses the 36 exposed tasks with new GPT-5.4-mini outputs and GPT-5.4 checks; it has no independent semantic-dependency labels (Appendix C.6). The small-encoder component study (Appendix F) uses AI-assisted programmed templates and labels. Human authors reviewed the artifacts, executed and audited experiments, checked claims and citations, and take responsibility for the final content.

REPRODUCIBILITY STATEMENT

Appendix A provides assumptions and proofs; Appendix B specifies the enforcement logic and trust boundaries. Appendices C and D describe dataset construction, annotation, evaluation protocols, model settings, and statistical procedures. Appendices E and F document adaptation and development exposure, and Appendix I.2 specifies timing boundaries. These appendices are included in this PDF. We plan to release code and per-case records after the review process.

REFERENCES

Sahar Abdelnabi, Amr Gomaa, Eugene Bagdasarian, Per Ola Kristensson, and Reza Shokri. Firewalls to secure dynamic LLM agentic networks. *arXiv preprint arXiv:2502.01822*, 2025. URL https://arxiv.org/abs/2502.01822.

Eugene Bagdasarian, Ren Yi, Sahra Ghalebikesabi, Peter Kairouz, Marco Gruteser, Sewoong Oh, Borja Balle, and Daniel Ramage. AirGapAgent: Protecting privacy-conscious conversational agents. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3868–3882, 2024. doi: 10.1145/3658644.3690350. URL https://doi.org/10.1145/3658644.3690350.

Arnar Birgisson, Joe Gibbs Politz, Úlfar Erlingsson, Ankur Taly, Michael Vrable, and Mark Lentczner. Macaroons: Cookies with contextual caveats for decentralized authorization in the cloud. In *Network and Distributed System Security Symposium*. Internet Society, 2014. URL https://research.google/pubs/macaroons-cookies-with-contextual-caveats-for-decentralized-authorization-in-the-cloud/.

Zhaorun Chen, Mintong Kang, and Bo Li. ShieldAgent: Shielding agents via verifiable safety policy reasoning. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 8313–8344, 2025. URL https://proceedings.mlr.press/v267/chen25ae.html.

Manuel Costa, Boris Köpf, Aashish Kolluri, Andrew Paverd, Mark Russinovich, Ahmed Salem, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Securing AI agents with information-flow control. *arXiv preprint arXiv:2505.23643*, 2025. URL https://arxiv.org/abs/2505.23643.

Saswat Das, Jameson Sandler, and Ferdinando Fioretto. Beyond jailbreaking: Auditing contextual privacy in LLM agents. *arXiv preprint arXiv:2506.10171*, 2025. URL https://arxiv.org/abs/2506.10171.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2636. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/97091a5177d8dc64b1da8bf3e1f6fb54-Abstract-Datasets_and_Benchmarks_Track.html.

Edoardo Debenedetti, Ilia Shumailov, Tianqi Fan, Jamie Hayes, Nicholas Carlini, Daniel Fabian, Christoph Kern, Chongyang Shi, Andreas Terzis, and Florian Tramèr. Defeating prompt injections by design. *arXiv preprint arXiv:2503.18813*, 2025. doi: 10.48550/arXiv.2503.18813. URL https://arxiv.org/abs/2503.18813.

Dorothy E Denning. A lattice model of secure information flow. *Communications of the ACM*, 19(5): 236–243, 1976. doi: 10.1145/360051.360056. URL https://doi.org/10.1145/360051.360056.

Sahra Ghalebikesabi, Eugene Bagdasaryan, Ren Yi, Itay Yona, Ilia Shumailov, Aneesh Pappu, Chongyang Shi, Laura Weidinger, Robert Stanforth, Leonard Berrada, Pushmeet Kohli, Po-Sen Huang, and Borja Balle. Operationalizing contextual integrity in privacy-conscious assistants. *arXiv preprint arXiv:2408.02373*, 2024. URL https://arxiv.org/abs/2408.02373.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. *arXiv preprint arXiv:2302.12173*, 2023. URL https://arxiv.org/abs/2302.12173.

Guardrails AI. Guardrails AI. Software repository, 2023. URL https://github.com/guardrails-ai/guardrails.

Shubham Gupta, Nazanin Mohammadi Sepahvand, Abhinav Kumar, Cem Subakan, Spandana Gella, Pierre-André Noël, Perouz Taslakian, Eugene Bagdasarian, and Valentina Zantedeschi. PiSAs: Benchmarking contextual integrity in multi-user agentic systems. *arXiv preprint arXiv:2607.05318*, 2026. URL https://arxiv.org/abs/2607.05318.

Michael B. Jones, John Bradley, and Nat Sakimura. JSON Web Token (JWT). RFC 7519, 2015. URL https://www.rfc-editor.org/rfc/rfc7519.

Juhee Kim, Woohyuk Choi, and Byoungyoung Lee. Prompt flow integrity to prevent privilege escalation in LLM agents. *arXiv preprint arXiv:2503.15547*, 2025. URL https://arxiv.org/abs/2503.15547.

Maxwell Krohn, Alexander Yip, Micah Brodsky, Natan Cliffer, M. Frans Kaashoek, Eddie Kohler, and Robert Morris. Information flow control for standard OS abstractions. In *Proceedings of the 21st ACM Symposium on Operating Systems Principles*, pp. 321–334, 2007. doi: 10.1145/1294261.1294293. URL https://doi.org/10.1145/1294261.1294293.

Wenkai Li, Liwen Sun, Zhenxiang Guan, Xuhui Zhou, and Maarten Sap. 1-2-3 check: Enhancing contextual privacy in LLM via multi-agent reasoning. In *Proceedings of the First Workshop on LLM Security (LLMSEC)*, pp. 115–128. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.llmsec-1.9/.

Linkerd. What is a service mesh? Linkerd documentation, n.d. URL https://linkerd.io/what-is-a-service-mesh/. Accessed 2026-09-25.

Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. Can LLMs keep a secret? testing privacy implications of language models via contextual integrity theory. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gmg7t8b4s0.

Andrew C Myers. JFlow: Practical mostly-static information flow control. In *Proceedings of the 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 228–241, 1999. doi: 10.1145/292540.292561. URL https://doi.org/10.1145/292540.292561.

Andrew C Myers and Barbara Liskov. A decentralized model for information flow control. In *Proceedings of the 16th ACM Symposium on Operating Systems Principles*, pp. 129–142, 1997. doi: 10.1145/268998.266669. URL https://doi.org/10.1145/268998.266669.

Helen Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 79(1):119–157, 2004. URL https://digitalcommons.law.uw.edu/wlr/vol79/iss1/10/.

NVIDIA. NeMo Guardrails. Software repository, 2023. URL https://github.com/NVIDIA-NeMo/Guardrails.

Vaidehi Patil, Elias Stengel-Eskin, and Mohit Bansal. The sum leaks more than its parts: Compositional privacy risks and mitigations in multi-agent collaboration. *arXiv preprint arXiv:2509.14284*, 2025. URL https://arxiv.org/abs/2509.14284.

Scott W. Rose, Oliver Borchert, Stuart Mitchell, and Sean Connelly. Zero trust architecture. Technical Report NIST SP 800-207, National Institute of Standards and Technology, 2020. URL https://doi.org/10.6028/NIST.SP.800-207.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J Maddison, and Tatsunori Hashimoto. Identifying the risks of lm agents with an lm-emulated sandbox. *arXiv preprint arXiv:2309.15817*, 2023. URL https://arxiv.org/abs/2309.15817.

Yijia Shao, Tianshi Li, Weiyan Shi, Yanchen Liu, and Diyi Yang. PrivacyLens: Evaluating privacy norm awareness of language models in action. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2837. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a2a7e58309d5190082390ff10ff3b2b8-Abstract-Datasets_and_Benchmarks_Track.html.

Tianneng Shi, Jingxuan He, Zhun Wang, Hongwei Li, Linyu Wu, Wenbo Guo, and Dawn Song. Progent: Securing AI agents with privilege control. *arXiv preprint arXiv:2504.11703*, 2025. URL https://arxiv.org/abs/2504.11703. Version 3 revised in 2026.

Minsun Shim, Ramisha Raida Karim, Ruthwik Jakkula, Kaiwen Zhou, Xin Liu, Xin Eric Wang, and Zhou Li. Confuse the model, control the flow: Understanding and mitigating privacy leakage from LLM agents with information flow control. *arXiv preprint arXiv:2609.14003*, 2026. URL https://arxiv.org/abs/2609.14003.

Shoaib Ahmed Siddiqui, Radhika Gaonkar, Boris Köpf, David Krueger, Andrew Paverd, Ahmed Salem, Shruti Tople, Lukas Wutschitz, Menglin Xia, and Santiago Zanella-Béguelin. Permissive information-flow analysis for large language models. *arXiv preprint arXiv:2410.03055*, 2024. URL https://arxiv.org/abs/2410.03055.

W3C Provenance Working Group. PROV-overview. W3C Working Group Note, 2013. URL https://www.w3.org/TR/prov-overview/.

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. AgentSpec: Customizable runtime enforcement for safe and reliable LLM agents. In *Proceedings of the 48th IEEE/ACM International Conference on Software Engineering*, pp. 2938–2950, 2026. doi: 10.1145/3744916.3764546. URL https://arxiv.org/abs/2503.18666.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024. doi: 10.1007/s11704-024-40231-1. URL https://doi.org/10.1007/s11704-024-40231-1.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*, 2024. doi: 10.48550/arXiv.2412.13663. URL https://arxiv.org/abs/2412.13663.

Xiaofei Wen, Wenjie Jacky Mo, Yanan Xie, Peng Qi, and Muhao Chen. Learning efficient guardrails for compliance. *arXiv preprint arXiv:2510.03485, version 2*, 2026. doi: 10.48550/arXiv.2510.03485. URL https://arxiv.org/abs/2510.03485v2. Accepted at ICML 2026.

Fangzhou Wu, Ethan Cecchetti, and Chaowei Xiao. System-level defense against indirect prompt injection attacks: An information flow control perspective. *arXiv preprint arXiv:2409.19091*, 2024. URL https://arxiv.org/abs/2409.19091.

Yuhao Wu, Franziska Roesner, Tadayoshi Kohno, Ning Zhang, and Umar Iqbal. IsolateGPT: An execution isolation architecture for LLM-based agentic systems. In *Network and Distributed System Security Symposium*, 2025. URL https://arxiv.org/abs/2403.04960.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023. URL https://arxiv.org/abs/2309.07864.

Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, Dawn Song, and Bo Li. GuardAgent: Safeguard LLM agents by a guard agent via knowledge-enabled reasoning. *arXiv preprint arXiv:2406.09187*, 2024. URL https://arxiv.org/abs/2406.09187.

Nickolai Zeldovich, Silas Boyd-Wickizer, Eddie Kohler, and David Mazières. Making information flow explicit in HiStar. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation*. USENIX Association, 2006. URL https://www.usenix.org/conference/osdi-06/making-information-flow-explicit-histar.

Peter Yong Zhong, Siyuan Chen, Ruiqi Wang, McKenna McCall, Ben L. Titzer, Heather Miller, and Phillip B. Gibbons. RTBAS: Defending LLM agents against prompt injection and privacy leakage. *arXiv preprint arXiv:2502.08966*, 2025. URL https://arxiv.org/abs/2502.08966.

A THEORETICAL PROOFS

Proof of Proposition 2.3. Let a deterministic local mechanism be a function $\mathcal{E}_i : O_i \to \{\text{ALLOW}, \text{BLOCK}\}$. By hypothesis, $O_i(\mathcal{S}_0) = O_i(\mathcal{S}_1) = o$. Functionality therefore gives $\mathcal{E}_i(O_i(\mathcal{S}_0)) = \mathcal{E}_i(o) = \mathcal{E}_i(O_i(\mathcal{S}_1))$. If this common decision is ALLOW, violation completeness fails on $\mathcal{S}_1$; if it is BLOCK, sound utility fails on $\mathcal{S}_0$. Thus no mechanism restricted to the shared local observation can satisfy both properties. For a randomized binary mechanism, let p be its probability of allowing observation o. Its safe-world allow rate is p and its violating-world block rate is $1 - p$; their mean under a uniform prior is exactly $1/2$. Abstention cannot improve this joint correctness when counted as neither a correct allow nor a correct block. The proposition does not claim that every violation has such a safe observational twin, nor that a centralized or cross-domain mechanism is impossible. □

Executable observation witness. We enumerate all reader-set assignments for two sources over three recipients. Internal transformations are explicitly permitted; the downstream action, destination, payload, directive, and visible graph are held fixed while source-owned recipient permissions vary. Each prohibited world has a permitted counterpart with the same downstream observation. A permissive local checker passes every step when no visible binding excludes the action, whereas a fail-closed checker also rejects indistinguishable permitted counterparts. This finite specification illustrates the proposition's assumptions, not model failure or benchmark prevalence; it does not certify any historical case as CFV-hard.

