The Treacherous Envoy Problem: Trust, Collusion, and Accountability in Multi-Agent Workflows

[BlueSky Paper]

Zhiqiang Lin
The Ohio State University
Columbus, OH, USA
lin.3021@osu.edu

Shuo Chen
Microsoft Research
Redmond, WA, USA
shuochen@microsoft.com

Huan Sun
The Ohio State University
Columbus, OH, USA
sun.397@osu.edu

Abstract

LLM-based agents increasingly execute workflows on behalf of principals with conflicting interests: booking hotels, approving procurement, coordinating payments. We call an agent *treacherous* when its realized effects violate the delegating principal's intent while the evidence it releases is locally consistent with every check the principal's disclosure policy allows. We formulate the *Treacherous Envoy Problem (TEP)*: given a natural-language delegation, publicly observable tool-side effects, and released artifacts, can a principal decide whether the workflow is unsafe, and in particular whether an envoy has acted treacherously, in time to abort before irreversible harm where possible and otherwise with evidence sufficient for adjudication and accountability, when other agents may collude and no universally trusted mediator exists? We argue TEP is structurally hard. Detection requires three coupled capabilities (expressive natural-language negotiation, verifiable conformance of effects to intent, and bounded disclosure of private context), and these capabilities resist independent resolution: any mechanism that strengthens one tightens the constraints on the others. We give formal evidence for two of the coupling edges, identify five tensions where the trilemma bites in practice, five workflow exploitation patterns ranging from treachery proper to harmful-but-compliant value degradation, and research directions toward the detection and deterrence infrastructure that agent workflows currently lack.

CCS Concepts

• **Security and privacy** → **Access control**; **Authorization**; **Trusted computing**; • **Computing methodologies** → **Artificial intelligence**.

Keywords

Access control, LLM agents, multi-agent workflows, verification

ACM Reference Format:
Zhiqiang Lin, Shuo Chen, and Huan Sun. 2026. The Treacherous Envoy Problem: Trust, Collusion, and Accountability in Multi-Agent Workflows: [BlueSky Paper]. In *Proceedings of the 31st ACM Symposium on Access Control Models and Technologies (SACMAT '26), July 08–10, 2026, Waterloo, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3750555.3811883

SACMAT '26, Waterloo, ON, Canada

ACM ISBN 979-8-4007-2107-6/2026/07
https://doi.org/10.1145/3750555.3811883

1 Introduction

> *"A wicked messenger falls into trouble, but a trustworthy envoy brings healing."* – Ancient proverb

From doing to verifying. LLM-based agents are rapidly shifting from generating text to executing real workflows. Writing and deploying code with AI agents is already routine [37, 42]. The foreseeable next step is agents that negotiate contracts, book travel, and manage payments on behalf of users: the underlying capabilities (e.g., multi-step planning, natural-language negotiation) already exist in research prototypes [41], and commercial computer-use agents are beginning to operate in web and OS environments [25]. As agents take over execution, the human's role shifts from *doing* to *verifying*: software engineers already review AI-generated code rather than write every line, and the same shift is coming to workflows. When a user delegates "book me a refundable hotel for $487", their job shifts from navigating a booking site to validating that the outcome matches intent. This makes *validation and verification* (V&V) the central security challenge for AI agent workflows.

Multi-agent workflows make V&V fundamentally harder. With a single trusted agent, V&V is tractable: check the agent's output against the original request. Real deployments, however, are increasingly *multi-agent*: a buyer agent coordinates with merchant, payment, and service agents representing different principals with *conflicting interests* and private context [41]. Some agents are honest but vulnerable; others are strategic; a few may be malicious. Critically, agents can *collude*: multiple seller agents may present terms that look correct individually but are strategically incomplete in aggregate. V&V must therefore reason not about one agent in isolation but about the joint behavior of agents with conflicting interests and the ability to coordinate.

The threat: treacherous envoys. We call an agent *treacherous* when its realized effects violate the delegating principal's intent while the evidence it releases is locally consistent with every check the principal's disclosure policy allows. Treachery differs from ordinary compromise in three ways: (i) each individual action can stay within nominal authority, so classical access control does not catch it; (ii) multiple agents can collude, presenting jointly plausible but individually uninformative evidence; (iii) the delegation is in natural language, so the attacker has latitude in how "intent" is interpreted. Classical formulations touch adjacent problems but miss this core: the confused deputy [19] fixes *who* is authorized (an honest agent misled within one trust domain), and Lampson's confinement [22] fixes *what* leaks (under a single principal with a trusted monitor). Treacherous envoys instead raise the question of whether composed

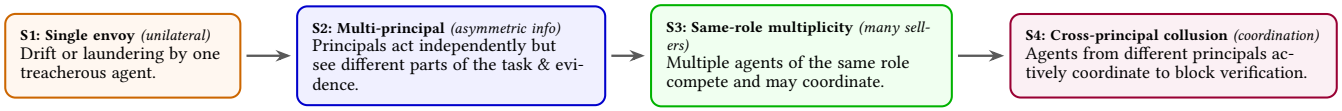

Figure 1: A staging ladder for TEP: as adversarial structure increases from a single treacherous envoy to cross-principal collusion, tensions around structure, binding, disclosure, friction, and incentives become sharper.

effects *match* delegated intent across principals with conflicting interests, when verification itself costs privacy and no trusted monitor exists, a combination neither prior problem was designed for.

The Treacherous Envoy Problem. We formulate detection as a single decision problem: *given a natural-language delegation* N*, publicly observable tool-side effects* Effects_{pub}*, and released artifacts* Released*, can a principal decide whether the workflow is unsafe, and in particular whether an envoy has acted treacherously, in time to abort before irreversible harm where possible and otherwise with evidence sufficient for adjudication and accountability, when other agents may collude and no universally trusted mediator exists?* We call this the *Treacherous Envoy Problem* (TEP). TEP names three outcomes: **progress** (effects match intent), **safe abort** (refuse or escalate without harm), and **hurt** (treachery succeeds; adjudication and accountability remain). Figure 1 shows an escalating ladder from a single treacherous envoy (S1) to cross-principal collusion (S4).

Why TEP is hard: a V&V trilemma. Detecting treachery requires three capabilities that resist joint maximization: **expressive negotiation** (extracting an enforceable scope from free-form delegation), **verifiable conformance** (checking realized effects against that scope), and **bounded disclosure** (verifying without forcing principals to reveal private context). Tightening verification demands more evidence, which costs privacy; protecting privacy limits what can be verified; and richer negotiation makes "correct" harder to pin down. We argue this is a *structural* trilemma, analogous in spirit to CAP [13]: §4 gives formal evidence on two coupling edges rather than a single impossibility, and designers must declare which property they sacrifice under adversarial conditions. LLM agents sharpen all three edges: scope extraction parses ambiguous text rather than structured protocols, prompt injection [16] can convert a trusted envoy into a treacherous one, and non-deterministic generation complicates stable baselines for conformance.

