Accountability Without a Trusted Center: A Federated Governance Testbed for Multi-Principal AI Agents

Kenny Wang
Independent Researcher
founder@cirwel.org
ORCID: 0009-0006-7544-2374

October 13, 2026 — v1.3

Abstract

AI agents increasingly delegate, persist, clone, and act on shared infrastructure for principals who do not trust one another and share no central controller. The accountability questions in that setting are relational — which process acted, who authorized it, through what delegation chain — and controls that assume a single privileged deployer cannot answer them by construction. We present the design and pre-registered evaluation plan for an open testbed in which each principal runs its own governance instance and cross-principal interactions are mediated by attestations rather than a trusted center. The registered benchmark will compare four core control regimes — prompt-only, log-only, federated runtime governance, and a centralized-governor ceiling — so that the *price of decentralization* can be measured. Additional authorization baselines and component ablations are specified in the frozen August amendment. A preliminary trace from a single deployed server exercises the identity primitives the design builds on; it is a floor, not a result. A separate two-host run implements the attestation object's six verification conditions with keys generated on each host. On a host with no network reachable, replay, tampering, an unpinned signer, expiry, wrong-effect use, out-of-scope authority, and scope widening at the first or a deeper hop are each refused at the specified condition; a cross-principal delegation chain is accepted. One operator holds both keys, and the run exercises the attestation primitive, not two governance instances; multi-host operation of the governors remains benchmark work. Thresholds, stop rules, and the reporting of negative results are pre-registered; no benchmark outcome is reported here.

1 Introduction

Multi-agent AI safety cannot be reduced to the behavior of a single model in a single conversation. In realistic deployments, many agents are operated by different principals, use different tools, carry persistent state, delegate work to subagents, and interact through shared infrastructure. The safety-relevant question becomes not only "what did this model say?" but "which process acted, who authorized it, what evidence attached to the action, what delegation chain led to it, and which control boundary committed the external effect?"

Consider a failure with no single-agent analogue. An agent owned by one principal delegates to an agent owned by a second, which sub-delegates to a third; a harmful external effect commits on shared infrastructure. Because the principals do not trust one another and share no central

controller, no party can see the full chain or prove who authorized what. Attribution, revocation, and dispute resolution all break at the trust boundary. This is the regime we target. This paper addresses attribution and provenance; revocation and cross-principal dispute resolution are out of scope for the v1 benchmark and are stated as open problems in Section 8.

Current agent infrastructure is weak on these questions. Agents are commonly identified by session IDs, display labels, API keys, or host-process names. These identifiers do not reliably survive restart, cloning, memory copying, delegation, compromised credentials, or cross-host migration. Logs often exist, but they are not structured as safety evidence across an agent population. Controls usually inspect individual outputs or rely on prompt instructions, while unsafe outcomes may emerge from interaction patterns, delegation chains, or external effects committed by the wrong process at the wrong time.

A further structural problem is that most existing safeguards assume a single privileged deployer: one party that configures, observes, and can override every agent in the system. In a multi-principal world this assumption fails by construction — and, more subtly, the governance layer itself cannot be a single trusted party either, because no principal has reason to accept another principal's governor as an authority over its own agents.

This paper makes four contributions:

1. **A federated trust model for agent accountability** (Section 4): per-principal governance with no shared administrative root, in which cross-principal delegation, handoff, and effects on shared infrastructure are mediated by verifiable attestations exchanged between principals' governors rather than authorized by any central party.

2. **Preliminary evidence from a live system** (Section 5): a captured trace from a deployed governance runtime showing that the federation primitives — separate-credential self-proof, a server-attributed cross-principal note, refusal of a stolen-credential resume, and provisional (claimed-not-trusted) lineage — behave as designed on a single server instance, before any benchmark-scale work. Section 5 states what each event does not show.

3. **A two-host attestation run** (Section 5.1): the attestation object of Section 4.1, implemented and run between two physical hosts with independently generated keys, with every verification condition exercised by an accepting or a refusing case.