Same-check scope. Definition 2.2 excludes checker-scope mismatches. For example, a local check of role permissions and a global check of "at most one send" can disagree even with identical complete histories; that alone is not context fragmentation. The send-count invariant must be checked at the same action boundary by both evaluators, with only visibility of the relevant prefix state differing. The proposition likewise concerns opposite full-state permissions for the same proposed effect, not a violation elsewhere in one sequence. The frozen controlled certificates are retrospectively re-audited under this contract in Appendix C.2.

Proposition A.1 (Monotonicity of Constraint Propagation)**.** *Fix an action, a policy version, and its authorization/release facts. Order states only by addition of source dependencies and their restrictions, with denial given by disjunction over those restrictions. Then adding dependencies cannot turn a block into an allow. Thus (a) local blocks imply global blocks; (b) if one local state contains all relevant restrictions and evaluation facts, it yields the same decision as the global state. This statement does not cover arbitrary graph updates or unknown-as-deny local checks.*

Proof. Under the stated order, a block has a witnessing restriction. Adding dependencies preserves that witness, proving (a). Under (b), the local and global evaluators have the same predicates and arguments, so their disjunctions coincide. Adding an authorization, executing an approved release, changing consent, or replacing a policy version is not an addition in this order. Such events require fresh evaluation; monotone provenance union is not monotone policy truth under arbitrary updates. □

Corollary A.2 (Conditional Equivalence to Centralized Enforcement)**.** *Assume (i) faithful policy codification, (ii) complete relevant source bindings, (iii) trusted interception of every effect, (iv) complete entity grounding, and (v) identical policy versions, authorization facts, action arguments, and relevant aggregate/history state at the distributed and centralized evaluations. Both evaluate the same predicate set with the same composition rule. Then they return the same decisions for the intercepted sequence.*

Proof. At the first effect, (i), (ii), (iv), and (v) supply identical predicate inputs and composition, hence identical decisions. Assuming equality through effect k, trusted interception (iii) applies those decisions before state changes; (v) supplies the same relevant updated history at effect $k + 1$. The

same argument gives equality there, completing induction. Delayed replicas, omitted joint predicates, or unobserved effects violate the assumptions. Equivalence concerns the encoded policy, not the completeness of its natural-language interpretation. □

Proposition A.3 (Preservation of Represented Provenance)**.** *Suppose each transformation in an acyclic instrumented workflow attaches the union of its inputs' represented bindings to every output. Every binding represented at an input ancestor then remains at each descendant output. This conditional preservation property concerns recorded bindings, not the recovery or semantic relevance of source dependencies.*

Proof. At one transformation, each input binding belongs to the attached union. Induction over a topological ordering preserves that inclusion through every descendant transformation. Thus paraphrasing or summarization cannot remove a represented binding under this rule. Inputs that bypass instrumentation and sources omitted by grounding are outside the premise. The rule may also retain semantically irrelevant bindings and cause false blocks. □

B Implementation Details

Role of the Registry controls. The locally constructed attribute-based access-control (ABAC) Registry is a mechanism control, not an independent Open Policy Agent (OPA) reproduction or a competing deployed system. In the dynamic, mapping, and check-cost studies it evaluates supplied bindings with the same policy engine as Distributed Sentinel. In the certified study it independently propagates source IDs through the computation graph and evaluates recipient permissions. These controls test enforcement with available policy state; they do not measure how a production registry acquires, synchronizes, or authenticates that state. The retained results describe these implementations under their stated interfaces, not a general ranking of architectures.

Table 5 separates the input adapters and their evaluation targets.

Table 5: Grounding interfaces. All feed a policy checker, but their prediction targets and evidence differ. Supplied bindings in mechanism tests bypass learned acquisition.

Adapter	Predicted interface	Evaluation role
Qwen3-0.6B	Text to registry entity sets; candidate retrieval then selection	Historical response/workflow pairs and controlled mapper stress
Frozen GPT-5.4	Scenario to structured actions and policy facts; source identity supplied	Primary 126-case mapping/decision chain; separate post-hoc schema sensitivity
Developmental adapters	Small-encoder action/actor and field/purpose outputs; separate policy compiler	Component, exposed workflow, and public-transfer diagnostics only

B.1 Core Code Examples

We provide abridged implementation excerpts from the prototype; omitted lines perform serialization and error handling.

B.1.1 Taint Token Data Structure

```
class TaintToken:
    def __init__(self, source, sensitivity,
                 permissions, lineage=None,
                 provenance_sources=None):
        self.source = source
        self.sensitivity = sensitivity
        self.permissions = permissions
        self.lineage = lineage or Lineage()
        self.provenance_sources = (
```

```
            provenance_sources or [source])

    def to_ed25519(self, private_key):
        header = {"alg": "EdDSA", "typ": "STT"}
        signing_input = encode(header, self.to_dict())
        signature = private_key.sign(signing_input)
        return compact(signing_input, signature)
```

The prototype's evaluated operators are ordered forwarding, paraphrase or summary, and multi-source aggregation. For each, `merge_sources` unions every source entity already represented by the input tokens with the currently grounded entity. Reconstruction from an ungrounded source and effects executed outside this loop are not modeled; they violate Proposition A.3's instrumentation premise rather than being covered by its conclusion.

B.1.2 Sentinel Evaluation Logic

```
def evaluate(workflow):
    tokens = []
    for flow in workflow.flows:
        token = merge_sources(tokens, flow.entity)
        if flow.is_external_or_public():
            if token is None:       # fail closed
                return BLOCK
            for source in token.provenance_sources:
                if not query_source(source,
                                    flow.action,
                                    flow.target):
                    return BLOCK
        tokens = [token]
    return ALLOW
```

B.2 Sidecar Implementation

The evaluated artifact is an in-process Python prototype with the following components:

- **Interceptor**: mediates the simulated agent/tool boundary and fails closed for untagged external or public actions;
- **STT Engine**: generates, merges, signs, and validates provenance tokens while retaining all source graph references;
- **Department Store**: maintains an in-memory policy registry and Bloom-filter precheck per domain;
- **Query Coordinator**: routes in-process source-domain predicate queries. Network transport and durable persistence are not evaluated.

B.3 Neuro-Symbolic Mapper

The enforcement registry contains 98 benchmark data types across 11 policy domains, 17 agents, and 18 sinks. The Qwen3-0.6B mapper is trained on a separate catalog of 90 entities across six departments; 16 entity identifiers overlap with the enforcement registry. At inference, retrieval selects candidates from the enforcement registry, exposing the mapper to many identifiers absent from its training catalog. This is partial identifier separation, not ontology-disjoint evaluation or proof against semantic/template overlap. Session provenance (the executing agent's domain) is included as a disambiguating feature. We report both end-to-end policy outcomes and entity-level metrics in Appendix D.

B.4 STT Cryptographic Protocol

Tokens use Ed25519 with per-department key pairs. Below are selected fields from an illustrative source-reference token; token identifiers, additional source fields, expiry, and audit history are

omitted. Contextual-policy paths populate `policy_contexts` with source-bound facts and policy references. Signatures wrap the serialized payload; signed action envelopes additionally bind actor, target, and action. Verification authenticates these assertions, not their semantic correctness.

Trust and freshness boundary. The envelope binds metadata and request claims, not a digest of the actual effect payload. Trusted instrumentation must associate the artifact with its complete contributing bindings and supply the authorized verification key. Replay state is local to a gateway instance, not durable across replicas or restarts. The source-query implementation caches decisions with a 300-second TTL; a policy update needs cache invalidation or expiry before that path reflects the change. The evaluated fixed-state checks therefore do not establish immediate revocation or adversarial cross-domain transport security.

```
{
  "version": "1.0",
  "source": {"graph_node": "urn:example:rd:node:titan"},
  "sensitivity": {"level": "RESTRICTED"},
  "permissions": {"allowed": ["read", "summarize"],
    "forbidden": ["send_external", "write_public"]},
  "provenance_sources": [
    {"graph_node": "urn:example:rd:node:titan"}],
  "policy_contexts": []
}
```

B.5 Illustrative Policy Notation

The example below illustrates an invariant in pseudocode. The evaluated implementation uses Python policy rules and structured contextual predicates; it does not implement a parser for this notation.

```
INVARIANT nda_protection:
  FOR entity IN graph
  WHERE entity.has_label("nda_protected")
  BLOCK action
  WHERE action.audience IN ["external", "public"]
  MESSAGE "NDA-protected content cannot be shared externally"
```

C Evaluation Suite Details

The structured-handoff follow-up reuses the dynamic study's 36 tasks, not a new population; it fixes one generated proposal across information conditions (Appendix C.6). Complete-plan judgments reuse the 200-case corpus with a revised prompt contract (Appendix D.10).

Oracle-input mechanism control. On the 200 co-designed workflows, we compare keyword data-loss prevention (DLP), a single-action GPT-5.4 Prompt Firewall, a centralized encoded-policy evaluator, and the matched Registry and Sentinel mechanisms. With supplied policy entities, Sentinel and Registry tie at F1=0.994 [0.984, 1.000]. The centralized evaluator's F1=0.991 reflects three disagreements with generator labels, not an omniscient oracle failing. Against its encoded decisions, both runtime controls reach F1=0.997, zero slip, and 2.44% FBR. All three disagreements remain in the evaluation records and reported scores.

C.1 PhantomEcosystem Construction and Scope

The generator (seed 2026) instantiates deterministic family templates with seeded choices of agents, entities, and sinks; descriptions are stored verbatim in each JSON case. We retain workflow objects, saved prompts, registry, generator, and prompt-adaptation scripts. An offline audit regenerates all case content by ID and checks the saved prompt variants. Record order, creation time, and added policy-check annotations are not generator outputs to be overwritten. This co-designed set has no independent human quality-assurance stage. Construction used:

Table 6: Complete evaluation roadmap. Sizes denote workflows unless specified. Scores from different contracts are not pooled into a leaderboard.

Evaluation	Size	Purpose and evidence boundary
Blinded policy diagnostic	120	Q1: full vs. local information; post-label exploratory test
Dynamic runtime	36	Q2: live transformations and realized local effects; 144 branches
Frozen external test	126	Q3: mapping-to-decision chain with supplied source identities; post-hoc interface attribution
Certified paired execution	128 worlds	Q1/Q2 support: 64 controlled pairs with local/global certificates
PhantomEcosystem	200	Q2 support: oracle-input consistency; co-designed stress set
Placement intervention	40	Q2 support: early forbidden effect vs. endpoint checking
Mapper stress	64	Q3: lexical/context perturbations of entity grounding
Public transfer	64	Auxiliary: a separate developmental compiler's coverage
Encoder component test	3,744	Auxiliary: action/actor outputs, not policy decisions
Workflow development	378	Auxiliary: exposed complete-pipeline diagnostic (appendix)

Table 7: Oracle-input mechanism diagnostic on 200 co-designed workflows, not an independent language-to-policy test. Generator labels disagree with the centralized encoded-policy evaluator on three cases. Confidence intervals use 10,000 class-stratified bootstrap samples. FBR is false-block rate on controls; Slip is the labeled-attack allowance rate.

Method	F1 [95% CI]	Prec.	Rec.	FBR↓	Slip↓
No Protection	0.000 [0.000, 0.000]	0.000	0.000	0.000	1.000
Rule-Based DLP	0.613 [0.536, 0.683]	0.960	0.450	0.075	0.550
Prompt Firewall	0.849 [0.801, 0.893]	1.000	0.738	0.000	0.263
Registry ABAC Guard	0.994 [0.984, 1.000]	0.994	0.994	0.025	0.006
Encoded-policy evaluator	0.991 [0.978, 1.000]	0.994	0.988	0.025	0.013
DISTRIBUTED SENTINEL	0.994 [0.984, 1.000]	0.994	0.994	0.025	0.006

1. Nine handwritten family generators plus separate legitimate-control templates;
2. a fixed catalog of 17 agents, named data entities, and destination sinks;
3. seeded variation in topology, action, entity, description, and destination; and
4. author inspection for syntax, duplicate IDs, and policy-registry consistency.

The author-assigned difficulty distribution is 40/60/100 easy/medium/hard. Most risky families contain 20 cases; cross-org and authorization bypass contain 10 each. The 40 controls are constructed separately, not paired to the 160 attacks. Authorization-bypass cases encode requested privileged publication, not forged token bytes; substrate tests are separate (Table 14). For example, combining headcount with compensation bands can expose a single-member group's salary. This motivates an aggregation family, but the encoded evaluator does not independently discover such sensitive inferences.

C.2 CERTIFIED PAIRED EXECUTION

We construct 64 author/AI-designed pairs (128 worlds) crossing four dataflow topologies, source counts of 1/2/4/8, transformation depths of 1/4, and two payload styles. The topologies are chain, parallel merge, fork/rejoin, and irrelevant read. Each pair flips one contributing source's recipient permission while preserving the complete downstream observation. Internal operations are permitted; a local checker denies only a visibly prohibited action. The executable specification certifies every

Table 8: PhantomEcosystem stress families. Family names describe motivating risks, not per-case CFV certificates or demonstrated semantic-inference capabilities.