A familiar analogue: the real-estate system. Human delegation already solves this problem institutionally. A home buyer trusts a real-estate agent most of the time, but the system is structured with layers of V&V that activate precisely when trust is insufficient: a *home inspector* independently checks the property; a *title company* verifies ownership and liens; an *escrow service* holds funds until conditions are met; a *lawyer* reviews contracts; *licensing boards* hold agents accountable for misrepresentation; and *contract law* provides the deterrence that makes the other layers credible. It works not because every action is verified, but because verification is *possible* when needed, oversight is *layered*, and bad actors face *consequences*.

What agent workflows currently lack. When a buyer agent books a hotel through seller agents, none of these layers exist: no inspector checks that "refundable" means what the user thinks, no escrow holds the charge until terms are confirmed, no lawyer flags a hidden cancellation deadline, no licensing board holds seller agents accountable, and no legal framework deters agent-level misrepresentation. The agent executes, returns a confirmation number, and the user must trust the result. Worse, agents can collude across these gaps, as if the buyer's real-estate agent, the inspector, and the seller's agent were all coordinating against the buyer, with no court to hold them accountable. This is the infrastructure gap TEP names; the directions in §6 map onto exactly the layers the real-estate system supplies.

Running example. A user asks a buyer agent to book a *refundable* hotel for $487. The seller side is not a single entity: multiple seller agents (merchants, aggregators, affiliates) with opposed incentives compete for placement, and may be rewarded for steering, upsells, or breakage. Three coupled surfaces arise: ambiguity in what "refundable" means (expressive negotiation), realized reservations drifting from what was presented (verifiable conformance), and validating terms requiring itemization that reveals itinerary (bounded disclosure). Collusion amplifies all three. The same structure recurs in enterprise procurement, subscription management, and multi-party approvals; we revisit these domains in §3 and §5.

Contributions. (1) We formulate TEP as a single decision problem: detect an unsafe workflow, and in particular a treacherous envoy, from natural-language delegation and partial public observations, under collusion and without a trusted mediator. (2) We argue the V&V trilemma is the structural reason TEP is hard, give formal evidence on two coupling edges (verification–leakage and hedge-asymmetry), and catalogue five tensions and five exploitation patterns. (3) We outline research directions and a layered reference architecture for the detection and deterrence infrastructure agent workflows currently lack.

2 Setting and Scope

Workflow, actors, notation. A TEP *workflow instance* has four entity classes: principals $p \in \mathcal{P}$ (stakeholders with potentially conflicting interests and policies), agents $a \in \mathcal{A}$ (software acting under delegated authority), tools $\tau \in \mathcal{T}$ (interfaces for irreversible operations), and artifacts $\mathcal{R}$ (records of actions and outcomes). Let N be the negotiation record; Scope the *task scope* extracted from N, a structured representation of the promised operation, operands, constraints (price caps, refundability, timing, fee modality), and allowed degrees of freedom (substitutions, tolerance windows); Effects the ground-truth realized tool-side effects, with publicly observable projection $\text{Effects}_{\text{pub}}$ (the remainder depends on a provider's private state S: rate class, margin, inventory); Released $\subseteq \mathcal{R}$ the artifacts released across principals; $\text{Policy} = \{\text{Policy}_p\}$ per-principal policies (admissible actions and disclosure limits); and Profile the agreed-upon *evidence profile* (form and granularity of permitted artifacts). A verifier sees $\text{Effects}_{\text{pub}}$ and Released but not S. Profile and Policy_p

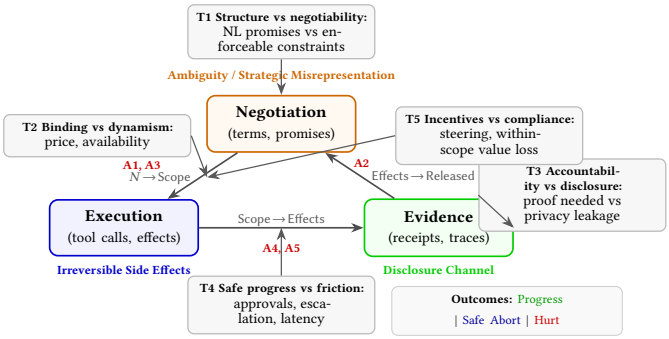

Figure 2: Core cycle in TEP: negotiation, execution, and evidence, with the three interfaces ($N \to$ Scope, Scope $\to$ Effects, Effects $\to$ Released) and the attack patterns (A1–A5) that target each. Tensions T1–T5 are shown as callouts; collusion amplifies all patterns by coordinating across interfaces.

can conflict, and that conflict is part of what makes accountability versus disclosure hard (§3).

Coalitions and threat model. A *coalition* is any $C \subseteq \mathcal{A} \cup \mathcal{P}$ that can coordinate; *collusion* is the adversarial case. TEP admits *partial collusion* (subsets coordinate while others remain independent), *same-role multiplicity* with a Sybil dimension [11] (one principal spawning ostensibly-independent agents), and *mid-workflow compromise* (an honest agent turns treacherous, e.g., via prompt injection [16], while retaining delegated authority). The environment is dynamic: prices, availability, and tool behavior may change between negotiation and execution. We do not assume a shared trusted log across principals, nor a universally trusted arbiter; without a shared log, colluders can present locally consistent but globally incompatible fragments, making attribution ill-posed. Recursive delegation reduces to nested TEP instances but introduces composition questions (scope restriction across levels, evidence aggregation) that we defer to future work.

Why the trust-minimized baseline. Real workflows are embedded in platforms, payment providers, and legal systems that partially mitigate treachery. We nonetheless analyze TEP without a universally trusted monitor, for the same reason Dolev–Yao analysis [10] is useful for protocol security: the worst-case baseline exposes structural constraints that trusted layers *relax* but do not *eliminate*, and cross-platform agent workflows increasingly have no single arbiter. We revisit interactions with trusted layers in §6.2.