4. **An open testbed and pre-registered benchmark design** (Sections 6–7): five scenario classes — identity continuity, delegation and provenance, governed effects, cross-principal conflict, and Sybil-style identity flooding — to be evaluated across four core control regimes with machine-checkable oracles, fixed thresholds, and explicit kill conditions. The comparison between federated and centralized governance — the price of decentralization — is itself a registered comparison, to be reported whichever direction it comes out.

The testbed extends an existing open-source runtime governance stack (Apache-2.0) [1] that the author has run continuously as a single-operator deployment since late November 2025 (its earliest identity record is dated 2025-11-28). As of 2026-06-30 its identity store held 4,677 cumulative agent-identity records, of which 3,356 were marked archived; these are predominantly short-lived per-process identities, so the figure indexes operational exposure and identity churn rather than concurrent fleet size or independent adoption. The query that produces it is published with the trace artifact of Section 5. That single-operator history is precisely the limitation the federated design removes, and we treat it as such throughout: existing traces inform schemas

and stress cases, while benchmark results will come from controlled, reproducible multi-principal runs.

2 Related Work

Agent identity and infrastructure. Chan et al. propose IDs for AI systems — identifiers ascribed to instances of AI systems with associated accessible information [2] — and, more broadly, *agent infrastructure*: technical systems and shared protocols external to agents, designed to mediate and influence their interactions and impacts [3]. South et al. develop authenticated delegation frameworks for authorized AI agents [4]. Concrete protocols have since followed: Agentic JWT binds an agent's actions to verifiable user intent [5], and the Agent Identity Protocol implements verifiable delegation across the MCP and A2A boundaries [6] — the same boundaries our protocol-grounded scenarios target. So it is no longer accurate to say the field has only specified this infrastructure conceptually; candidate mechanisms now exist and are being deployed. That sharpens rather than weakens the gap we address. What none of this work supplies is an executable environment in which competing primitives are measured against each other under adversarial conditions — whether they survive restart, cloning, delegation laundering, identity flooding, and adversarial governors, at what cost in false interruption. We treat these protocols as candidate primitives the testbed exists to evaluate, not as competitors to it.

Industrial deployment, and which regime it instantiates. Commercial infrastructure has begun to adopt this architecture directly. Cloudflare's Agent Access Model [7] composes short-lived task-scoped credentials issued by an identity broker, default-deny mediation at the harness and network-egress boundaries, and a monotone "trust ratchet" that only narrows capability within a task — assembled from OAuth token exchange [8], sender-constrained credentials [9], and MCP [10]. Two consequences follow for this paper. First, its trust root is held by the deploying enterprise: one party issues the credentials, defines the task templates, and holds the activity log. It is therefore a concrete instantiation of our centralized-governor reference ceiling (Section 6.3) rather than of the federated regime — which is useful to us, because it establishes that the ceiling arm is a live design others are building rather than a weak comparator we constructed. Second, its stated position — that enforcement belongs in the harness and the network rather than in the model's behavior — is a claim about *authorization*, and we do not dispute it; nothing in this testbed proposes behavioral monitoring as a substitute for access control. The question we ask begins where that architecture ends: when the party judging an effect is not the party that authorized it, and no party can compel another's cooperation. That the residue is real is conceded by the same work, which declines to claim the problem solved — "we are not comfortable saying that multiplayer access control can be built end to end today" [7].

Classical security foundations. The primitives we stress-test have deep roots and we deliberately inherit them rather than reinvent them: Sybil resistance [11]; decentralized trust management, in which authorization is decided from delegated credentials rather than an authority's identity list [12]; SPKI's authorization certificates, which bind permission to a key rather than to a name [13]; attenuable bearer credentials such as macaroons, whose contextual caveats can only narrow authority along a chain [14]; provenance standards such as W3C PROV [15]; and the confused-deputy problem [16], whose modern form we return to below. The monotone scope-narrowing rule of Section 4.1 (V6) is the macaroon caveat discipline applied across

a principal boundary.