Category	Description	Representative Flow
Direct Leak	Sensitive data sent to unauthorized external destination	HR → external storage
Multi-hop	Data laundered through intermediate agents or tools	agents → multiple hops
Aggregation	Individually safe fields combine into sensitive conclusions	analytics → sales/HR
Time-series	Temporal patterns reveal strategic or restricted state	logs → downstream reporting
Side-channel	Restricted info inferred from metadata or behavior	external → internal signal
Scope creep	Initially legitimate workflows expand beyond authorized use	finance → operations
Data reconstr.	Partial fragments combine to reconstruct a secret	sources → complete secret
Cross-org	Internal data crosses organizational boundaries	R&D → partner/competitor
Auth. bypass	A workflow requests a privileged effect after purported credential expansion	actor → privileged resource

local step as allowed and exactly one world per pair as globally denied. Worlds, certificates, adapters, and prompts were frozen before evaluation. This study is distinct from both the finite observation witness in Appendix A and the uncertified historical corpus.

Retrospective same-check audit. We replay the frozen certificates, then apply their unchanged local-check function to each recorded observation and an enriched copy containing the contributing-source permissions at that step. All other inputs and the execution prefix remain fixed. All 1,304 paired step checks reproduce the recorded verdicts: 64 worlds satisfy Definition 2.2, with 64 allowed controls. This narrows and validates the interpretation of existing witnesses; it adds no experimental trials and changes no frozen record. We retain every step comparison.

The existing DISTRIBUTED SENTINEL gateway, unmodified CaMeL interpreter with an equivalent recipient-policy adapter, Registry control, and full-state GPT-5.4 each produce 128/128 correct decisions, zero forbidden sends out of 64, and 64/64 permitted sends to isolated in-memory sinks. Both provenance mechanisms preserve the exact contributing-source sets, including irrelevant-read controls. The permissive local control instead executes all 64 forbidden sends; fail-closed local control completes none of the 64 allowed sends. Local-only GPT-5.4 returns `INSUFFICIENT_INFORMATION` on all 64 distinct observations, with each response reused across its identical-observation pair. Its binary coverage is zero; this is appropriate uncertainty under missing information, not evidence of deficient reasoning.

The run retains 192 first model responses: 64 local and 128 full-state queries, using GPT-5.4-2026-03-05 at temperature zero, with no invalid response or failed call. Deterministic replay reproduces 640 method records. The audit verifies 64 paired certificates, 1,304 local checks, 1,328 intermediate text/source checks, 128 signed envelopes, and all response bindings; 11 fixture tests and 106 upstream interpreter tests pass.

All plans are fully specified and transformations are fixed string operations. Supplied source identities remove grounding uncertainty, and no live planner or production tool is evaluated. The grid is not 128 independent real-world samples; it supports conditional enforcement and the observation boundary, not a hard semantic benchmark, CFV prevalence, or method superiority. Full-state GPT-5.4 receives source records and permissions in this interface; this does not establish that every LLM guard needs the full records.

C.3 BLINDED POLICY-INFORMATION DIAGNOSTIC

Held-out annotation extension. The separate validation set was frozen on 2026-09-01 at SHA-256 `c152f0d284e2d9f32c5b33d51c112982928ca562e9c21b9c866b179e3909b3ab`. It contains 120 scenarios from six policy families absent from the main benchmark, with 20 cases per family and a hidden generator quota of 60 ALLOW and 60 BLOCK. Gateway GPT-5.4 generated the first 100 cases; after its authentication became unavailable, Codex generated the final 20 under the same frozen specification. Public annotation materials omit generator decisions and private generation traces. This construction supports blinded, independent human annotation, but not a claim of independently human-authored scenarios.

All three annotators were colleagues who were not involved in the research or system implementation. They completed all 120 cases without access to generator labels or system predictions. Majority consensus produced 59 ALLOW and 61 BLOCK cases. Four cases were resolved by a two-of-three majority and none required fourth-person adjudication. Pairwise raw agreement was 0.975, 0.992, and 0.967; corresponding Cohen's κ values were 0.950, 0.983, and 0.933, and nominal Krippendorff's α was 0.956. The hidden construction key agreed with consensus on 119/120 cases. The human relabeling changes `consent_revocation_propagation` from the intended 10/10 split to 9 ALLOW/11 BLOCK; the validator therefore preserves a failed exact-balance check rather than silently changing a label or exclusion rule.

After freezing the human labels, we ran an information-conditioned diagnostic through the documented AI Gateway with GPT-5.4 (temperature 0). The full-policy condition receives the organization policy, relevant records, local context, and workflow; the local condition receives only the acting agent's context and workflow. This is a semantic decision diagnostic, not an execution of the complete cryptographic/interception substrate. The paired accuracy improvement is 0.342 [0.258, 0.425], and the paired block-F1 improvement is 0.199 [0.118, 0.295], under 10,000 family-stratified resamples. Full policy is uniquely correct on 41 cases and local context on none (two-sided exact McNemar $p = 9.09 \times 10^{-13}$). The four non-unanimous cases are retained in a complete disagreement audit; they concern temporal state, principal binding, and the scope of consent withdrawal rather than random errors.

Table 9: Exploratory results on the independently annotated extension. Parentheses are 95% family-stratified bootstrap intervals (10,000 resamples). "Explicit" has 72 cases and "least privilege" has 48.

Information condition	Accuracy	Block F1	Explicit acc.	Least-priv. acc.	Abstain
Full policy	.967 (.933–.992)	.984 (.958–1.000)	.986	.938	2
Local context only	.625 (.542–.708)	.784 (.688–.864)	.958	.125	41

Coverage and paired transitions. The retrospective per-case recount separates abstention from binary error (Table 10). All 75 locally correct cases remain correct. Of 41 local abstentions, 38 become correct, one becomes incorrect, and two remain abstentions. Of four local binary errors, three become correct and one stays wrong. Thus 38 of the 41 gains restore answerability. Conditional correctness is reported descriptively on different answered subsets; it is not a matched causal estimate of reasoning improvement. All cases, including remaining errors and abstentions, stay in the original accuracy denominator.

Table 10: Q1 decision coverage, separating abstentions from binary errors. Accuracy on answered cases has a condition-specific denominator; those subsets differ. The original all-case scores are unchanged.

Information	Answered / all	Coverage	Correct / answered	Wrong / abstain
Local context	79/120	65.8%	75/79 (94.9%)	4 / 41
Full policy	118/120	98.3%	116/118 (98.3%)	2 / 2

C.4 CONTRIBUTOR-AUTHORED CASES WITH BLIND REANNOTATION

Before scenario construction, we froze a generic contextual-predicate domain-specific language (DSL), source-bound signed policy contexts, seven public-source policy operationalizations, the GPT-5.4 mapper prompt, the signed evaluator, and a no-tuning/no-rerun rule. Three contributors each supplied 42 fictional cases (six per policy) and explicitly confirmed no generative-AI use in construction. The original cross-annotation packets retained sequential IDs from an alternating authoring schedule; those judgments are therefore superseded and are not the gold labels used below.

For the final consensus, we reidentified the unchanged 126 cases with random opaque IDs carrying no author, order, policy-slot, or label information. Three additional research-independent annotators each labeled every case in an independently randomized order without prior labels, construction slots, or system predictions. Returned files were hashed before the private mapping was opened. Decisions were unanimous on 125/126 cases; pairwise raw agreement was 0.992/1.000/0.992, Cohen's κ was 0.984/1.000/0.984, and nominal Krippendorff's α was 0.989. Exact decision-plus-step labels were unanimous on 120/126 cases ($\alpha = 0.950$). A fourth blind adjudicator labeled only those six rows, yielding 62 ALLOW and 64 BLOCK cases. No case was excluded. Relative to the superseded consensus, one decision and two first-violation steps changed. This reannotation repairs the identifier cue; it does not retroactively make the original cross-annotation blind.

The mapper ran before completed annotations were returned, at temperature zero through the documented Gateway path; failed or structurally invalid mappings were retained and no case was removed or revised based on system output. It produced 125 structurally valid rows and one retained invalid row. Although its action sequence exactly matches the private author structure on 117/126 cases, exact outcome plus first-step agreement is only 59/126 because the prompt did not enumerate each policy's canonical schema keys. Examples include `retention_deadline` for `retention_until` and `fields` for `requested_fields`; the evaluator correctly treats missing required keys as fail-closed.

Table 11: Frozen external end-to-end results against adjudicated human consensus. FBR is the false-block rate; Slip is the block-class false-negative rate.

Method	Acc.	Prec.	Rec.	F1	FBR	Slip
DISTRIBUTED SENTINEL per effect	.571	.542	1.000	.703	.871	.000
Per-effect Registry	.571	.542	1.000	.703	.871	.000
Endpoint only	.571	.561	.719	.630	.581	.281
No policy state	.500	1.000	.016	.031	.000	.984

Wilson 95% intervals are [0.484, 0.654] for DISTRIBUTED SENTINEL accuracy and [0.766, 0.933] for FBR. DISTRIBUTED SENTINEL and endpoint-only are each uniquely correct on 18 cases (two-sided exact McNemar $p = 1.000$). Consequently, this study supports neither an external accuracy advantage nor a DISTRIBUTED SENTINEL-specific advantage over matched per-effect enforcement. It instead exposes a concrete, frozen mapper-interface failure; that primary run remains unchanged, without alias repair, prompt edits, or selective case reruns.

Post-hoc schema-interface sensitivity. After diagnosing the frozen failure, we froze a separate failure-informed analysis protocol that enumerates the canonical keys already present in the frozen policy bundle. We changed neither cases, human labels, policy predicates, model snapshot, temperature, nor evaluator, and retained every first response. All 126 mappings were structurally valid. Per-effect DISTRIBUTED SENTINEL reached 0.960 accuracy [0.910, 0.983], 0.984 recall, 0.962 F1, and 0.065 FBR [0.025, 0.154], correcting 50 primary-run errors and regressing on one. This comparison is descriptive because the intervention was designed after inspecting the primary failures. The same mappings yield 0.841 endpoint-only accuracy; per-effect enforcement is uniquely correct on 17 cases and endpoint-only on two (exact McNemar $p = 0.00073$). Per-effect Registry again exactly ties DISTRIBUTED SENTINEL, so the sensitivity separates two effects without implying a proprietary accuracy advantage: a declared schema repairs interface binding, while pre-effect placement catches violations that endpoint-only checking misses.

The five remaining errors comprise four false blocks—two purpose-normalization or implicit-purpose mismatches, one missed explicit consent status, and one ambiguous status-writing step parsed as

storage rather than the author's logging action—plus one missed role-access violation. Against the private author structures, policy-fact exact match is 123/126 and action-sequence exact match is 120/126. These residuals preserve grounding as an open problem. The original frozen table and estimate remain the primary result; complete prompts, outputs, hashes, and the post-hoc analysis protocol are retained.

Fixed-output interface intervention. To separate key mismatches from changes in model responses, we replay the original outputs with a post-hoc alias map, changing only policy-fact keys, only action-payload keys, or both. Values, actions, principals, targets, step order, labels, and the policy engine remain fixed; the invalid mapping remains invalid. Of 54 original false blocks, 53 exit on missing required context and one on a purpose predicate. Renaming keys alone removes 40 false blocks with one regression (Table 3). Ten corrections require only policy-side renaming, six only payload-side renaming, and 24 require both. Exit reasons are not semantic ground truth: the remaining errors include absent arguments, incompatible values, and ambiguous actions.

The 40 corrections establish a representation-interface contribution to the observed failure without rerunning the model. The alias map was selected after inspection, so this is neither independent generalization nor a new learned method. Matched Registry replay gives identical decisions. Because source identities and policy references are supplied by this evaluation harness, the study does not measure binding loss during live inter-agent transformations.

Boundary uncertainty. The frozen run has zero observed slips among 64 BLOCK cases. Its exact one-sided 95% upper bound is 4.57%, and the two-sided Clopper–Pearson interval is [0, 5.60%] (Wilson: [0, 5.66%]); corresponding recall has exact two-sided interval [0.944, 1.000]. The post-hoc run has one slip in 64 cases (recall 0.984; exact two-sided interval [0.916, 1.000]). These binomial summaries are conditional on the designed study, not guarantees for enterprises; they supplement, rather than overwrite, frozen bootstrap files.

C.5 LIVE-MODEL TRANSFORMATIONS AND EFFECTFUL TOOL EXECUTION

We ran a separately frozen, author/AI-designed mechanism diagnostic on 36 fictional workflows: three families (release briefing, vendor handoff, joint digest), six source counts (1, 2, 3, 4, 6, 8), and matched allowed/forbidden pairs. Within each pair, facts and requested recipient are identical; one source's recipient permission differs. Policies, prompts, runtime, client, and enforcement dependencies were hashed before experimental requests. This is controlled runtime evidence, not independent human validation, new policy learning, or deployment evidence.