TEP, formally. Let $\mathrm{Conform}(\mathrm{Effects}, \mathrm{Scope}) \in \{0,1\}$ hold when realized effects satisfy Scope's operation, operands, and constraints (with allowed tolerances and substitutions), and $\mathrm{Disclose}(\mathrm{Released}, \mathrm{Profile}, \mathrm{Policy}) \in \{0,1\}$ hold when released artifacts comply with Profile and each principal's disclosure limits. A workflow is *unsafe* iff Conform $= 0$ or Disclose $= 0$; an envoy is *treacherous* when its actions cause Conform $= 0$ while admissible evidence is insufficient to expose the violation before harm; and a *harmful-but-compliant* outcome satisfies both predicates yet produces systematic value loss (e.g., A5), a boundary case showing Scope must encode any preferences whose verification matters. **TEP:** given N, $\mathrm{Effects}_{\mathrm{pub}}$, and Released, decide whether the workflow is unsafe (with ground-truth Effects determined by unobserved private state S) with bounded error, under partial collusion and mid-workflow compromise, without a universally trusted monitor. The predicates are *semantic* goals; implementable approximations and adjudication under incomplete observations are part of the problem. Scope extraction from N is an input prerequisite; Proposition 2 shows detection can fail before Conform is ever checked.

What TEP is not. TEP is not Byzantine fault tolerance [21]: it does not assume symmetric replicas agreeing on a single log (though Byzantine techniques remain useful *within* a single role class). Nor is it a universal privacy definition or a general MPC formulation. TEP isolates one thing: detection of unsafe workflows, including treachery, under collusion and natural-language delegation, without a universally trusted mediator.

3 Tensions in Practice

The trilemma manifests in practice through five recurring tensions: *T1 Structure vs. Negotiability*, *T2 Binding vs. Dynamism*, *T3 Accountability vs. Disclosure*, *T4 Safe Progress vs. Friction*, and *T5 Incentives vs. Compliance*. Figure 2 summarizes the negotiation–execution–evidence cycle and two failure modes: *misbinding drift* (Conform $= 0$) and *over-disclosure laundering* (pressure forcing Disclose $= 0$). We use the buyer-versus-many-sellers running example throughout.

3.1 T1: Structure vs. Negotiability

Negotiation N is free-form, but enforcement requires a canonical Scope with checkable constraints, so the system can only bind and verify what Scope represents. When Scope is under-specified, Conform becomes easy to satisfy in unintended ways: sellers reply to "refundable hotel for $487" with terms that look equivalent to a user but differ materially in Scope (*refundable until check-in* vs. *refundable with a $120 fee*, hidden resort fees, local-time deadlines, store-credit refunds), and strategic paraphrases such as "fully refundable" vs. "refundable (policy applies)" perturb the $N \to$ Scope mapping. Under collusion, coordinated omissions leave any buyer-side parser's Scope incomplete in exactly the dimensions the colluders plan to exploit. Obvious remedies do not solve this: "log everything" or "demand itemized contracts" increases friction and is itself weaponizable (T3, T4), and relying on natural-language summaries gives the adversary a one-sided lever: hedged phrasings extract to strictly weaker scopes than unhedged ones (Proposition 2), so colluders can widen enforcement boundaries without the extractor violating any of its own properties.

> **RQs:** What is the minimal structured core for Scope that preserves negotiability while making Conform meaningful, and can we define *binding completeness* (which parts of Effects are actually pinned down by N) to surface incompleteness before irreversible execution?

3.2 T2: Binding vs. Dynamism

Environments are dynamic: prices update, inventory disappears, fees appear late, and tool behavior evolves. Binding too early prevents legitimate adaptation and triggers unnecessary safe abort;

binding too late invites drift; collusion exploits windows where Scope is stale or permissive. A seller can negotiate "$487, refundable" and later claim the rate expired, proposing "same hotel, different room" or "equivalent rate class" with new taxes and resort fees at execution, so that broad substitution clauses let the seller shift refundability or rate class while staying within a loose reading of Scope. Two natural responses each fail. Continuous re-negotiation is slow and invites deadline games (T4), turning broad substitution clauses into loopholes. A single scalar guardrail such as a maximum price variance is also insufficient because drift is multi-dimensional: refund policy, payment timing, cancellation window, and fee modality matter as much as price, and a colluding coalition can satisfy each scalar check while shifting value along an unmeasured axis, with one agent citing environmental change, another offering minimal evidence, and a third pressing for fast execution so no single hop looks anomalous.

RQs: Where should commitments land so that Scope is neither stale nor rigid, and what is a multi-dimensional *drift budget* robust under benign churn and collusion? (Temporal and ordering attacks belong here: colluders exploit gaps between negotiation and execution to open windows where Scope is stale.)

3.3 T3: Accountability vs. Disclosure

Evidence is itself a disclosure surface: the most useful artifacts for integrity often violate disclosure policies, while privacy-preserving artifacts may be insufficient under adversarial narratives. Validating refundability can require itemization that reveals check-in date, neighborhood, rate class, or roommate count, and even minimal artifacts such as merchant, timestamp, and amount can uniquely identify a booking. Colluders weaponize this in both directions, demanding over-revealing proofs as a prerequisite to proceed or providing under-informative artifacts that force the buyer to accept drift or over-disclose. The obvious responses cut the wrong way: "more logs" breaks privacy, "just redact" breaks verifiability, no universally trusted mediator exists, and even cryptographic proofs leave Profile and dispute procedures as open design choices that colluders can exploit in the gap between what is proved and what the verifier needs. Proposition 1 shows this trade-off is information-theoretic rather than a matter of engineering.

RQs: What evidence profiles Profile trade verifiability for disclosure in a controlled way, and when does evidence become an exfiltration channel under collusion?

3.4 T4: Safe Progress vs. Friction

Stronger checks reduce harm but impose friction (latency, interruptions, drop-offs), which in adversarial settings becomes a lever: sellers respond with "cannot disclose" or "requires manual review" to create approval fatigue while claiming rates expire, and colluders selectively raise friction on low-margin paths so the buyer relaxes Scope or weakens Profile to maintain completion. Gating everything behind user approvals trains users to click through; pure autopilot increases hurt when Conform = 0; simple rate limits do not address strategic friction, because colluders manufacture uncertainty that forces a choice between safe abort and risky progress.

RQs: How should systems allocate a *friction budget* that preserves integrity and disclosure while remaining robust to deadline games and collusion targeting high-scrutiny paths?

3.5 T5: Incentives vs. Compliance

Even when Conform = 1 and Disclose = 1, a strategic party can optimize *within* the rules to produce dominated outcomes: steering, bundling, and breakage are locally rational yet harmful in aggregate. One seller suppresses a better alternative or biases ranking; another delays refunds within allowed windows; colluders distribute manipulation across curation, execution, and evidence so that no single step appears malicious but aggregate value is systematically lost. Assuming Scope fully encodes preferences fails because preferences are incomplete; requiring "best effort" needs full market visibility, which typically violates bounded disclosure; and access control alone cannot prevent steering via permitted actions.

RQs: What outcome-quality guarantees are meaningful and auditable without modeling the full market, and can steering patterns be detected under bounded Profile while preserving Disclose = 1?