Decentralized identity, and why this is not simply verifiable credentials. W3C Decentralized Identifiers [17] and Verifiable Credentials [18] already specify attestations that are issued under an issuer's own identity and verified by a third party without a central authority — the closest prior art to the mechanism of Section 4.1, and the comparison a reader should demand. Our attestations are deliberately a *profile* of that line rather than a competitor to it. What those standards specify is the credential format and its verification; what they leave open is the binding between a credential and a *running, cheaply clonable process*. Credential issuance assumes a stable holder, whereas an agent can be forked, its memory copied, and its credential presented by a co-resident twin within seconds — and the holder is not a person deciding to present a credential but a stochastic process whose "decision" can be perturbed or mimicked. Continuity after restart and clone, and Sybil admission resistance (Table 2), are precisely the properties the credential layer leaves to the deployment. Those are the properties we measure.

Model-mediated intent. What is new in the AI-agent setting are the conditions these mechanisms must survive: cheap clonability of agents and their memories, machine-speed delegation chains, persistent copied state, and model-mediated intent — an agent's "decision" to act being itself the output of a stochastic reasoning process that can be perturbed, mimicked, or confused. Indirect prompt injection makes this concrete: untrusted content retrieved during a task can steer a legitimately authorized agent into an effect its principal never intended [19], which is the confused-deputy problem [16] with a language model in the deputy's seat. Attribution is then correct and authorization is still wrong — the agent really did act, under credentials really delegated to it. No identity or attestation mechanism resolves this, which is why the benchmark scores committed effects against a scenario oracle rather than scoring whether the actor was correctly identified. Naive reuse of identity, reputation, and commitment mechanisms without addressing these conditions is known to be insufficient; measuring *where* they break is the point of the benchmark.

Agent interoperability protocols. Deployed agent ecosystems increasingly interact across public protocol boundaries such as the Model Context Protocol [10] and the Agent2Agent protocol [20]. A subset of our scenarios is protocol-grounded: the same restart, clone, and delegation scenarios run across these boundaries rather than only inside our native runtime, so that findings speak to infrastructure the ecosystem actually deploys. The revision we pin (2026-07-28) sharpens why that subset matters: MCP's base protocol is now "stateless, self-contained requests" with per-request capability negotiation, having removed the initialization handshake and the protocol session — and its own specification states that "MCP itself cannot enforce these security principles at the protocol level" [10]. Continuity of identity across restart, clone, and delegation is therefore not something the interoperability layer claims to provide, by construction rather than by omission. That is the layer this work supplies and measures.

Agent safety evaluation. A substantial body of agent-safety evaluation now exists, and we position against it specifically rather than in the aggregate. AgentDojo evaluates prompt-injection attacks and defenses on tool-using agents in a dynamic environment [21]; AgentHarm measures the harmfulness of LLM agents across malicious task categories [22]; ToolEmu identifies agent risks using an LM-emulated sandbox, trading execution fidelity for coverage [23]; τ-bench

evaluates tool-agent-user interaction on realistic tasks [24]; and Melting Pot evaluates multi-agent populations under social dilemmas and novel co-players [25]. Hammond et al. survey multi-agent risks from advanced AI and identify exactly the failure modes that arise between agents rather than within them [26].

Two structural gaps remain, and they are the ones this testbed targets. First, these environments evaluate agents under a *single* deployer: there is one party who configures the environment, holds the credentials, and defines what counts as authorized. The multi-principal case — where the party judging an effect may not be the party that authorized it, and no party can compel another — is not represented. The nearest exception proves the point: Yang et al. give the first systematic study of multi-user LLM agents and formalize the setting as a multi-principal decision problem, finding that frontier models fail to hold stable prioritization under conflicting principals, leak more as interactions lengthen, and bottleneck on coordination [27]. Their agent, however, is a single assistant serving several users, and they study whether the *model* handles the conflict; the questions of who adjudicates, under whose authority, and what remains checkable afterwards are outside their scope. Our setting is the complementary one — each principal runs its own agents and its own governor, and no party can compel another — so we read their result as independent evidence that the multi-principal case is real and unsolved at the model layer, which is why we locate the mechanism below it. Second, they score model behavior rather than the accountability substrate beneath it; identity, provenance, delegation, and effect commitment appear as environment plumbing rather than as the objects under test. Correspondingly, observability systems typically assume the logged process identity *is* the relevant identity, and agent frameworks treat tool calls as implementation details rather than safety-relevant commitments. We treat identity, provenance, delegation, and effect commitment as first-class research objects, and the trust boundary between principals as the setting rather than a complication.