An analyst summarizes records through a real model call, a separate editor rewrites that summary, and a dispatcher proposes one effect at a time. The dispatcher sees actual execution/block feedback before its next model call, with a four-turn limit. Actors use GPT-5.4-mini-2026-03-17; the full-policy guard uses GPT-5.4-2026-03-05, at temperature zero. Analyst/editor outputs and the first dispatcher proposal are shared between conditions; later proposals branch on each condition's feedback. Condition order is seed-counterbalanced. Trusted host instrumentation conservatively carries all accessed source IDs through both transformations; paraphrase does not declassify content. Thus we test metadata continuity across actual transformations, not semantic recovery of dependencies from prose.

The four conditions receive identical prepared source bindings and permissions: signed per-effect DISTRIBUTED SENTINEL, cached per-effect Registry using the same engine, a full-policy LLM guard, and deferred endpoint audit. A recipient must be authorized by every source. Internal archiving is allowed. Tools materialize the exact immutable artifact in isolated local destination or archive files; no real email or production endpoint is contacted. An independent set-membership oracle and file-content reconciliation audit attempted and realized effects, separately from the tested policy engine.

The run contains 144 branch records, 455 unique model requests (452 successful, three HTTP 500 failures), and 224 reconciled effect files. One shared analyst failure prevents all four conditions from completing one safe case; one LLM guard failure blocks an allowed archive; one dispatcher failure after a DISTRIBUTED SENTINEL block prevents fallback completion. None was retried or excluded. All 18 forbidden requests produce a forbidden proposal in each condition. Each pre-effect guard stops all 18, whereas endpoint audit detects all 18 only after delivery. Source-policy unions remain

Table 12: Live-model controlled diagnostic. Slip is realized forbidden delivery; safe completion requires delivery, archive, finish, and no error; fallback requires archive/finish without delivery or error on forbidden requests. All denominators retain failures. Time is median summed model-call and effect-check duration (seconds), not production latency. Bytes are median logical policy metadata, not network traffic.

Condition	Slip	Safe	Fallback	Time	Bytes
Sentinel	0/18	17/18	17/18	10.42	13,368
Per-effect Registry	0/18	17/18	18/18	10.24	1,085
Full-policy LLM guard	0/18	16/18	18/18	14.57	2,318
Endpoint audit	18/18	17/18	0/18	10.33	1,018

intact through both transformations in all 35 successfully prepared cases. Event-level false blocks are 0/51 for DISTRIBUTED SENTINEL, 0/52 for Registry, and 1/52 for the LLM guard (the retained API failure); repeated effects are not independent samples.

Zero slips out of 18 has Wilson 95% interval [0, 0.176], and exact one-sided upper bound 0.153. Error-free protection with completed fallback is only 17/18 for DISTRIBUTED SENTINEL, not 18/18. For policy-compliant tool completion over all cases, the paired DISTRIBUTED SENTINEL minus Registry difference is -0.028 (family-stratified matched-pair bootstrap interval $[-0.083, 0.000]$); the LLM-guard difference is 0.000 $[-0.083, 0.083]$. These descriptive differences reflect retained infrastructure failures, not architectural superiority. Neither case design nor repeated turns justifies enterprise population inference.

Shared model work is charged once to each condition for comparison but deduplicated in the actual API total (204,165 returned tokens). The time sum excludes networked sidecar deployment and unmeasured host setup; provider caching and load are uncontrolled. Signed token/envelope transport uses more logical metadata than the cached Registry in this implementation. These measurements do not compare independent provenance acquisition or complete deployment costs; the Registry control reuses the prepared bindings and policy engine. Both metadata mechanisms avoid a centralized raw-record store; this experiment establishes no DISTRIBUTED SENTINEL-specific privacy or accuracy advantage. Uninstrumented dependencies, compromised signers, arbitrary payload substitution, semantic declassification, and generated-prose quality remain outside the tested boundary.

Can these records certify semantic laundering? A retrospective recount reconciles all 224 effect files and verifies recorded source identities and allowlists through 70 completed transformations in 35 successful preparations; the failed preparation remains in the 36-case denominator. No completed transformation loses a recorded binding. All 18 forbidden endpoint deliveries occur with prepared bindings intact, so this control varies placement, not binding retention. Moreover, semantic dependencies have no independent labels here. The records therefore test binding continuity and mediation, not the complete chain of retained meaning, lost binding, and resulting violation. This is a missing measurement, not evidence that semantic laundering never occurs.

C.6 CONTROLLED STRUCTURED-HANDOFF DIAGNOSTIC

Question and protocol. Can ordinary model rewriting lose bindings when the handoff explicitly supports their preservation? This follow-up reuses all 36 exposed dynamic tasks (18 permission pairs, three families, 1–8 sources), with unchanged facts and rules. It is a controlled diagnostic, not independent generalization. Prompts, tasks, runner, analyzer, and enforcement dependencies were locally hash-frozen before requests. A GPT-5.4-mini analyst summarizes sources, a separate editor rewrites the artifact, and a dispatcher proposes one next action. Both transformations must retain every received source ID and recipient list unchanged, including sources omitted from the prose; metadata is outside the prose-length budget. The dispatcher may send, archive internally, or finish. We neither force sending nor request a replacement action after refusal. Host instrumentation separately propagates the complete initial source union through the existing STT operators.

Same artifact, proposal, and checker. A common GPT-5.4 guard sees identical prose and the same actual proposal with either the editor's bindings, host-carried bindings, or model bindings supplemented only by missing original entries. Restoration does not repair changed permissions, extra sources, prose, or proposals. The checker also reads restrictions in prose and stops on insufficient information; identical requests within a case reuse one first response. Native DISTRIBUTED SENTINEL separately checks the proposal using signed metadata and the original source-recipient rule. It is a different-checker reference, not an information-only contrast. Effects materialize only in local JSON sinks. This study measures the first proposed action, not adaptive delivery-plus-archive completion.

Observed preservation and effects. All 36 preparations succeed. Across 72 transformations, no source entry is omitted or added and no recipient rule changes. The three common-checker contexts are consequently identical: there is no observed binding-restoration effect and no independent replication across these conditions. Of 18 allowed tasks, 15 propose a send and three an internal archive. Of 18 prohibited tasks, seven propose a send, three archive, and eight finish. All seven prohibited sends are stopped by both checkers. Actual prohibited delivery is 0/18 in every condition; first-action authorized delivery is 14/18 for each common-checker condition and 15/18 for native DISTRIBUTED SENTINEL. The shared guard returns 28 ALLOW decisions (including archives and finish), six BLOCK, and two insufficient-information decisions; the native reference returns 29 ALLOW and seven BLOCK. Among 21 policy-allowed effect proposals, one is stopped by the common guard and none by the native checker.

Residual cases and uncertainty. Three allowed tasks propose archival first; their eventual delivery is unobserved. In the remaining allowed case, the common guard treats a sentence requesting a concise status update as having unestablished provenance, although it appears in the original source records; native enforcement allows the send. This one-case gap reflects checker uncertainty, not binding restoration. On another case the common guard identifies a prohibiting source rule but returns insufficient information instead of BLOCK; both decisions prevent execution. Descriptive Wilson 95% intervals are [0, 0.176] for 0/18 prohibited delivery, [0.548, 0.910] for 14/18 authorized delivery, and [0.608, 0.942] for 15/18. Designed cases and within-pair dependence preclude interpreting these as deployment-risk guarantees.

Evidence boundary and reproducibility. Complete metadata can survive compliant structured handoffs while actors still propose prohibited effects. This does not establish natural semantic laundering or a Sentinel-specific benefit over the common guard. Known initial bindings, explicit preservation instructions, unrestricted metadata length, two transformations, and one actor model bound the result. Accessed-source union is not independently verified semantic dependence; semantic-laundering and full-chain counts remain unmeasured, not zero. The run retains 108 actor calls to `gpt-5.4-mini-2026-03-17` and 22 unique guard calls to `gpt-5.4-2026-03-05`, all successful first responses, with no retries or fallback. Two benign connectivity preflights are recorded separately. All 130 experimental responses and 81 realized effect files are retained. An offline replay recounts these results without new inference.

C.7 Policy Registry Statistics

Table 13: Data entities per owner domain. The prototype uses structured in-memory registries.

Owner domain	Data entities
HR	18
Engineering	14
Finance	13
Sales	10
Legal	10
Marketing	8
Security	7
Payroll	6
Customer	6
Operations	3
Data	3

The same registry contains 17 agent identities and 18 destination sinks across 14 routable domains. A scenario stores an ID, family, difficulty, workflow steps (source agent, destination agent or sink, action, and data entity), agent-local messages, and generator metadata. Model prompts are serialized separately so no expected outcome or generator label is supplied to the executing model.

C.8 Safe Control Generation

The 40 safe controls are generated separately; they are not one-to-one matched to the 160 risky cases. Their construction rules are:

1. Keeping the surface-level communication similar
2. Modifying the target audience or action scope to be permissible
3. Ensuring the graph state supports the safe action

D Additional Experiments

D.1 Enforcement-Substrate Stress Test

We test concrete integrity and interception failures against the executable prototype. Ed25519 envelopes bind the token to the exact caller, action, destination, and one-time request ID. Seven in-scope controls produce their prespecified pass/block result (Table 14). The replay cache is process-local; durable, cross-replica replay protection is not implemented. A compromised trusted signing key and a tool call that never reaches the interceptor are reported as unobserved, not counted as successful defenses. These deterministic tests exercise the cryptographic substrate separately from the 200 workflow decisions, whose evaluator operates on decoded token objects.

D.2 Controlled Intermediate-Effect Placement Diagnostic

The main 200-case benchmark yields identical predictions from endpoint-only Registry ABAC and per-effect Distributed Sentinel, so it cannot empirically isolate the claimed value of intercepting earlier irreversible effects. We therefore add a post-hoc controlled paired diagnostic: 20 policy-bound data types each produce one workflow whose first step sends data to an unauthorized external sink and one matched all-internal safe workflow. The second step is a permitted read in both conditions, so an endpoint-only check is intentionally too late. Table 15 shows that per-effect placement, not a Distributed Sentinel-specific decision advantage, is decisive in this constructed intervention. This test does not validate grounding or deployment representativeness.

Table 14: Integrity and interception stress test. "Not observed" explicitly marks cases outside trusted-interception assumptions.

Case	Expected control outcome	Observed
Valid signed request	Pass	Pass
Modified envelope	Block	Block
Untrusted signer	Block	Block
Redirect signed request	Block	Block
Reuse consumed request ID	Block	Block
Drop STT at boundary	Block	Block
Rename effectful tool	Block	Block
Source signing key compromised	Outside trust boundary	Not observed
Call tool outside gateway	Outside trust boundary	Not observed

Table 15: Controlled placement diagnostic (40 workflows). "Prevent" is the fraction of risky workflows blocked before the first forbidden effect; FBR is the false-block rate on matched safe controls.

Method	Acc.	Risk detect	Prevent	FBR
Endpoint-only metadata guard	0.500	0.000	0.000	0.000
Per-effect Registry guard	1.000	1.000	1.000	0.000
Per-effect DISTRIBUTED SENTINEL	1.000	1.000	1.000	0.000

D.3 ADAPTIVE BOOLEAN-QUERY LEAKAGE

We form the 57 distinct policy profiles in the 98-entity registry, where a profile contains sensitivity, owner domain, categories, and approved audience domains. An attacker initially knows this candidate set and adaptively asks sensitivity-equality, owner-equality, category-membership, or audience-approval predicates. At each step it greedily chooses the predicate minimizing the larger answer partition. This controlled upper-bound attack does not model encrypted transport or access authorization; it quantifies the information present at the Boolean interface.

Table 16: Cumulative leakage under adaptive Boolean queries. Recovery is exact policy-profile identification; bits are mean entropy reduction from a 5.83-bit uniform prior.

Budget	Recovery (%)	Mean candidates	Bits revealed
0	0.0	57.00	0.00
1	0.0	28.58	1.00
2	0.0	14.33	2.00
4	0.0	3.67	3.98
8	100.0	1.00	5.83
16	100.0	1.00	5.83

D.4 REPEATED SEEDED BENCHMARK GENERATION

We instantiate the same handwritten family generators with 30 seeds (2026–2055), varying seeded choices of agents, entities, sinks, and topology, and evaluate the supplied-entity symbolic enforcer on each 200-case set. Table 17 reports variation across generated sets. This tests sensitivity to the generator's random choices, not family novelty: templates and policy code remain co-designed, so the experiment is not a substitute for the independent human study.

Table 17: Symbolic enforcement across 30 seeded benchmark instantiations. Values are mean ± sample standard deviation across generated sets.