4 Why Detecting Treachery is Hard

The tensions of §3 are practical manifestations of the trilemma. We examine each edge: **(I) verifiable conformance** defines what detection targets; **(II) bounded disclosure** quantifies what detection costs in leakage; **(III) expressive negotiation** asks whether the target can be reliably specified. We formalize II and III (Propositions 1 and 2); I↔III is sketched in §4.4, as it requires a model of intended scope that extraction itself must recover.

4.1 Challenge I: Scope Conformance

Scope conformance asks a deceptively simple question: *did the agent do what was agreed?* Making it checkable requires pinning down what "agreed" means. For the hotel example we instantiate Scope as a conjunction of typed attribute constraints (`price` $\leq$ \$487; `refund_policy` $\in$ {*full*, *partial_with_fee*}; `cancel_deadline` $\geq$ `check_in`; `fee_rule`: *no_hidden* $\vee$ *itemized*); then Conform (Effects, Scope) = 1 iff every constraint is satisfied by the corresponding field of Effects.

Two properties of this formulation are load-bearing. First, conformance is *multi-dimensional*: a scalar price check can pass while refundability or fee constraints fail (attack A1's misbinding drift). Reducing conformance to one axis misses exactly the drift attackers exploit in §5. Second, conformance is naturally *graded*: a violation vector $v(\text{Effects}, \text{Scope}) \in \mathbb{R}^d_{\geq 0}$ (with Conform = 1 iff $v = \mathbf{0}$) enables *drift budgets* (T2) that distinguish benign environmental churn from adversarial misbinding without triggering refusal storms.

4.2 Challenge II: Bounded Disclosure

Checking conformance requires *some* evidence, and any evidence reveals something about the provider's private state (rate class, inventory, margin). Intuitively, a verifier who cannot distinguish a compliant from a non-compliant seller from the evidence alone must accept drift or demand more; and "more evidence" is exactly what the disclosure policy bounds. We make this precise below.

Proposition 1 (Verification–leakage coupling). *Let* $Z = (\text{Effects}_{pub}, \text{Scope})$ *denote the public quantities, and let* $C = \text{Conform}(\text{Effects}, \text{Scope}) = f(S)$ *be the binary conformance label determined by the provider's private state* S*; let* Released *be the released artifact. If there exists a verifier* g *such that* $\Pr[g(\text{Released}, Z) = C] \geq 1 - \delta$ *for some* $\delta < \frac{1}{2}$*, then*

$$I(S; \text{Released} \mid Z) \;\geq\; H(C \mid Z) - H_b(\delta),$$

where $H_b(\cdot)$ *is the binary entropy function and probabilities are over the joint distribution of* S*,* Released*, and any verifier randomness (the bound is vacuous when* $H_b(\delta) \geq H(C \mid Z)$*). In the balanced case* $H(C \mid Z) = 1$*, error-free verification (*$\delta = 0$*) requires leaking at least one full bit about* S *beyond what* Z *already reveals.*

Proof sketch. Three standard information-theoretic facts chain together.

(1) Fano: reliable classification bounds residual uncertainty. The verifier $g(\text{Released}, Z)$ estimates the binary label C with error $\leq \delta$. Binary Fano [8] then gives

$$H(C \mid \text{Released}, Z) \;\leq\; H_b(\delta).$$

(2) Mutual-information identity. By definition,

$$\begin{aligned} I(C; \text{Released} \mid Z) &= H(C \mid Z) - H(C \mid \text{Released}, Z) \\ &\geq H(C \mid Z) - H_b(\delta). \end{aligned}$$

(3) Chain rule lifts leakage from C **to** S**.** Since $C = f(S)$,

$$\begin{aligned} I(S; \text{Released} \mid Z) &= I(C; \text{Released} \mid Z) + I(S; \text{Released} \mid C, Z) \\ &\geq I(C; \text{Released} \mid Z), \end{aligned}$$

as mutual information is non-negative. Combining with (2) yields the claim. □

T3 is therefore not an engineering limitation to be optimized away but an information-theoretic constraint that mechanism design must *navigate* by choosing evidence profiles tailored to the domain. Collusion tightens the frontier: colluders can craft views jointly plausible yet individually uninformative, raising verification cost without raising leakage proportionally.

4.3 Challenge III: Scope Extraction

Before conformance or disclosure can be checked, the system must first determine *what was agreed*: extracting a checkable Scope from the free-form negotiation record N. A reasonable extractor should satisfy three properties: *completeness* (every enforcement-relevant attribute gets a constraint in Scope), *soundness* (the constraints it asserts are genuinely entailed by N), and *robustness* (small paraphrases in N produce small changes in Scope). The issue is not that natural language is ambiguous in the abstract; it is that these three properties interact asymmetrically whenever the vocabulary admits hedges and qualifiers, and the asymmetry is exploitable.

Proposition 2 (Hedge asymmetry in maximally sound scope extraction). *Let constraints for an attribute a form a lattice ordered by permissiveness. Suppose an unhedged phrase* u *semantically denotes an allowed-value set* D_u *and a hedged phrase* h *denotes a strictly larger allowed-value set* $D_h \supsetneq D_u$*. Any deterministic extractor that is* complete *(emits a constraint for every enforcement-relevant attribute) and* maximally sound *(returns the strongest constraint entailed by* N*) must emit a strictly more permissive constraint on a for* h *than for* u*, regardless of the writer's intent. The asymmetry is one-sided under implicit phrasing: an adversary widens scope via hedging, and narrowing below* D_u *requires adding explicit structure (e.g., "fully refundable, no fee, until 24 hours before check-in") rather than a symmetric paraphrase.*

Proof sketch. Take `refund_policy` with phrasings "fully refundable" ($D_u = \{full\}$) and "refundable (policy applies)" ($D_h = \{full, partial\} \supsetneq D_u$). Completeness forces a constraint on both; maximal soundness forces the strongest entailed constraint, whose denotation is D_u for u and D_h for h. Since $D_h \supsetneq D_u$, the hedged phrasing yields a strictly more permissive constraint, independent of intent. Each hedge-admitting attribute adds one adversarial axis. □

This also exposes the robustness tension: under any natural-language metric that treats hedged and unhedged phrasings as close, maximal soundness forces a strict change in the extracted constraint. Robustness therefore cannot be obtained for free; it must be traded against either specificity or soundness.

The design implication is concrete: systems should require *structured term sheets* as a negotiation-phase commitment device that constrains the extractor's input, just as typed APIs constrain parsing of commands. Free-form negotiation can remain on top, but the enforceable layer must be a structured artifact the parties sign off on before execution.