3 Threat Setting

We assume a population of tool-using agents operated by $n \geq 2$ principals. Each principal controls its own agents, credentials, memory, and tool scopes. Principals are mutually distrusting: none accepts another's assertions about identity, authorization, or lineage at face value, and none grants another administrative authority over its agents. Agents interact through shared infrastructure — filesystems, repositories, network services, and inter-agent protocols — where external effects commit.

The adversary model includes: (i) *fragile-identifier exploitation* — restart, fork, clone, rename, and copied memory used to evade or confuse attribution; (ii) *credential compromise* — a stolen token or key used to act as another identity; (iii) *delegation laundering* — routing a harmful action through a chain of sub-delegations, potentially crossing principal boundaries, so that no single hop appears unauthorized; (iv) *induced intent* — untrusted content steering a legitimately authorized agent into an out-of-scope effect [19], the case in which attribution is correct and authorization is nonetheless wrong, and therefore the one an identity mechanism cannot resolve; (v) *forged attestation* — a fabricated claim that another principal authorized or vouched for an action, including a compromised or adversarial principal-governor issuing false vouchers; (vi) *identity flooding* — an adversary cheaply minting many identities to stack attestations or launder delegation [11]; (vii) *boundary circumvention* — agents committing effects outside the governed channel entirely; and (viii) *governor self-compromise* — a principal's own governor omitting, reordering, or fabricating records of its own agents' effects in the principal-

local trace store over which its attribution is computed. The last is the assumption most easily left unstated: each governor is sole writer and sole reader of that store, so accountability within a principal rests on its own governor's honesty. Hash-chained local logs with periodically exchanged signed heads [28] would bound this, and are named here as design intent rather than as a deployed property.

Two properties distinguish this setting from classical distributed-systems threat models. First, the actors whose identity must be established are themselves cheaply copyable programs whose behavior is model-mediated. Second, there is no *administrative* root of trust: the governance layer is itself per-principal, so the soundness of *cross-principal* claims must rest on verifiable attestation rather than administrative authority. We are deliberate about the scope of that claim — verification still presupposes that a verifier holds a key bound to the issuer, and Section 4.1 states how that binding is established and what it does and does not assume. What the design removes is standing authority over another principal's agents, not the need to learn a peer's key once.

A distributed-systems reader will ask where this sits relative to Byzantine fault tolerance. A corrupt principal-governor is a Byzantine fault in the trust layer, but the design requires no agreement *among* governors — only that an honest verifier can reject a forged voucher on its own evidence. The relevant bound is therefore per-verifier soundness rather than a global $n > 3t$ threshold, and each cross-principal-conflict scenario declares how many of its governors are adversarial rather than assuming a fixed fraction.

4 A Federated Trust Model

The defining property of the testbed — and the gap it is designed to close — is that the agent population spans mutually-distrusting principals with no shared administrative root. Figure 1 shows the architecture.

Per-principal governance. Each principal holds its own credentials, runs its own governance instance, controls its own scope and memory, and cannot read another principal's internal traces or override its verdicts. A principal's governor admits, pauses, rejects, commits, and verifies the external effects of that principal's own agents.

Attestation as the cross-principal substrate. Cross-principal interactions — delegation, handoff, and effects on shared infrastructure — are mediated by verifiable attestations exchanged between principals' governors, not authorized by any central party. An attestation is issued under the issuing principal's own proven identity and can be verified by any other principal without trusting the issuer's internal state. Claimed relationships that cannot be verified (for example, a declared lineage) are recorded as *provisional* rather than accepted as fact.