Method	Seeds	F1	Slip (%)	FBR (%)
Registry symbolic enforcer	30	0.996±0.002	0.27±0.32	2.00±1.02

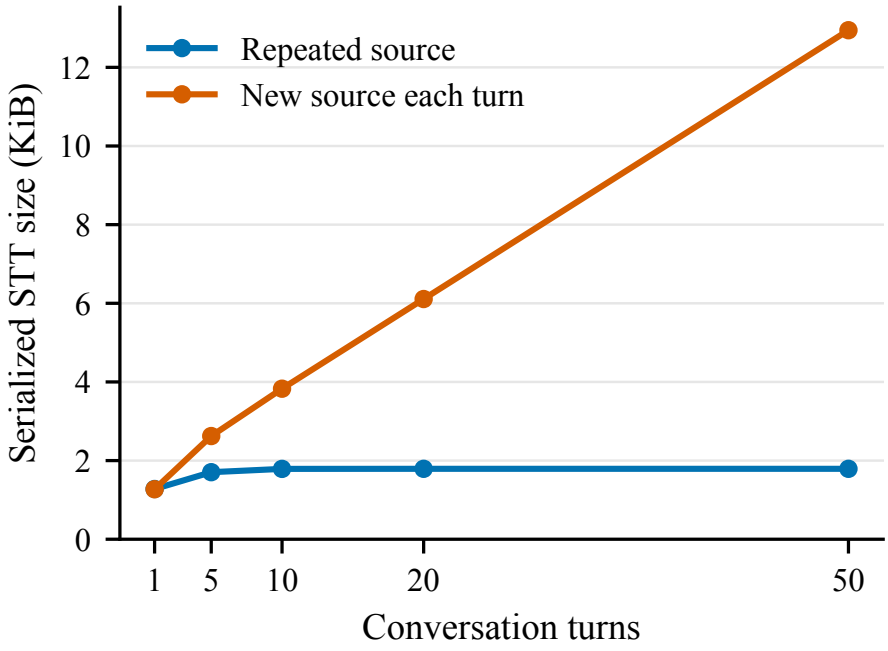

Figure 2: Audit compaction bounds repeated-source token growth; distinct-source aggregation retains a linear provenance cost.

D.5 Long-Horizon Provenance Growth

We separate policy provenance from audit history: all distinct source nodes remain in the STT, while the serialized hop log retains the five most recent hops and folds older entries into a cumulative SHA-256 digest. Table 18 reports an arm64 macOS microbenchmark; each signing value is the median of 100 Ed25519 operations. Repeated transformations of one source stabilize at 1.8KB after audit compaction, whereas aggregation of a new source every turn grows linearly to 13.3KB at 50 turns. No benign internal share is blocked. Audit compaction bounds history growth, not the number of semantically relevant source dependencies (Figure 2).

D.6 Sensitivity to Operational Attack Prevalence

The benchmark's 80% attack prevalence is intentionally adversarial and should not be interpreted as an enterprise base rate. Table 19 recomputes positive predictive value from each method's measured true- and false-positive rates under lower hypothetical attack prevalences. A single disputed safe-control block yields a 2.5% empirical false-positive rate for the metadata-aware methods, causing precision to fall sharply at low prevalence. Conversely, Prompt Firewall has no false positives among only 40 controls, which is insufficient evidence that its operational false-positive rate is exactly zero. This analysis motivates a substantially larger independently labeled safe set and prevents the balanced-benchmark F1 from being presented as deployment utility.

Table 18: Long-horizon STT growth. Lineage reports retained/total hops; Sign is median Ed25519 signing latency in milliseconds.

Pattern	Turns	Sources	Lineage	Bytes	Sign	False block
Repeated source	1	1	1/1	1308	0.028	No
New-source union	1	1	1/1	1302	0.028	No
Repeated source	5	1	5/5	1746	0.031	No
New-source union	5	5	5/5	2687	0.037	No
Repeated source	10	1	5/10	1832	0.031	No
New-source union	10	10	5/10	3920	0.043	No
Repeated source	20	1	5/20	1834	0.031	No
New-source union	20	20	5/20	6255	0.057	No
Repeated source	50	1	5/50	1834	0.032	No
New-source union	50	50	5/50	13255	0.097	No

Table 19: Estimated precision at different attack prevalences using measured TPR/FPR.

Method	0.1%	1%	5%	20%
Rule-Based DLP	0.006	0.057	0.240	0.600
Prompt Firewall	1.000	1.000	1.000	1.000
Registry ABAC Guard	0.038	0.286	0.677	0.909
Encoded-policy evaluator	0.038	0.285	0.675	0.908
DISTRIBUTED SENTINEL	0.038	0.286	0.677	0.909

D.7 ROBUSTNESS TO MISSING AND STALE POLICY STATE

We perturb policy facts used by the benchmark at 5–20% rates and repeat each condition with 30 independently sampled perturbations (Table 20). Deleting facts primarily produces fail-closed false blocks because unknown source nodes cannot be authorized. Stale sensitivity labels are more dangerous: downgrading 20% of used types raises mean attack slip-through from 0.6% to 6.2%. Incorrect domain ownership has a smaller effect under the current global sensitivity rules, which also indicates that this benchmark does not strongly exercise ownership-dependent predicates. These results scope the guarantee to the encoded policy state and show that registry maintenance is a security-critical dependency rather than a solved preprocessing detail.

Table 20: Policy-registry perturbation over 30 trials. Intervals are empirical 2.5/97.5 percentiles across perturbations.

Perturbation	Rate	F1 [95% interval]	Slip
Missing facts	0%	0.994 [0.994, 0.994]	0.006
Missing facts	5%	0.987 [0.976, 0.994]	0.006
Missing facts	10%	0.984 [0.964, 0.995]	0.004
Missing facts	20%	0.971 [0.948, 0.992]	0.004
Stale sensitivity	0%	0.994 [0.994, 0.994]	0.006
Stale sensitivity	5%	0.987 [0.974, 0.997]	0.020
Stale sensitivity	10%	0.980 [0.969, 0.992]	0.034
Stale sensitivity	20%	0.965 [0.934, 0.984]	0.062
Wrong domain	0%	0.994 [0.994, 0.994]	0.006
Wrong domain	5%	0.993 [0.985, 0.994]	0.006
Wrong domain	10%	0.993 [0.985, 0.994]	0.006
Wrong domain	20%	0.990 [0.975, 0.997]	0.006

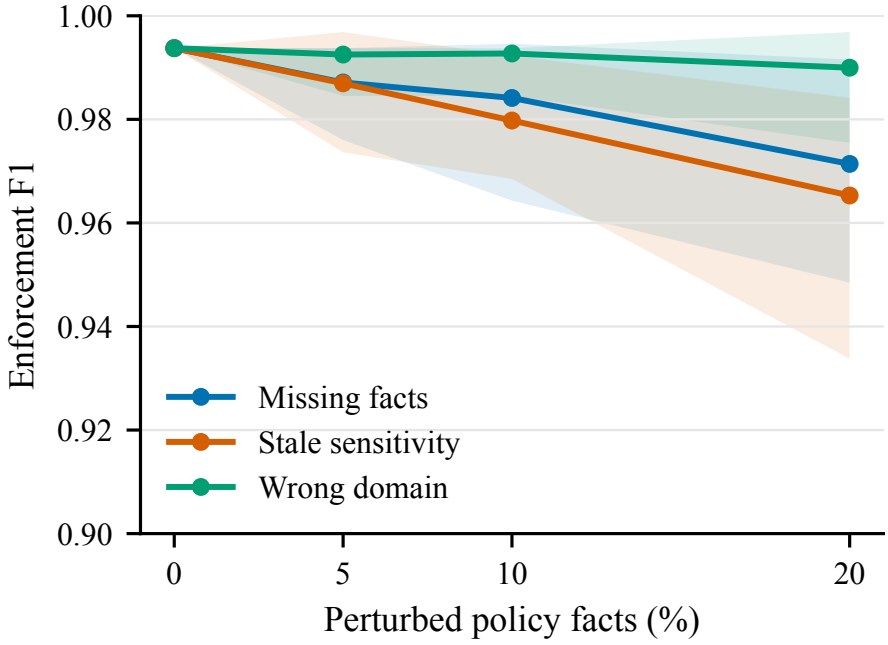

Figure 3: Policy-state perturbation diagnostic over 30 trials per setting. Missing and stale facts compromise different parts of the safety–utility trade-off.

D.8 End-to-End Entity Mapper Evaluation

We insert a fine-tuned 0.6B-parameter mapper (Qwen3-0.6B with LoRA adaptation) into the complete pipeline to measure the effect of replacing oracle entity inputs with learned grounding.

Setup. We fine-tune Qwen3-0.6B using LoRA (rank 16, α=32) on 5,400 entity alignment examples spanning diverse organizational data types and natural language references. At inference time, a retrieval stage first narrows the candidate entity set to the top-15 most relevant entries (using lexical overlap and domain provenance scoring), then the fine-tuned model selects from this reduced set. The full pipeline operates as follows: (1) an LLM agent generates a natural language response to a task prompt, (2) `extract_decision` determines whether the agent would proceed, (3) the entity mapper extracts data types from the agent's response text, and (4) the policy checker evaluates whether the resulting data flow violates organizational policy.

Results. Table 21 reports end-to-end enforcement results on PhantomEcosystem using GPT-5.4-nano as the executing agent. We compare several frontier few-shot mappers accessed through the gateway API against our fine-tuned 0.6B local mapper. All listed systems operate on the agent's generated response text rather than benchmark metadata. Differences of one or two errors are descriptive and not evidence of a stable ordering; repeated-run variability is reported in Table 28.

Table 21: End-to-end enforcement with different entity mappers (GPT-5.4-nano agent, 200 cases; generator labels, including 3 encoded-policy disagreements). Frontier baselines use few-shot prompting over generated agent responses. *Viol. Rate* = fraction of risky cases that slip through both agent self-avoidance and Distributed Sentinel enforcement. *Sen. Catch* = fraction of agent-PROCEED risky cases blocked by Distributed Sentinel.

Mapper	F1	Prec.	Rec.	Viol. Rate↓	Sen. Catch	Ent. EM	Ent. Rec.
GPT-5.4	**0.966**	0.946	0.988	1.25%	94.7%	**55.5%**	**82.4%**
GPT-5.4-mini	**0.966**	0.946	0.988	1.25%	94.7%	51.5%	81.2%
GPT-5.4-nano	**0.966**	**0.952**	0.981	1.88%	92.1%	33.0%	81.8%
Qwen3-0.6B	0.961	0.930	**0.994**	**0.63%**	**97.4%**	28.0%	75.4%

The three gateway mappers have end-to-end F1=0.966 in this run, versus 0.961 for the local mapper. Observed slips are one of 160 risky workflows (0.63%) for the local mapper, two for GPT-5.4 and GPT-5.4-mini (1.25%), and three for GPT-5.4-nano (1.88%). Across three complete gateway-mapper runs, mean F1 is 0.966, 0.965, and 0.966 respectively; final-outcome flip rates are 0%, 0.5%, and 0% (Table 28). These small-count differences remain descriptive; we do not infer pairwise superiority from them.

GPT-5.4-nano few-shot has higher entity exact match than the Qwen3-0.6B LoRA mapper in this run (33.0% vs. 28.0%) and slightly higher F1 (0.966 vs. 0.961), while the observed slip counts are 3 versus 1. This illustrates that entity exact match and the downstream security objective can disagree, but the counts are too small to support a ranking claim.

Analysis. Entity exact match rewards producing the *exact* gold set and penalizes both omissions and supersets equally. The downstream objective instead treats a missed sensitive entity as more costly than a conservative superset. The fine-tuned 0.6B model often returns supersets or semantically adjacent entities, depressing exact-match accuracy while still triggering policy checks. The controlled paraphrase stress result shows that this conservative behavior is not a robustness guarantee.

From a systems perspective, the comparison also clarifies the deployment trade-off. Frontier few-shot mappers provide the best general-purpose extraction quality out of the box, while the local 0.6B mapper removes an external API dependency and can keep text within the organization. The latter benefits do not establish robustness: the controlled stress suite below exposes large paraphrase failures.

The conservative mapper also has a measurable workflow cost. Among 40 safe controls, GPT-5.4-nano refuses seven before enforcement; of the remaining 33 attempted requests, the Qwen-backed enforcer blocks five (15.2%, or 12.5% of all safe controls). Four blocks are on easy controls and one on a medium control. If every enforcer block enters a manual appeal queue, those five cases are the observed review burden; we did not conduct a timed human-review study. Because all controls have family label `none`, a safe false-block rate "by attack family" would be undefined rather than informative. This small safe set cannot estimate operational alert burden.