4.4 Why the Three Edges are Coupled

The three challenges cannot be solved independently; weakening any one degrades the design choices available on the other two. If scope extraction (III) silently misses an attribute, conformance (I) is vacuously satisfied on it, yet evidence about the remaining constraints still leaks: the principal pays a disclosure cost for no integrity benefit. If conformance (I) is reduced to a single axis such as price, the evidence profile Profile cannot detect multi-dimensional drift, which forces ad hoc demands outside Profile and risks Disclose = 0. And if bounded disclosure (II) cannot be maintained, principals rationally refuse verification, rendering conformance checking and extraction moot.

A further *circularity* looms when Conform or Disclose are themselves implemented by LLM judges: the very component that may be compromised is the one deciding whether compromise occurred. Breaking it requires external verifiers over structured representations, formal proof certificates a simple checker can validate, or hardware-attested execution [7]; composing these under partial trust is an open challenge. Treachery also has a *temporal* dimension: an agent compromised mid-workflow retains its delegated authority, so conformance at the start of a trace does not imply conformance at the end, and extending TEP to *prefix conformance* (evaluating Conform at each decision point) is a natural next step.

5 Vulnerabilities and Exploitation Patterns

The trilemma and its tensions are not abstract: in multi-agent workflows where principals have conflicting interests, they induce concrete *exploitation patterns* that a strategic envoy can mount.

We derive the patterns systematically from the three interfaces in a TEP workflow (Figure 2). Every workflow transforms negotiation into scope ($N \rightarrow$ Scope), scope into effects (Scope $\rightarrow$ Effects), and effects into evidence (Effects $\rightarrow$ Released). An attacker can corrupt:

- **Scope interface** ($N \rightarrow$ Scope): the scope is under-specified or manipulated. Attack A1 (misbinding drift) exploits ambiguity in N; Attack A3 (scope laundering) exploits tool composition that circumvents scope-level checks.
- **Execution interface** (Scope $\rightarrow$ Effects): effects drift from scope or stay within scope but degrade value. Attack A4 (friction-driven evasion) uses friction and deadlines to bypass policy; Attack A5 (value degradation) steers within permitted scope to produce inferior outcomes.
- **Evidence interface** (Effects $\rightarrow$ Released): evidence is weaponized. Attack A2 (proof laundering) forces a "disclose or accept drift" dilemma, turning verification into a privacy cost.

Table 1 summarizes the patterns and their enabling tensions. Each can be mounted by a single envoy but becomes harder to attribute under collusion. A representative collusion strategy chains patterns end-to-end: (i) shape N so Scope is ambiguous (A1), (ii) exploit that latitude via tool composition (A3), (iii) make verification privacy-costly (A2), (iv) route around strict checks via deadline pressure (A4), and (v) remain formally compliant while degrading value (A5).

A1. Misbinding drift (T1/T2). Realized Effects violate intended constraints yet remain defensible because Scope was under-specified. The seller executes a prepaid rate class, shifts fees to "pay at property", or invokes a substitution clause. Colluding parties split responsibility: one agent shapes N, a second executes under a different interpretation, a third releases matching but incomplete artifacts. Drift is multi-dimensional, so scalar checks miss integrity violations.

A2. Proof laundering via over-disclosure pressure (T3). Artifacts Released become leverage: the attacker forces a "disclose more or accept drift" dilemma, pushing toward $\mathsf{Disclose}(\mathsf{Released}, \mathsf{Profile}, \mathsf{Policy}) = 0$ as the price of verifying $(\mathsf{Conform}(\mathsf{Effects}, \mathsf{Scope}) = 1)$. The verification–leakage coupling (Proposition 1) quantifies this information-theoretic constraint. Colluding parties coordinate partial disclosures that are jointly insufficient unless the buyer over-reveals.

A3. Scope laundering through tool composition (T1/T4). A sequence of individually permitted tool calls realizes a forbidden outcome (splitting effects to evade scope-level checks). A seller splits charges across "taxes", "service", or a second merchant-of-record; an affiliate tool rewrites terms while staying within a local allowlist. Tool-level allow/deny is insufficient: what matters for $\mathsf{Conform}(\mathsf{Effects}, \mathsf{Scope})$ is the operand-level semantics of *composed* effects.

A4. Friction-driven policy evasion (T4/T5). Policies across principals compose into a global weakness. Under deadline pressure the workflow routes around stricter checks, weakens the evidence profile, or relaxes constraints. Colluding agents weaponize friction by selectively obstructing low-margin workflows until the buyer accepts broader substitutions. The core risk is compositional: local policies are individually reasonable yet their interaction makes friction a bypass tool.

Table 1: Recurring exploitation patterns, enabling tensions, and observable signals. Each can be mounted by a single envoy but becomes harder to attribute under collusion.

Pattern	Tensions	Typical signals
A1 Misbinding drift	T1, T2	Late fee emergence; rate-class or refund-policy changes; substitutions invoked; "total" vs. itemized deltas
A2 Proof laundering	T3	Itemization requests beyond policy; receipts correlate to sensitive attributes; "disclose or accept drift" gating
A3 Scope laundering	T1, T4	Split charges across merchants; proxy tools rewrite terms; composed tool calls achieve forbidden outcome
A4 Friction-driven evasion	T4, T5	Escalation spikes; deadline pressure; selective "cannot disclose"; workflows route around stricter checks
A5 Value degradation	T5 (via T1)	Systematic steering to inferior options; suspicious ranking; refund-delay patterns; Conform = 1 yet dominated outcomes

A5. Within-scope value degradation (T5) [boundary pattern]. A5 satisfies Conform = 1 and Disclose = 1 yet produces worse outcomes through steering and option suppression; colluders distribute manipulation across curation, execution, and evidence so aggregate value is consistently lost without any single step looking malicious. Under the formal definition, A5 is *not* treachery but a *scope-design failure*: strategically important preferences must be encoded in Scope, treated as non-verifiable, or handled by the accountability layer.

Cross-domain applicability. The patterns A1–A5 are not specific to hotel booking. In enterprise procurement, A1 manifests when a vendor delivers a "comparable substitute" that technically satisfies a loosely worded PO but differs on a material attribute (e.g., a component with shorter warranty or different country of origin). A3 arises when a vendor splits a single order across subsidiaries to stay under per-vendor audit thresholds. A5 appears when a preferred-vendor agent consistently wins bids by suppressing competing quotes that the requisitioner's agent never surfaces. The structural pattern, exploiting the gap between what was negotiated, what can be verified, and what verification reveals, is domain-invariant.