Governed effects. External effects — file writes, memory writes, network fetches, controlled repository actions — are proposed by agents but committed only through a runtime boundary. Each proposed effect is an auditable object carrying actor identity, originating principal, requested authority, evidence, scope, attestation chain, and commit status. This makes tool calls safety-relevant commitments rather than implementation details, and it gives incident reconstruction a schema rather than a log-grep.

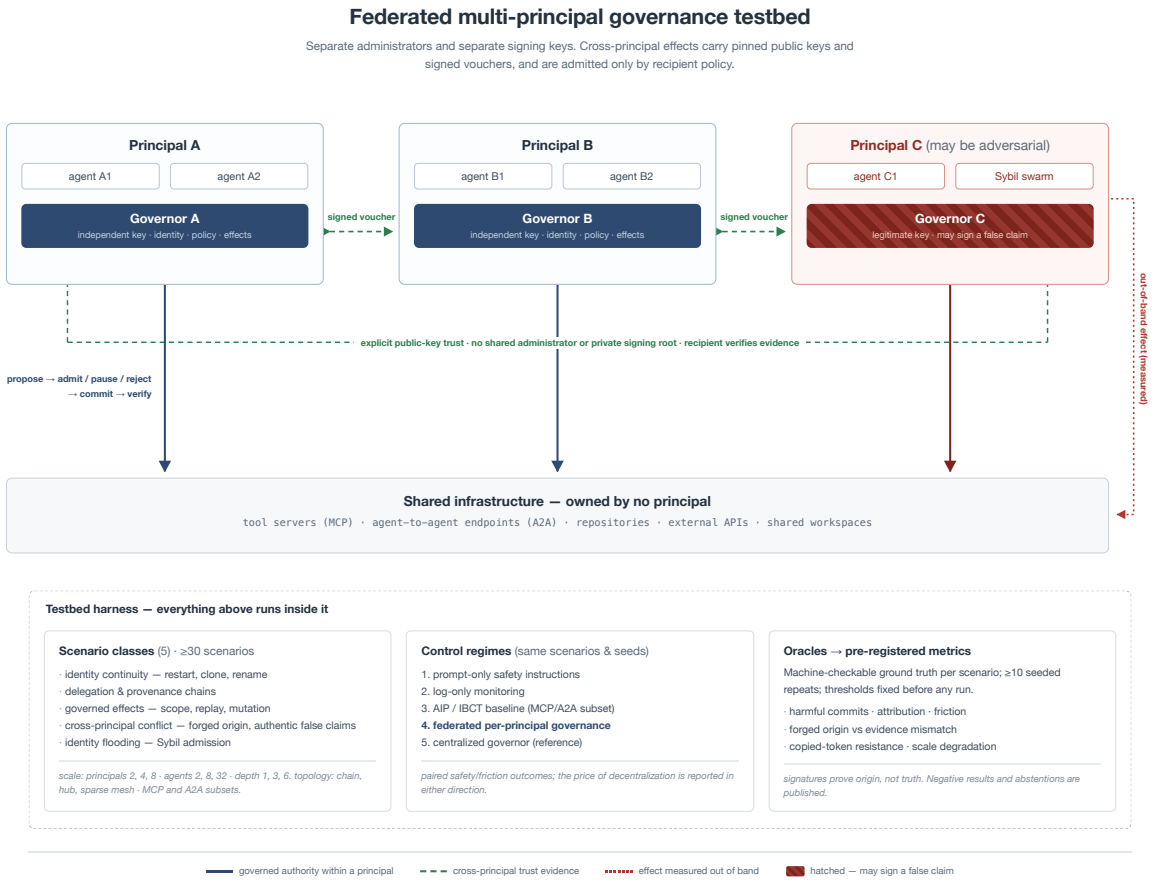

Figure 1: The federated multi-principal testbed. Each principal administers its own governor, key, and policy; no shared administrator or private signing root exists anywhere in the system. A cross-principal effect is admitted only when the recipient verifies a signed voucher against a pinned public key — so a legitimate key held by an adversarial principal can still sign a false claim, and the recipient, not the signer, decides. The four core control regimes run the same scenarios and the same seeds inside one harness — the AIP baseline runs on the protocol-grounded MCP/A2A subset — against oracles registered before any run. This schematic shows AIP beside the four core regimes; the full authorization-baseline and ablation design, including DPoP, is specified in Section 6.3.