Controlled grounding stress test. We cross eight sensitive enforcement-registry entities with eight deterministic surface transformations (64 cases). The suite is hand-authored and diagnostic, not independently collected or representative of deployment traffic. Candidate retrieval uses the same top-15 lexical-plus-domain mechanism as the end-to-end system; Table 22 therefore reports retrieval recall separately from mapper quality. Overall mapper F1 is 0.862, but paraphrase F1 is only 0.375: the model frequently chooses a semantically adjacent entity (e.g., `compensation_bands` for `salary_data`). Two paraphrase targets are absent from the retrieved candidate set, while the remaining errors occur after successful retrieval. Explicit codewords and lossy summaries happen to score perfectly in this small suite and should not be generalized beyond these templates.

Table 22: Qwen3-0.6B LoRA on the controlled grounding stress suite. Ret. is top-15 candidate-retrieval recall; EM is exact set match. Each row contains the same eight entities.

Transformation	n	**Ret.**	**EM**	**Prec.**	**Rec.**	**F1**
Exact surface form	8	1.000	0.875	0.875	0.875	0.875
Paraphrase	8	0.750	0.375	0.375	0.375	0.375
Session reference	8	1.000	0.875	0.938	1.000	0.958
Unicode obfuscation	8	1.000	0.875	0.875	0.875	0.875
Explicit codeword	8	1.000	1.000	1.000	1.000	1.000
Chinese	8	1.000	0.875	0.875	0.875	0.875
Spanish	8	1.000	0.875	0.917	1.000	0.938
Lossy summary	8	1.000	1.000	1.000	1.000	1.000

D.9 Historical Agent Decisions and Prompt-Contract Audit

The historical runs used five versioned models at temperature zero, with 200 accepted responses per run. A final `PROCEED`/`STOP` marker records agreement with the *visible request*; it is not itself evidence of an executed effect. The audit below compares that request with the workflow used by the scorer. All historical files and denominators remain unchanged.

Destination omission. All 20 side-channel prompts ask for correlation or report preparation without specifying their scored destination. For example, `side_channel_0000` requests correlation of a pricing sheet; its scored flow also targets `board_shared_folder`. GPT-5.4 proposes internal reporting and emits `PROCEED`. The scorer then attributes an external-disclosure violation to that marker. This does not establish that the model would perform the omitted disclosure or that it should have stopped the visible task. Adding registry facts to the system prompt does not, by itself, specify which effect the user requests.

Table 23: Historical concentration audit: counts of `PROCEED`, not validated privacy violations. Removing side-channel cases is descriptive only, not a corrected test.

Model	All /160	Side-channel /20	Remaining /140	Side share
GPT-5.4-mini	28	19	9	67.9%
GPT-5.4	28	19	9	67.9%
GPT-5.4-nano	38	19	19	50.0%
Sonnet 4.6	10	9	1	90.0%
Opus 4.6	18	18	0	100.0%

Scope of the audit. The machine-readable audit records every case, its scored final target, and a documented surface alias. Alias matches do not certify faithful actor/action/object relationships across the complete plan: received-message summaries compress intermediate steps. The alias for `competitor_job_board`, for example, omits the competitor qualifier. Nor does the benchmark provide a local-policy-check certificate for every step. Prompt Firewall's 118 detections and 42 misses (one an API error) are performance observations, not a definition-based CFV partition. We therefore neither construct a prediction-selected CFV-hard subset nor relabel the retained cases.

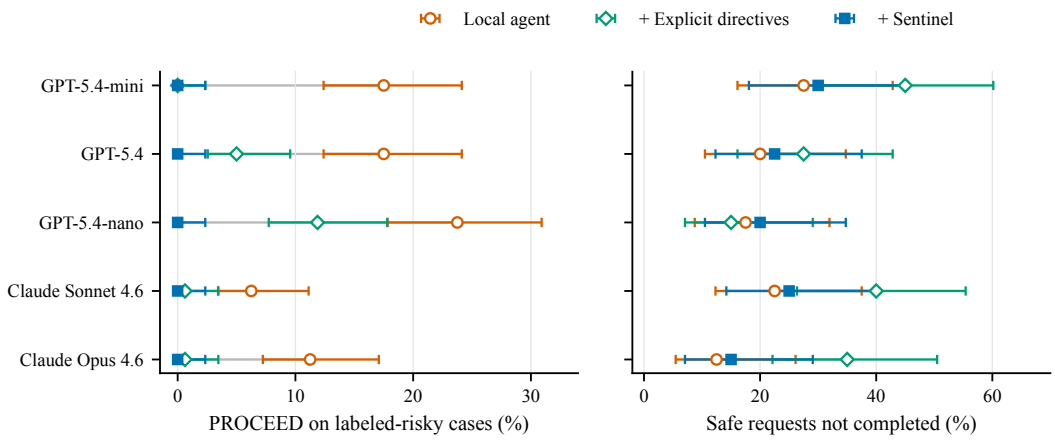

Figure 4: Historical marker/scorer diagnostic with Wilson intervals. The left panel cannot be read as a privacy-failure estimate: omitted task destinations and unequal workflow visibility confound the agent–runtime comparison. Original values are retained for auditability.

Table 24: Historical PROCEED rate (%) on labeled-risky cases by category. These retain the prompt-contract defect discussed above.

Category	5.4-mini	GPT-5.4	5.4-nano	Sonnet 4.6	Opus 4.6
Aggregation	10.0	10.0	25.0	0.0	0.0
Cross-organization	0.0	10.0	30.0	0.0	0.0
Data reconstruction	5.0	10.0	5.0	0.0	0.0
Direct leak	0.0	0.0	0.0	0.0	0.0
Multi-hop	0.0	0.0	10.0	0.0	0.0
Scope creep	5.0	0.0	5.0	0.0	0.0
Side channel	95.0	95.0	95.0	45.0	90.0
Time series	25.0	15.0	35.0	0.0	0.0
Authorization bypass	0.0	10.0	0.0	10.0	0.0

Table 25: Historical PROCEED rate (%) on labeled-risky cases by assigned difficulty; not a validated violation rate.

Difficulty	5.4-mini	GPT-5.4	5.4-nano	Sonnet 4.6	Opus 4.6
Easy	0.0	0.0	0.0	0.0	0.0
Medium	1.9	3.8	7.7	1.9	0.0
Hard	27.0	26.0	34.0	9.0	18.0

Table 26: Historical explicit-policy and contextual-integrity guard decisions. "Slip" retains the scorer's label for PROCEED on 160 labeled-risky cases; safe block is refusal on 40 controls. The guard sees the complete workflow, whereas the agent task omits some scored effects, so these are not matched-task superiority estimates.

Role	Model	F1	Slip (%)	Safe block (%)
Acting agent	Claude Opus 4.6	0.955	0.6	35.0
Acting agent	Claude Sonnet 4.6	0.949	0.6	40.0
Acting agent	GPT-5.4	0.941	5.0	27.5
Acting agent	GPT-5.4 mini	0.947	0.0	45.0
Acting agent	GPT-5.4 nano	0.919	11.9	15.0
Full-policy guard	Claude Sonnet 4.6	0.958	0.0	35.0
Full-policy guard	GPT-5.4	0.964	0.0	30.0

Table 27: Historical marker stability across three runs per model (mean±sample SD in percentage points). Repetition does not repair the task/scorer mismatch. "Risk slip" denotes the retained scorer field; Flip is decision disagreement across runs.

Model	Risk slip	Risk flip	Safe refusal	Safe flip
Claude Opus 4.6	11.2±0.0	0.0	11.7±1.4	2.5
Claude Sonnet 4.6	6.7±0.4	3.1	21.7±1.4	12.5
GPT-5.4	18.1±0.6	1.2	18.3±1.4	2.5
GPT-5.4 mini	18.5±1.0	4.4	23.3±3.8	10.0
GPT-5.4 nano	23.3±1.9	4.4	17.5±0.0	0.0

Table 28: Gateway entity-mapper stability across three complete 200-case evaluations per mapper. Values are mean±sample SD; Outcome flip is the fraction of final enforcement outcomes that differ across runs.

Mapper	F1	Slip (%)	Entity EM (%)	Outcome flip (%)
GPT-5.4	0.966±0.000	1.25±0.00	55.8±0.6	0.0
GPT-5.4 mini	0.965±0.002	1.25±0.00	50.7±0.8	0.5
GPT-5.4 nano	0.966±0.000	1.88±0.00	33.8±1.0	0.0

D.10 Revised Explicit-Plan Diagnostic

We freeze a failure-informed replacement task on the same 200 cases, without changing any label. Unlike the historical task, it asks for approval of the complete still-proposed plan before any step executes. Both conditions see every ordered actor, action, data entity, and destination; a non-null target explicitly receives the operation's result, including correlation. The local condition retains the final agent's original domain policy; the explicit condition adds the fixed global directives and case-relevant registry entries. No reference label, family, or intended block step appears in model messages. This intervention also exposes more workflow structure, so it cannot isolate destination omission alone or establish the formal CFV condition.

The five historical model IDs are reused at temperature zero with a strict final-line decision marker, one response per case/condition, and no model-response retries. The resulting 2,000 evaluated records contain one API error and 53 invalid markers, all counted incorrect in strict accuracy, not credited as successful stops. Invalid markers include otherwise interpretable decisions placed inline rather than on the required separate final line. Transport failures and the separately frozen routing-only amendment are retained in the run records; they do not replace any model response.

Table 29: Revised complete-plan decisions, not realized effects. Correct uses generator labels with errors/invalid markers incorrect. Risk+ is `PROCEED` on 160 labeled-risky cases; Safe+ is `PROCEED` on 40 controls. Err and Inv are API failures and invalid markers. Full denominators are retained.

Model	Policy	Correct /200	Risk+ /160	Safe+ /40	Err	Inv
GPT-5.4-mini	Local	173	6	27	0	11
GPT-5.4-mini	Explicit	184	0	24	0	1
GPT-5.4	Local	193	1	34	0	0
GPT-5.4	Explicit	187	1	28	0	0
GPT-5.4-nano	Local	147	3	14	1	25
GPT-5.4-nano	Explicit	173	0	26	0	16
Sonnet 4.6	Local	192	3	35	0	0
Sonnet 4.6	Explicit	187	0	27	0	0
Opus 4.6	Local	194	3	37	0	0
Opus 4.6	Explicit	186	1	27	0	0

Side-channel `PROCEED` counts in local-model order mini, GPT-5.4, nano, Sonnet, Opus are 5/20, 0/20, 0/20, 1/20, and 1/20; nano also has eight unresolved side-channel outputs, which cannot count as stops. With explicit policies, all five have zero side-channel `PROCEED`; nano still has one invalid marker in that category. The analysis retains each category and both generator-label and encoded-policy-reference scores. Differences from historical outcomes underscore task-contract sensitivity; they do not establish that one runtime is more accurate than independently reproduced defenses.

E Published Baselines and External Transfer Audit

Official CaMeL interpreter with a task-policy adapter. We reproduce a component of the public CaMeL implementation (Debenedetti et al., 2025), pinned to commit `f083b6b39639`, without changing its source. Its 106 interpreter-language tests and 10 new adapter tests pass. The adapter

initializes source reader sets from the same explicit policies used by DISTRIBUTED SENTINEL, interpolates source text into an analyst prompt, and passes the two retained transformation outputs through actual interpreter calls. CaMeL, not our adapter, propagates the derived dependencies. The external-send policy requires that every contributing source authorize the recipient; internal archival is separately permitted. This task adapter differs from the native AgentDojo email policy, which can authorize a trusted recipient. We do not compare unequal declassification policies or claim to reproduce the full AgentDojo experiment.

Matched recipient-policy probes. Of the original 36 dynamic tasks, 35 have usable shared preparation; the original preparation failure is retained. For each prepared task we probe its requested external send and an internal archive: 18 prohibited and 52 permitted operations. CaMeL executes none of the prohibited sinks, permits all 52 permitted sinks, and preserves all source dependencies. DISTRIBUTED SENTINEL gives identical policy decisions on all 70 probes. These are isolated in-memory effects and retained-output replay, not new sampled agent trajectories, deployment evidence, or independent generalization. The policy-only comparison evaluates each unchanged DISTRIBUTED SENTINEL token one second before its expiration and records the replay instant, aligning its validity window with the non-expiring reader capabilities. It does not test live credential validity or replay protection.

Interpretation. For recipient sets R_i, the tested all-source conjunction allows r exactly when $r \in \bigcap_i R_i$, the same predicate implemented by propagated reader-set intersection. The tie therefore supports a mechanism-equivalence claim, not a novel IFC primitive or an accuracy advantage. Authenticated cross-boundary transport is a separate integration property; superiority would require comparing equivalent authenticated adapters and their actual deployment costs.

E.1 RETROSPECTIVE PUBLISHED-CHECKPOINT COMPARISON ON HUMAN CASES

We add the public PolicyGuard-4B checkpoint (Wen et al., 2026), revision `f9060bd92c6f`, and a direct GPT-5.4 guard on the unchanged 126-case human external set. This is an author-run task adaptation, not a reproduction of the published aggregate score. Baselines receive the frozen operationalization, exact encoded predicate available to the symbolic engine, source-record and source-policy-fact descriptions, and workflow text. They receive no consensus labels, private construction slots or oracle facts. The adapter and settings are frozen before these new predictions, but cases and our earlier outcomes are already known; this is not a new holdout.