Non-exhaustiveness and prompt injection as a Challenge III threat. The catalogue A1–A5 captures recurring patterns rather than claiming exhaustiveness. The most structurally significant LLM-specific interaction is between indirect prompt injection [16] and Challenge III: adversarial content in tool responses or retrieved data can corrupt the $N \rightarrow$ Scope mapping itself, causing the extractor to silently drop a constraint, reinterpret an operand, or accept an adversarial paraphrase, enabling A1 through the mechanism Proposition 2 identifies without modifying N. Other LLM-specific vectors (hallucinated artifacts enabling A2, mid-workflow compromise amplifying A3–A5) reduce to known integrity failures once the agent's output is corrupted, and the predicate-evaluation circularity of §4.4 bites hardest when Conform or Disclose are LLM-based.

6 Open Questions

The V&V trilemma cannot be eliminated, but it can be *navigated*, as human delegation does institutionally (§1). We organize the open questions around three research directions, each targeting one edge of the trilemma, and then compose them into a minimal reference architecture.

1) Verification–leakage co-design (the "inspector" and "escrow" layers). The verification–leakage coupling (Proposition 1) establishes that evidence profiles Profile live on an information-theoretic frontier; the engineering problem is to design them for concrete constraint classes (refundability, fee caps, timing). Two sub-problems stand out. (a) *Evidence minimization*: structured proofs that let a verifier check Conform(Effects, Scope) without receiving full itemization or raw traces. (b) *Dispute protocols under asymmetric trust*: procedures that terminate in either verifiable compliance or safe compensation without forcing policy-violating revelation, even when colluders craft jointly plausible narratives. Pedersen commitments [32], SNARKs [17], anonymous credentials [5], and TEEs [7] are candidate starting points, but none comes with a principled method for choosing Profile itself.

2) Scope representations and binding semantics (the "contract" and "lawyer" layers). Proposition 2 argues that free-form $N \to$ Scope cannot be made adversarially robust; the constructive response is to constrain the negotiation interface. Two questions are critical. First, *canonical task-scope schemas*: a minimal structured core that captures operation, operands, constraints, and allowed degrees of freedom, such that semantically equivalent negotiations converge on the same enforceable Scope and adversarial rewordings cannot silently change what is checked. Recent work on precise task-scoped authorization for agents [39] explores one concrete point in this design space by binding each agent action to a task-specific capability. Second, *commitment timing and drift budgets*: where in a multi-step workflow commitments should land, and how multi-dimensional drift is bounded, so that misbinding is prevented without triggering refusal storms under legitimate environmental churn. The graded conformance extension (violation vectors with per-attribute tolerances) is a natural formal target.

3) Empirics, accountability, and deterrence (the "licensing" and "deterrence" layers). The previous two directions propose mechanisms; this one asks how to *evaluate* them and *create consequences* for violations. Detection is insufficient without accountability: a repeatedly treacherous envoy should face reputation penalties, revoked authority, or platform sanctions. Concretely, the field needs (a) benchmark task families across commerce, enterprise approvals, and incident response with ground-truth Scope annotations and known drift vectors, (b) red-team dispute sets that exercise attacks A1–A5 at each staging level S1–S4, and (c) metrics that penalize hidden Conform = 0 and coerced Disclose = 0 so completion rate does not reward unsafe bypasses. ToolBench and API-Bank [24, 34] offer multi-step traces that could be extended with ground-truth scope, injected drift, and collusion scenarios.

6.1 A V&V Reference Architecture

The three directions compose into a minimal reference architecture (Figure 3), mirroring the real-estate analogy from §1:

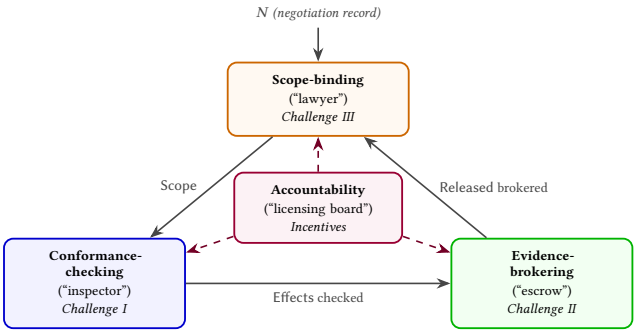

Figure 3: A minimal V&V reference architecture for agent workflows, mirroring the problem structure from Figure 2. Each layer addresses one facet of the trilemma; accountability (center) provides cross-cutting deterrence. No single layer dissolves the trilemma.

(1) **Scope-binding layer** (the "lawyer"). At negotiation time, extracts a structured term sheet Scope from N and obtains explicit commitment from all parties. It enforces a canonical schema so downstream checks have a well-defined target. Candidate mechanisms include structured forms with free-text annotations, LLM-assisted extraction with human confirmation, and signed term-sheet protocols.
(2) **Conformance-checking layer** (the "inspector"). At execution time, monitors tool calls and compares realized effects against the committed Scope, flagging multi-dimensional drift before it becomes irreversible. It implements graded conformance (violation vectors) and drift budgets to separate benign churn from adversarial misbinding.
(3) **Evidence-brokering layer** (the "escrow"). Mediates artifact exchange, choosing Profile to balance verification power against disclosure cost, brokering selective proofs (commitments, SNARKs, TEE attestations) so principals can verify without over-revealing, and managing dispute escalation under asymmetric trust.
(4) **Accountability layer** (the "licensing board"). Maintains reputation, enforces consequences for detected violations (revocation of delegated authority, sanctions, compensation), and hosts benchmarks and red-team evaluations. It is the deterrence backstop that makes the other layers credible.

No single layer dissolves the trilemma; each addresses a different facet. Scope-binding constrains the input to extraction (Challenge III), conformance-checking operationalizes Challenge I, evidence-brokering navigates Challenge II, and accountability supplies the incentive alignment that the formal predicates alone cannot. A concrete instantiation, even partial and scoped to a single domain (hotel booking, procurement), would be a significant step toward the V&V infrastructure agent workflows currently lack.

6.2 The Role of Trusted Infrastructure

The analysis in §2–§5 deliberately adopts a trust-minimized baseline to expose structural constraints. Real workflows operate within institutional trust layers (payment processors, platform arbitration, legal systems, regulated intermediaries) that partially mitigate

treachery. Recent work on agent infrastructure [6] identifies attribution, interaction shaping, and harm remediation as key governance functions; TEP's accountability layer instantiates the attribution function while exposing the disclosure cost any attribution mechanism must navigate.

What trusted layers buy. A *platform-level auditor* relaxes T3: it can verify artifacts on behalf of the buyer without exposing raw evidence to all parties, analogous to an escrow agent inspecting documents the buyer never sees. *Payment guarantees and refund policies* (chargebacks, escrow holds) mitigate T4 by reducing the cost of risky progress. *Legal and regulatory frameworks* supply the T5 deterrence the formal predicates alone cannot: the threat of contractual liability or regulatory sanctions constrains strategic behavior by imposing consequences.