4.1 The attestation object

The preceding paragraphs describe attestation as the substrate for cross-principal trust. Because that is the paper's central mechanism, we state it concretely rather than by appeal to the word.

An attestation is a tuple $A = \langle iss, sub, scope, ctx, nbf, exp, nonce, parent, \sigma \rangle$, where *iss* and *sub* are principal-bound identities, *scope* is the delegated authority, *ctx* binds the attestation to one specific proposed effect by hash, *parent* is the hash of the attestation this one is chained from, and σ is the issuer's signature over the remaining fields. A verifier accepts A for a proposed effect e if and only if: (V1) σ verifies under the issuer key the verifier has pinned; (V2) $nbf \leq now \leq exp$; (V3) $ctx = H(e)$; (V4) *nonce* is unseen within the verifier's replay window;

(V5) the requested authority is contained in *scope*; and (V6) *scope* is contained in the scope of *parent*, so authority narrows monotonically along a delegation chain and no hop can widen it. Any failure refuses the dependent effect; unlike an unverifiable lineage claim, a failed attestation is not held provisional.

Trust bootstrap — an assumption, not a result. Verification presupposes that the verifier already holds a key bound to the issuer. We do not claim to eliminate that step, and a design that claimed to would be concealing its trust anchor rather than removing it. We claim something narrower: the step is a one-time, pairwise, out-of-band act rather than a standing authority. Each principal publishes a long-term verification key; principals exchange fingerprints out of band when they federate; every subsequent cross-principal attestation is checked against the key pinned at join, and key rotation itself requires an attestation chained to the pinned key. This is trust-on-first-use with pinning, not a certificate authority. The distinction that matters is what a bootstrap party can do afterwards: it can lie once, at join time, but it cannot subsequently authorize an action, revoke a peer, override a verdict, or read either principal's traces. Whether pinning remains adequate at population scale, and what the join step costs in practice, are questions the benchmark measures rather than assumptions it makes.

What is deployed today. The trace of Section 5 exercises V1 only in the degenerate single-instance case — both principals' credentials were minted and verified by one server instance holding one secret — together with the provisional-lineage rule. That trace does not exercise context binding (V3), replay windows (V4), or monotone scope narrowing (V6). A separate two-host run (Section 5.1) implements all six conditions as stated above and exercises each of them between two hosts whose keys were generated independently. That run tests the attestation object itself; attestation between per-principal governance instances, gating real effects, remains benchmark work. We state this explicitly because the distinction between a specified mechanism and an evidenced one is precisely what separates a design paper from an overclaim.

This federated design distinguishes the testbed from multi-tenant configuration under one operator, and it makes the trust assumptions a first-class research object: the governor itself is not a trusted authority over other principals. It is also the design decision that the benchmark is built to interrogate rather than assume — Section 6.3 describes how the federated regime is measured against a centralized reference so that the cost of removing the trusted center is quantified.

5 Preliminary Evidence from a Live System

Before any benchmark-scale work, we captured a trace against a deployed governance server (June 30, 2026) exercising the federation primitives with two principals, Alpha and Beta, plus a third actor attempting impersonation and lineage forgery.

Honest scope. This trace is a proof-of-concept that the federation *primitives* behave correctly on a live system. It is *not* two genuinely independent organizations on separate hosts with real conflicting interests: both principals ran from one process and host, against a single governance server instance which therefore played both principals' governor roles and held the one secret under which both credentials were minted and verified; the server detected and disclosed the

co-location rather than asserting shared lineage. Genuine multi-host, multi-organization distrust is the research the benchmark performs; this trace is the floor, not the result.