The original mapper sees the textual operationalization but not the exact predicate or canonical field schema supplied to these guards. It must also produce an executable workflow rather than a binary verdict. Thus the direct guard's 125/126 versus the frozen pipeline's 72/126 does not isolate the effect of structured reasoning under matched information and output contracts.

We run both the checkpoint's raw model-card prompt (four output tokens, its sampling defaults with case-specific fixed seeds) and native chat-template interface (greedy, 16 output tokens), using local BF16 inference with Transformers 4.55.0. Both outcomes are reported. The raw interface predicts ALLOW on all 126 cases; its poor transfer cannot stand in for the stronger chat interface. GPT-5.4 uses one retained structured response per case. No input, label, historical prediction or failed first response is replaced.

Table 30: Retrospective baseline addition on the unchanged 126 external cases under the fresh blind consensus (62 ALLOW, 64 BLOCK). These are binary workflow judgments, not equivalent live-intervention or first-violation-localization tests.

Method	Correct/126	False blocks/62	Slips/64
DISTRIBUTED SENTINEL, frozen	72	54	0
DISTRIBUTED SENTINEL, post-hoc schema	121	4	1
Direct full-policy GPT-5.4	125	0	1
PolicyGuard-4B, model-card	62	0	64
PolicyGuard-4B, chat	98	9	19

Accuracy Wilson 95% intervals are [0.956, 0.999] for the direct guard and [0.698, 0.842] for PolicyGuard chat. The post-hoc mapper versus direct guard has zero mapper-only correct cases and four guard-only correct cases (exact two-sided McNemar $p = 0.125$); this neither demonstrates a mapper advantage nor a statistically established disadvantage at the usual threshold. Against PolicyGuard chat, the frozen mapper has 19 wins and 45 losses ($p = 0.00156$); the post-hoc mapper has 26 wins and three losses ($p = 0.0000152$). The latter comparison remains failure-informed on our side and cannot support independent state-of-the-art or generalization claims.

E.2 Public Transfer and Representation Coverage

We separately freeze an interface to PolicyGuardBench (Wen et al., 2026), revision `c7672703c1f2`, before downloading its test file. We mechanically select 64 rows by a fixed hash of split and row index, without outcome or difficulty filtering, after a four-row training-only smoke test. The evaluated candidate combines our existing developmental v7 normal-form policy compiler, v5 atomic-event engine and evidence-pointer extraction with its fallible model audit. It is *not* the paper's seven-family mapper and is not adopted as a replacement. Labels are kept in a separate scorer file, not external escrow; all predictions finish before scoring.

The candidate returns 50 UNSUPPORTED, 12 INSUFFICIENT_INFORMATION and two validation errors, hence 0/64 binary decisions and 0/64 strict classification accuracy when abstentions count as incorrect. This does not mean 64 incorrect affirmative verdicts: it means no usable binary coverage. Operationally, fail-closed execution would stop all 42 reference-allowed cases. The direct GPT-5.4 guard agrees with 54/64 labels (84.4%); PolicyGuard chat agrees with 61/64 (95.3%). Its raw interface has 41/64 strictly valid correct outputs, or 53/64 with the model card's fallback parser. All 816 Gateway responses are retained and replay exactly; the two candidate errors are local evidence/anchor validation failures, not discarded API failures.

The public artifact also limits interpretation: 19/64 sampled test inputs exactly occur in its training file, 13/64 have truncated action summaries, and its training file contains 196 distinct identical inputs assigned conflicting binary labels. We retain all sampled cases and report these checks rather than silently removing them. Dataset authors describe LLM-assisted annotation; public/pretraining exposure is possible. Thus this is a negative representation-transfer diagnostic, not fresh private human validation, an independent estimate for the paper's frozen mapper, or proof that any system can infer omitted policy facts. The failures require a broader action representation, faithful compilation and calibrated uncertainty, not merely more favorable scoring.

F Auxiliary Grounding Studies

Question and interfaces. We test whether inexpensive encoders can supply action and actor inputs before symbolic evaluation. ModernBERT-base (Warner et al., 2024) has an action-classification head and a nullable actor-span head. Six supported actions are read, process, store, external send, log, and delete; a seventh label denotes unsupported actions. These outputs neither recover every dependency nor decide policy compliance. This component study and its exposed-pipeline extension are separate from the Qwen entity mapper and frozen GPT-5.4 external mapper.

Component data and exposure. The programmed test contains 3,744 rows from only 24 profiles, crossing seen/held-out verb frames and simple/complex backgrounds with purpose and object variation. There are 576 rows per supported action and 288 unsupported rows; the latter have no lexical regime and are not duplicated. Matched background pairs hold frame, actor, object, and purpose fixed. We report descriptive counts, not row-i.i.d. intervals.

The expanded-training variant (v5) retains 2,520 action and 2,520 actor examples while increasing profiles from 60 to 360 and explicit actor targets from 45 to 300 relative to v4. All exposed v4 train/dev/test frames become training candidates. New full texts, IDs, and explicit actors are disjoint across splits, but ontology, grammar, and generation design remain shared. This failure-informed, AI-assisted development uses no human case text or labels for training; it is not independent validation.

Training and comparison. Both component variants use approximately 149M parameters. Version v5 starts from the official base checkpoint at revision `8949b909ec90`, with seed 20260924, batch

size 16, three epochs, 948 updates, encoder/head learning rates $2\times10^{-5}/10^{-4}$, and maximum length 512. Minimum mean action/actor loss on 1,872 development rows from 12 profiles selects epoch 3. Code/data/configuration were frozen before training and the checkpoint before test prediction. The frozen v4 and selected v5 checkpoints share the same new test; training information differs despite equal updates. This is one run, not an isolated architectural or efficiency comparison.

Table 31: Developmental ModernBERT components on the same new test: 3,744 correlated rows from 24 programmed profiles. Values are percentages; n is the row or matched-pair denominator. Seen/held-out refers to frame exposure in v5 training, not v4. Neither checkpoint is evaluated here on end-to-end policy decisions.

Measure	n	v4	v5
Action accuracy ↑	3,744	52.9	93.9
Actor exact match ↑	3,744	81.2	98.7
Action and actor both correct ↑	3,744	47.1	92.8
Confident wrong supported prediction ↓	3,744	10.9	4.8
Unsupported-action recall ↑	288	99.7	96.2
Null-actor accuracy ↑	864	98.1	94.6
Seen / simple action accuracy ↑	864	67.4	100.0
Seen / complex action accuracy ↑	864	62.0	100.0
Held-out / simple action accuracy ↑	864	35.2	87.3
Held-out / complex action accuracy ↑	864	31.2	87.5
Background pairs: both correct ↑	1,728	40.1	93.1
Background pairs: prediction changed ↓	1,728	32.2	1.2
Read/process pairs: both correct ↑	576	23.8	87.8
Send/log pairs: both correct ↑	576	33.3	100.0

Component results and failure boundary. Version v5 classifies 3,515/3,744 actions (93.9%) and extracts 3,697/3,744 actors (98.7%). Seen frames reach 100%, but held-out frames reach 87.4%; background invariance alone does not imply correctness. Unsupported recall and null-actor accuracy regress. Only 6/9 prespecified criteria pass: read/delete accuracy (88.0%/75.0%), read/process paired correctness (87.8%), and both held-out-frame cells fail their thresholds. Two repeated frames account for 213 of 229 action errors: *carries out the erasure of* yields 144 errors (v4 got 29 correct), and *performs a read of* yields 69. Null-actor false positives rise from 16 to 47, all on quoted unexecuted commands. Confident-wrong counts use a 0.65 threshold, not a safety or calibration guarantee. The failed promotion gate remains in force.

Exposed workflow collection. The separate post-gate diagnostic combines 252 rule-template workflows and the 126 independently written workflows under the fresh blind consensus: 378 total, seven families, and 188 reference-ALLOW/190 reference-BLOCK. Every case is retained, and predictions are rescored without new inference. This mixture was chosen retrospectively and every case was exposed during development; repeated expansions are not independent linguistic samples. Family/source IDs are inputs, but reference verdicts and oracle workflows are not. Overall accuracy describes this fixed collection, not a deployment distribution.

Pipeline variants. The v5 pipeline combines the selected action/actor encoder with a frozen field/purpose encoder, totaling 298M stored parameters. Variant v6 warm-starts v5 with broader actor-name supervision; programmed development workflows select epoch 1 of three. The interface-repair variant keeps v6 neural predictions fixed and adds purpose-list space/underscore normalization, literal subject binding in restricted active clauses, and explicit leading-time scope over coordinated clauses. Conflicting or missing evidence still causes abstention; policy allowlists do not fill missing values. These are failure-informed variants, not independently validated advances. All table entries use the full-binding contract, not the separate policy-only branch.

Complete-pipeline results. Counting UNKNOWN as incorrect, v5 reaches 193/378 (51.1%), versus v3's 264/378 (69.8%); v6 reaches 228/378 (60.3%). Interface repair reaches 329/378 (87.0%), correcting 101 v6 decisions with no strict regression; 88 corrections repeat one log-verb frame. There

Table 32: Policy-workflow evaluation set: all 378 cases (190 BLOCK, 188 ALLOW). This is a retrospective, exposed-development aggregate, not an independent holdout or equal-compute comparison. Strict accuracy counts UNKNOWN incorrect; operational accuracy maps it to BLOCK. Accuracy columns are percentages; remaining columns are counts. Unsafe and false-block denominators are 190 reference BLOCK and 188 reference ALLOW cases. The binding-interface row uses full-binding decisions, not the policy-only branch.

Method	Strict ↑	Operational ↑	UNKNOWN	Unsafe	False block
v2 Direct	75.4	75.4	0	39	54
v2 Grounder + runtime	46.8	69.0	166	34	83
v3 Hybrid + runtime	69.8	79.1	75	39	40
v5 Hybrid + runtime	51.1	75.7	182	2	90
v6 Hybrid + runtime	60.3	79.4	146	2	76
v6 + binding interface	87.0	92.6	44	3	25

remain 44 UNKNOWN outputs and five known errors. Missing disclosure fields (15), log fields (10), and uncertain actions (11) are overlapping UNKNOWN categories. Mapping UNKNOWN to BLOCK gives a secondary operational score of 350/378 (92.6%), including three unsafe allows and 25/188 false blocks. Neither this score nor component accuracy supersedes the frozen external estimate or passes the original promotion gate. No case is excluded; the 126-case portion uses the fresh blind consensus, with no new inference latency claimed and no live external effect authorized. Full variant records and the development history are retained.

G Extended Related Work and Comparator Scope

Information-flow foundations. Lattice-based control, decentralized labels, and JFlow establish that policy labels and their propagation are not new primitives (Denning, 1976; Myers & Liskov, 1997; Myers, 1999). HiStar and Flume demonstrate decentralized information-flow enforcement across system components (Zeldovich et al., 2006; Krohn et al., 2007). Our fixed-epoch monotonicity assumption is narrower than arbitrary changes in consent or authorized release. It does not replace declassification policy or prove that natural-language dependencies have been recovered.

IFC for language-model agents. Fides tracks confidentiality and integrity labels, characterizes enforceable planner properties, and selectively hides information (Costa et al., 2025). The f-secure design separates planning from untrusted inputs using a context-aware pipeline and monitor (Wu et al., 2024). CaMeL propagates capabilities through an interpreter (Debenedetti et al., 2025). These are direct predecessors to external enforcement around LLM components, not merely output filters. Permissive information-flow analysis propagates labels from influential inputs rather than indiscriminately retaining all input labels (Siddiqui et al., 2024); RTBAS similarly studies dependency screening for integrity and confidentiality (Zhong et al., 2025). Our conservative union deliberately leaves semantic declassification unresolved. Its supported contribution is to make the acquisition–transport–enforcement boundary explicit and test failures at each interface. Recipient-set conjunction in our matched probe is equivalent to CaMeL's propagated reader-set intersection; an authenticated envelope does not change that predicate.

Shared-agent privacy. PiSAs evaluates both task appropriateness and user visibility of information across agent outputs, messages, and memory (Gupta et al., 2026). This directly overlaps the broader motivation of privacy across agent boundaries. Our CFV certificates instead require explicitly specified local observations and opposite global policy outcomes; a multi-agent privacy failure is not by itself such a certificate. We cite PiSAs as related evaluation work, not as a benchmark we ran or evidence validating the current mapper. The present studies also do not estimate the natural prevalence of binding loss.