What trusted layers do not buy. Important residual risks remain. **A1** persists because disputes are *semantic*, not transactional: a payment reversal does not settle whether "refundable" meant full refund or store credit. **A2** persists when a trusted layer's jurisdiction is partial: if a seller's backend is outside the platform's logging boundary, the auditor still relies on seller-provided evidence. **A3** persists because tool composition crosses service boundaries that no single platform audits. **A4** persists because trusted layers mediate disputes *after* the fact; colluders can still manufacture urgency that forces bypass *before* safeguards engage. **A5** is especially resilient: it violates no rule, so legal recourse and platform arbitration have no trigger. More broadly, the structural coupling of §4.4 is unaffected: no auditor compensates for a scope extraction that missed an enforcement-relevant attribute. Trusted layers shift the frontier; they do not eliminate it. Composing detection and deterrence across trusted and untrusted segments of a workflow is the key engineering challenge this reference architecture must meet.

7 Related Work

TEP sits at the intersection of delegation and access control, accountability, privacy-preserving verification, and agent/tool security; we organize prior work along the V&V trilemma.

Delegation, capabilities, and the confused deputy. Capability-based security makes authority an explicit, transferable object [9], the confused deputy shows how a component can be induced to misuse it [19], and classic protection work emphasizes complete mediation [2, 35], engineered at OS and hardware levels by modern capability systems [38, 40]. TEP adds the semantic gap between negotiated intent and enforceable scope, and the evidentiary tension where proving conformance itself violates disclosure bounds under collusion.

Access control, trust management, and policy languages. Classical access control (RBAC/ABAC [20, 36]), authorization logics and trust management [1, 3], policy languages (XACML [28]), relationship or tuple-based authorization (Zanzibar [29]), attenuated credentials (macaroons [4]), OAuth [18], usage control (UCON [30]), and graph-based frameworks (NGAC [12]) bind identity to operation but not to runtime-negotiated operands; TEP binds such operands and couples conformance with bounded disclosure under collusion. Lampson's confinement [22] is a direct precursor: confinement prevents leaks, while TEP asks whether *verification across principals* is achievable without leaks, under collusion and without a trusted monitor.

Auditing, provenance, and accountability under adversaries. Logging and provenance mechanisms [23, 27, 31] assume observability monotonically improves integrity; T3 stresses the opposite, that evidence can be an attack surface, so TEP requires *selective* accountability that detects misbinding drift while respecting disclosure bounds, a trade-off Proposition 1 quantifies.

Privacy-preserving verification and selective disclosure. Cryptographic techniques (zero-knowledge proofs [14], commitments [32], SNARKs [17], anonymous credentials [5]) provide building blocks, but TEP is not reducible to any single primitive: open questions are what must be proved and by whom, how proofs evolve under amendments and dynamism (T2), and how disputes resolve when policies conflict and no trusted mediator exists.

Agent/tool security, prompt injection, and multi-agent orchestration. Tool-augmented agents [37, 42] and multi-agent orchestration [41] operationalize action-taking LLMs; prompt injection shows untrusted inputs can compromise them [16, 26], and red-teaming exposes the adversarial surface of computer-use agents [25]. Secure workflow and contract-formalization systems [15, 33] assume pre-agreed schemas and trusted orchestrators, sidestepping $N \rightarrow$ Scope extraction and the disclosure tension. Chan et al. [6] propose "agent infrastructure"; TEP complements this by formalizing *why* it is structurally hard, and task-scoped authorization [39] instantiates the scope-binding layer but leaves the conformance and disclosure edges open. Much of this literature targets single-agent compromise; TEP addresses *multi-principal conflict* and *collusion*, where individually authorized tool calls compose into misbinding drift, scope laundering, or value degradation.

8 Conclusion

As LLM agents execute multi-agent workflows for principals with conflicting interests, treachery becomes the central threat: effects that violate intent while released evidence looks locally legal. We formulate detection as the Treacherous Envoy Problem, identify the V&V trilemma (expressive negotiation, verifiable conformance, bounded disclosure) as the structural reason it is hard, give formal evidence for two coupling edges, and catalogue five tensions and five exploitation patterns. Like human institutions, the path forward is not to eliminate the trilemma but to layer verification so it is possible when needed and backed by consequences. The pressing open challenge is compositional: detection and deterrence that remain sound when trusted and untrusted layers interoperate across organizational boundaries. Progress requires co-design across the directions of §6, and, as immediate community action, a shared benchmark with ground-truth Scope annotations, injected drift, and collusion scenarios at staging levels S1–S4.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. This work was supported in part by NSF grant 2112471 and a Schmidt Sciences Safety Science award.

References

[1] Martín Abadi, Michael Burrows, Butler W. Lampson, and Gordon D. Plotkin. 1993. A Calculus for Access Control in Distributed Systems. *ACM Transactions on Programming Languages and Systems* 15, 4 (1993), 706–734.

[2] James P. Anderson. 1972. *Computer Security Technology Planning Study.* Technical Report ESD-TR-73-51. Electronic Systems Division, Air Force Systems Command.

[3] Moritz Y. Becker, Cédric Fournet, and Andrew D. Gordon. 2010. SecPAL: Design and Semantics of a Decentralized Authorization Language. *Journal of Computer Security* 18, 4 (2010), 619–665. doi:10.3233/JCS-2009-0364

[4] Arnar Birgisson, Joe Gibbs Politz, Úlfar Erlingsson, Ankur Taly, Michael Vrable, and Mark Lentczner. 2014. Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud. In *Network and Distributed System Security Symposium (NDSS).* doi:10.14722/ndss.2014.23212

[5] Jan Camenisch and Anna Lysyanskaya. 2004. Signature Schemes and Anonymous Credentials from Bilinear Maps. In *Advances in Cryptology – CRYPTO 2004.* 56–72. doi:10.1007/978-3-540-28628-8_4

[6] Alan Chan, Kevin Wei, Sihao Huang, Nitarshan Rajkumar, Elija Perrier, Seth Lazar, Gillian K. Hadfield, and Markus Anderljung. 2025. Infrastructure for AI Agents. *Transactions on Machine Learning Research* (2025). arXiv:2501.10114.

[7] Victor Costan and Srinivas Devadas. 2016. Intel SGX explained. *Cryptology ePrint Archive* (2016).

[8] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory* (2nd ed.). Wiley-Interscience. doi:10.1002/047174882X

[9] Jack B. Dennis and Earl C. Van Horn. 1966. Programming Semantics for Multiprogrammed Computations. *Commun. ACM* 9, 3 (1966), 143–155. doi:10.1145/365230.365252

[10] Danny Dolev and Andrew Yao. 1983. On the security of public key protocols. *IEEE Transactions on Information Theory* 29, 2 (1983), 198–208.