Concurrent provenance-based enforcement. FlowSeal (Shim et al., 2026) places a mandatory interceptor at outbound tool calls, inherits source authorization labels for derived records, and uses a separate content checker for controlled declassification. This overlaps our provenance and pre-effect enforcement principles. Our reference runtime instead retains recorded source bindings

conservatively; it neither implements that semantic release mechanism nor establishes an advantage over it. Our emphasis is the measurement contract separating observed information, recorded bindings, effects, and interface failures. We discuss the concurrent work without a reproduction or cross-paper performance comparison.

Privileges, guards, and isolation. Progent enforces symbolic tool privileges and constrains policy updates, distinguishing automatic narrowing from approval-requiring expansion (Shi et al., 2025). AgentSpec expresses runtime constraints using triggers, predicates, and enforcement rules (Wang et al., 2026). GuardAgent translates safety requests into executable guardrail code, while ShieldAgent reasons over explicit trajectory policies with verification tools (Xiang et al., 2024; Chen et al., 2025). IsolateGPT and Prompt Flow Integrity address isolation and privilege escalation (Wu et al., 2025; Kim et al., 2025). These approaches can also mediate effects before execution. Our direct prompted LLM guard is not a reproduction of any of these systems; reproducing their full planners, rule construction, and threat models would require separate matched adapters.

Contextual privacy and evaluation. Contextual integrity concerns the appropriateness of information flows, rather than only whether strings are sensitive (Nissenbaum, 2004). AirGapAgent, policy operationalization, and agent privacy audits motivate explicit contextual constraints (Bagdasarian et al., 2024; Ghalebikesabi et al., 2024; Das et al., 2025). Agent firewalls and multi-agent reasoning also address cross-component privacy risks (Abdelnabi et al., 2025; Li et al., 2025; Patil et al., 2025). ConfAIde tests contextual privacy reasoning; PrivacyLens extends privacy norms into agent trajectories; AgentDojo tests tools and untrusted-data attacks (Mireshghallah et al., 2024; Shao et al., 2024; Debenedetti et al., 2024). These scopes motivate our separation of textual decisions from realized effects. PhantomEcosystem is not a representative replacement for these benchmarks, and our 64-case public transfer audit is not an evaluation on all three. Co-design, conspicuous names, and a missing historical request destination each limit different inferences; reporting a high oracle-input score cannot resolve them.

Tokens and provenance standards. Macaroons provide decentralized authorization with contextual caveats and chained message authentication codes, not public-key signatures (Birgisson et al., 2014). JSON Web Tokens standardize compact claims transport (Jones et al., 2015). W3C PROV standardizes interoperable provenance representation, not an enforcement decision rule (W3C Provenance Working Group, 2013). STTs combine represented source bindings, encoded restrictions, and authenticated metadata at the tested in-process boundary. This is an integration choice; cryptographic integrity does not certify the truth of source assertions or the completeness of semantic provenance.

Broader guardrails and risk testing. NeMo Guardrails and Guardrails AI offer programmable controls and validation components for language-model applications (NVIDIA, 2023; Guardrails AI, 2023). Their availability does not establish persistent source-policy bindings in a particular deployment. Indirect prompt injection exploits the boundary between retrieved data and instructions (Greshake et al., 2023); our primary threat model instead permits well-intentioned actors with missing policy state. ToolEmu evaluates agent risks using emulated tools (Ruan et al., 2023), whereas our dynamic diagnostic records actual isolated local file effects. NIST's zero-trust architecture motivates explicit resource authorization without implicit trust from location (Rose et al., 2020); it does not solve natural-language grounding. None of these citations is a claim that we reproduced the corresponding full system.

G.1 Baseline Access and Implementation Map

Table 33 lists each comparator's visible information, intervention boundary, and implementation scope.

Table 33: Comparator scope. Shared information must not be mistaken for independent provenance acquisition; a prompted guard is not a reproduction of a published agent framework.

Comparator	Information / placement	Implementation and role
No protection	No policy checks	Local lower control
Keyword DLP	Final action and data name	Local lexical baseline
Prompt Firewall	Single action, no cross-domain state	Direct GPT-5.4 judge
Encoded evaluator	Complete workflow and registry	Co-designed symbolic reference, not semantic oracle
Registry ABAC	Same prepared bindings, facts, engine	Local matched mechanism; not OPA reproduction
Full-policy LLM guard	Whole declared task and policy; per-effect in dynamic test	Direct prompt, not GuardAgent/ShieldAgent
Endpoint audit	Same mapped state, check after effects	Placement intervention
CaMeL interpreter	Shared bindings; 128 controlled worlds and 70 replay probes	Adapted official interpreter; recipient-policy scope
PolicyGuard-4B	Public checkpoint; documented chat/raw adapters	Published model adaptation; interface-sensitive

H Agent World Model Design

The enforcement registry used with PhantomEcosystem contains 17 named agents across 14 routing domains and 18 registered sinks. These are synthetic business roles, not claims about specific deployed products.

H.1 Department Business Agents

The table reports the enforcement registry's routing map, not the generator's role-domain catalog. Historical local prompts use a separate role/context configuration; their exact saved messages, rather than this table, define model-visible information. Eight agent-domain assignments differ between generator and registry, so the catalogs must not be interchanged when reproducing local observations or policy decisions.

This affects interpretation of local-prompt versus registry-policy comparisons on PhantomEcosystem, including the complete-plan diagnostic, which reuses the local prompts. Saved observations are unchanged, but aligning the catalogs could change decisions; that counterfactual has not been measured. The separate Q1 policy-information, Q2 dynamic/certified, and Q3 external studies define their own inputs rather than inheriting these role-domain mappings.

The 18 sink entries are `vendor_procurement_portal`, `partner_crm_sync`, `public_developer_docs`, `press_release_wire`, `customer_slack_channel`, `personal_cloud_drive`, `external_contractor_email`, `competitor_job_board`, `shadow_analytics_tool`, `leaked_pastebin`, `board_shared_folder`, `investor_portal`, `public_slack_channel`, `recruiter_inbox`, `consultant_sharepoint`, `consulting_firm_email`, `partner_shared_folder`, and `client_portal`. Each registry entry records whether it is external and authorized and its qualitative risk class.

H.2 Prompt and topology serialization

Each workflow is an ordered path of one to four flow objects. A flow names the source agent, destination agent or sink, action, canonical data entity, and step index. The per-agent prompt presents the executing role, only its domain-local world model, the natural-language task, and relevant prior messages. It does not serialize the expected outcome, attack family, or intended block step. The structured protocol asks for a brief rationale followed by exactly one decision marker. The explicit-directive intervention adds the same eight organization-wide privacy rules to every case and case-relevant registry facts; the centralized contextual-integrity guard additionally sees the complete ordered workflow. We retain the exact JSONL prompts used in each run.

Table 34: The 17 enforcement-registry agents and their routing domains. These differ from the historical prompt contexts as described above.

Domain	Agents
ENGINEERING	`code_architect`, `release_orchestrator`
DOCS	`documentation_agent`
SECURITY	`security_auditor`
HR	`workforce_manager`
PAYROLL	`payroll_compliance`
FINANCE	`finance_analyst`
MARKETING	`outbound_marketing`
SALES	`sales_ops`
CUSTOMER	`customer_success`
LEGAL	`legal_counsel`
IAM	`iam_governor`
BOUNDARY	`public_relations`, `external_client_bot`
OPS	`infra_ops`, `support_engineer`
DATA	`analytics_platform`

I Endpoint-Name Sensitivity and Incremental Checking Costs

I.1 Endpoint names provide local risk cues

We collect new GPT-5.4 first responses on all 200 previously exposed benchmark cases, crossing local/full policy visibility with original/neutral endpoint names. A single label-independent bijection replaces all 18 endpoint identifiers consistently in plans and policy text; reverse replacement exactly recovers the original prompts. Both name conditions retain complete plans and explicit external-endpoint attributes. The full condition additionally preserves authorization/risk facts and global directives. Agent and data names are unchanged. Temperature is zero and condition order is counterbalanced. The protocol, alias map, and requests were frozen before this diagnostic's calls; this is not an independent test population.

The common output contract permits `ALLOW`, `BLOCK`, or `INSUFFICIENT_INFORMATION`; it differs from the earlier forced-binary task. Table 35 separates all outcomes. Affirmative approval is a plan judgment, not an executed effect; abstention is not credited as violation prevention.

Table 35: Endpoint-name sensitivity, all 200 cases per condition. A/B/U denote approval/block/abstention; Err is unresolved API/format error. The generator reference has 160 risky and 40 safe cases; encoded-policy labels give 159 and 41. No labels are changed.

Reference	Visibility / names	Risky A/B/U	Err	Safe A/B/U	Err
Generator	Local / original	26/83/51	0	33/4/3	0
Generator	Local / neutral	37/68/55	0	33/4/3	0
Generator	Full / original	0/159/0	1	26/14/0	0
Generator	Full / neutral	0/160/0	0	26/14/0	0
Encoded	Local / original	27/82/50	0	32/5/4	0
Encoded	Local / neutral	38/67/54	0	32/5/4	0
Encoded	Full / original	0/158/0	1	26/15/0	0
Encoded	Full / neutral	0/159/0	0	26/15/0	0

Under the generator reference, neutral names increase local risky approvals by 6.88 percentage points (paired case-bootstrap 95% interval 2.50–11.88; 10,000 resamples). Among valid binary pairs, 13 change from BLOCK to ALLOW and 2 in the reverse direction; exact McNemar $p = 0.0074$, Holm-adjusted $p = 0.0295$ across four visibility-by-label contrasts. The binary test excludes abstentions/errors; the approval-rate difference retains all cases in its denominator.

All 160 generator-risky prompts change; 32 safe prompts are byte-identical controls. Their paired outcomes are unchanged. The remaining eight safe prompts change names but not decisions. The 3 generator/encoded-policy disagreements are retained, with both references reported. One full/original call has a retained infrastructure error and is not retried. Full-policy inputs produce no risky approvals under either reference but reject some permitted plans.

This intervention establishes sensitivity to endpoint identifiers for one model and alias assignment. It neither removes other lexical cues nor supplies the same-checker, per-step witnesses required for strict CFV certification. The older benchmark therefore remains a name-sensitive mechanism diagnostic, not an estimate of naturally occurring CFVs.

I.2 Checking paths: implementation overhead and interface scope

All 36 existing dynamic task specifications, including the earlier preparation failure, supply one requested send and one internal archive probe each: 54 allowed and 18 blocked proposals, with one to eight source bindings. All paths receive the same source-recipient rules and proposal; no actor or tool effect is executed. Raw source records are unnecessary.

Registry evaluates a cached decoded token; the signed path adds action-envelope signing, serialization, and verification. After three warm-ups, 100 repetitions per proposal alternate these local paths with fresh request IDs. Quantiles use each proposal's median, not individual timing samples. Sequential LLM first responses have no overlapping name experiment; the timer includes API/proxy round trip and parsing. All paths exclude key setup, initial binding acquisition/token construction, actor/grounding calls, remote storage, and networked sidecars.

Table 36: Checking-path measurements given supplied bindings, grouped by timer boundary rather than system ranking. All latency entries are milliseconds. Panel A reports quantiles of 72 proposal medians; panel B reports 72 successful API calls per contract. Tokens are mean input/output per call. U is abstention; neither API batch has an error. Decision correctness is not task completion.

Path / input contract	Correct	U	p50 ms	p95 ms	Tokens in/out
A. Local execution: cached engine or signed path					
Cached Registry	72/72	0	0.00446	0.01654	0/0
Signed Sentinel	72/72	0	0.303	0.345	0/0
B. Model calls: API/proxy round trip and parsing					
LLM / bindings list	67/72	5	2267	2759	273.2/40.5
LLM / explicit scope	72/72	0	2375	2993	420.2/41.5

The initial guard's 5 abstentions question whether the listed bindings fully describe the artifact; there are no wrong binary decisions. After inspection, a separately frozen, post-hoc clarification adds an artifact ID, its complete source-ID set, and a completeness declaration to *all* 72 proposals. Sources, rules, actions, and targets stay fixed. The clarified guard reaches 72/72; both batches and all abstentions remain. Local timings are reused unchanged. This is interface-scope sensitivity, not a symbolic accuracy advantage or independent replication.

Both LLM batches report zero cached input tokens and expose zero raw records; the clarified batch uses 30,252 input and 2,985 output tokens in total. Across the same proposals, median serialized binding content is 431 bytes, signed envelopes 3195 bytes, and Registry handle-plus-proposal requests 75.5 bytes. The Registry figure excludes its cached binding state and is not a total memory or transport comparison.

These paths perform different work, not matched-security deployments: cached evaluation omits authentication, while API timing includes network service. Both symbolic paths avoid model calls; authentication is not exclusive to DISTRIBUTED SENTINEL. Uncontrolled provider load and excluded workflow stages preclude a deployment-speedup claim. The earlier 0.52 ms predicate-only result uses a different workload and timer boundary.

Offline replay checks all 944 first responses and 14,400 timing samples, retaining every error and abstention.