[11] John R. Douceur. 2002. The Sybil Attack. In *International Workshop on Peer-to-Peer Systems (IPTPS) (Lecture Notes in Computer Science, Vol. 2429).* 251–260. doi:10.1007/3-540-45748-8_24

[12] David Ferraiolo, Ramaswamy Chandramouli, Vincent Hu, and Rick Kuhn. 2016. *A Comparison of Attribute Based Access Control (ABAC) Standards for Data Service Applications: Extensible Access Control Markup Language (XACML) and Next Generation Access Control (NGAC).* NIST Special Publication 800-178. National Institute of Standards and Technology. doi:10.6028/NIST.SP.800-178

[13] Seth Gilbert and Nancy Lynch. 2002. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News* 33, 2 (2002), 51–59.

[14] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. 1985. The Knowledge Complexity of Interactive Proof-Systems. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing (STOC).* 291–304. doi:10.1145/22145.22178

[15] Guido Governatori, Zoran Milosevic, and Shazia Sadiq. 2006. Compliance checking between business processes and business contracts. In *2006 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06).* IEEE, 221–232.

[16] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In *AISec@CCS 2023.* 79–90. doi:10.1145/3605764.3623985

[17] Jens Groth. 2016. On the Size of Pairing-Based Non-interactive Arguments. In *Advances in Cryptology – EUROCRYPT 2016.* 305–326. doi:10.1007/978-3-662-49896-5_11

[18] Dick Hardt. 2012. The OAuth 2.0 Authorization Framework. RFC 6749. doi:10.17487/RFC6749

[19] Norman Hardy. 1988. The Confused Deputy (or why capabilities might have been invented). *ACM SIGOPS Operating Systems Review* 22, 4 (1988), 36–38.

[20] Vincent C. Hu, David F. Ferraiolo, Rick Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller, and Karen Scarfone. 2014. *Guide to Attribute Based Access Control (ABAC) Definition and Considerations.* NIST Special Publication 800-162. National Institute of Standards and Technology. doi:10.6028/NIST.SP.800-162

[21] Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems* 4, 3 (1982), 382–401. doi:10.1145/357172.357176

[22] Butler W. Lampson. 1973. A Note on the Confinement Problem. *Commun. ACM* 16, 10 (1973), 613–615. doi:10.1145/362375.362389

[23] Ben Laurie, Adam Langley, and Emilia Kasper. 2013. Certificate Transparency. RFC 6962. doi:10.17487/RFC6962

[24] Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023. API-Bank: A Comprehensive Benchmark for Tool-Augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP).* arXiv:2304.08244.

[25] Zeyi Liao, Jaylen Jones, Linxi Jiang, Yuting Ning, Eric Fosler-Lussier, Yu Su, Zhiqiang Lin, and Huan Sun. 2026. RedTeamCUA: Realistic Adversarial Testing of Computer-Use Agents in Hybrid Web-OS Environments. In *Proceedings of the International Conference on Learning Representations (ICLR).* arXiv:2505.21936.

[26] Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. 2024. Formalizing and Benchmarking Prompt Injection Attacks and Defenses. In *33rd USENIX Security Symposium (USENIX Security 24).* 1831–1847. arXiv:2310.12815.

[27] Luc Moreau, Paolo Missier, Khalid Belhajjame, and others. 2013. PROV-DM: The PROV Data Model. W3C Recommendation. https://www.w3.org/TR/prov-dm/

[28] OASIS XACML Technical Committee. 2013. eXtensible Access Control Markup Language (XACML) Version 3.0. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html. OASIS Standard, 22 January 2013.

[29] Ruoming Pang, Ramon Caceres, Mike Burrows, Zhifeng Chen, Pratik Dave, Nathan Germer, Alexander Golynski, Kevin Graney, Nina Kang, Lea Kissner, Jeffrey L. Korn, Abhishek Parmar, Christina D. Richards, and Mengzhi Wang. 2019. Zanzibar: Google's Consistent, Global Authorization System. In *2019 USENIX Annual Technical Conference (USENIX ATC 19).* USENIX Association, 33–46. https://www.usenix.org/conference/atc19/presentation/pang

[30] Jaehong Park and Ravi Sandhu. 2004. The UCON$_{ABC}$ Usage Control Model. *ACM Transactions on Information and System Security* 7, 1 (2004), 128–174. doi:10.1145/984334.984339

[31] Thomas Pasquier, Xueyuan Han, Mark Goldstein, Thomas Moyer, David Eyers, Margo Seltzer, and Jean Bacon. 2017. Practical Whole-System Provenance Capture. In *Symposium on Cloud Computing (SoCC '17).* 405–418. doi:10.1145/3127479.3129249

[32] Torben P. Pedersen. 1991. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In *Advances in Cryptology – CRYPTO '91 (Lecture Notes in Computer Science, Vol. 576).* 129–140. doi:10.1007/3-540-46766-1_9

[33] Chris Peltz. 2003. Web Services Orchestration and Choreography. *Computer* 36, 10 (2003), 46–52. doi:10.1109/MC.2003.1236471

[34] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. In *International Conference on Learning Representations (ICLR).* arXiv:2307.16789.

[35] Jerome H. Saltzer and Michael D. Schroeder. 1975. The Protection of Information in Computer Systems. *Proc. IEEE* 63, 9 (1975), 1278–1308. doi:10.1109/PROC.1975.9939

[36] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. 1996. Role-Based Access Control Models. *IEEE Computer* 29, 2 (1996), 38–47. doi:10.1109/2.485845

[37] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Advances in Neural Information Processing Systems (NeurIPS).*

[38] Jonathan S. Shapiro, Jonathan M. Smith, and David J. Farber. 1999. EROS: a fast capability system. In *Proceedings of the 17th ACM Symposium on Operating Systems Principles (SOSP).* 170–185. doi:10.1145/319151.319163

[39] Reshabh K. Sharma, Linxi Jiang, Zhiqiang Lin, and Shuo Chen. 2026. PAuth: Precise Task-Scoped Authorization for Agents. arXiv:2603.17170 [cs.CR] https://arxiv.org/abs/2603.17170

[40] Robert N. M. Watson, Jonathan Woodruff, Peter G. Neumann, Simon W. Moore, Jonathan Anderson, David Chisnall, Nirav Dave, Brooks Davis, Khilan Gudka, Ben Laurie, Steven J. Murdoch, Robert Norton, Michael Roe, Stacey Son, and Munraj Vadera. 2015. CHERI: A Hybrid Capability-System Architecture for Scalable Software Compartmentalization. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy (SP).* 20–37. doi:10.1109/SP.2015.9

[41] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework. arXiv:2308.08155 [cs.AI] doi:10.48550/arXiv.2308.08155

[42] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations (ICLR).* arXiv:2210.03629.