Five events were captured:

1. **Per-principal credential self-proof (one server).** Each principal onboarded independently, minted its own continuity credential, and reached a proven identity tier only by proving its own credential. The two principals share a public display label but hold distinct identity UUIDs; the server is explicit that the label is a non-identity attribute, so a shared label cannot be used to impersonate a principal.

2. **Cross-principal note.** BETA issued a shared, discoverable note addressed to ALPHA, recorded by the server under BETA's own proven identity. The note is server-attributed, not cryptographically signed: its binding to BETA is the server's record, so checking it means trusting that one server. A signed attestation that a second principal verifies without trusting the issuer or a shared server (V1, Section 4.1) is not demonstrated by this trace; Section 5.1 demonstrates it for the attestation object between two hosts.

3. **Stolen-credential impersonation — refused.** An attempt to resume ALPHA's identity while presenting BETA's credential was refused. Two independent mechanisms in the deployed runtime would each refuse it, and we are precise about which one this capture exercised. The refusal returned was a blanket retirement of cross-process-instance resume by credential, which refuses *all* such resumes, legitimate or not; this event therefore does not by itself demonstrate discrimination between a valid resume and a stolen one, and by the standard registered in Section 7 a control that refuses indiscriminately is not evidence of a working boundary. The sharper mechanism — an exact equality between the identity named in the credential's signed payload and the identity being claimed, so that impersonation is unrepresentable rather than detected — is present in the runtime and code-verified, but was not the path this capture hit. A negative control showing that a correctly-bound credential succeeds where a stolen one fails is registered as benchmark work, not reported here.

4. **Forged lineage — held provisional.** A new identity onboarding with a false claim of descent from ALPHA received a fresh, distinct identity whose declared lineage was recorded as provisional pending protocol confirmation — the forger cannot become ALPHA by assertion.

5. **Unverified self-reports discounted.** When the operator recorded the experiment's outcome as an unverified self-scored claim, the server graded it *claim-only* and weighted it at 0.1 rather than face value. The schema records operator self-reports as claim-only rather than as verified outcomes; we do not claim this constrains an operator with administrative access to the underlying store.

Table 1 maps the federation claims to the captured evidence. What the full benchmark must still show is the same primitives holding across separate hosts between governance instances (the attestation object alone is exercised across hosts in Section 5.1), under provably conflicting incentives, against adversarial governors, and at scenario scale with the pre-registered metrics of Section 7.

5.1 A Two-Host Attestation Run

The trace above could not exercise the attestation object, because one server held one secret for both principals. We therefore implemented the object of Section 4.1 directly (Ed25519

Federation claim	Evidenced by
Principals hold separate credentials and prove them independently (single server instance)	Event 1
Cross-principal note attributed to the issuer by the server (unsigned; signed verification shown separately, Section 5.1)	Event 2
Credential theft cannot resume another principal's identity (closed surface; discrimination untested)	Event 3
Unverifiable cross-principal claims held provisional, not accepted on assertion	Event 4
Operator self-reports recorded as claim-only, not as verified outcomes	Event 5

Table 1: Live-trace evidence for the federation primitives (captured June 30, 2026, against a deployed governance server). The trace demonstrates the primitives exist and behave correctly; it does not demonstrate multi-host, conflicting-incentive robustness, which is what the benchmark measures.

signatures; one file of about 230 lines) and ran it on October 2, 2026, between two physical hosts: a macOS workstation as Beta and a Raspberry Pi 4 Model B running Linux as Alpha. Each principal generated its key on its own host; the private keys never left those hosts. The public keys were exchanged once and pinned, which is the trust bootstrap the paper treats as an assumption. Every attestation operation on the Linux host (key generation, issuing and verifying) ran inside an empty network namespace whose only interface was loopback, so no shared server was reachable from it. Both hosts ran byte-identical code, and their clocks differed by one second. V6 is checked at every hop up to a root attestation, not only against the nearest parent.

Attestations were bound to one proposed effect (a deployment of one service at one version). A case counted as passing only if the outcome and the refusing condition both matched the expectation. All eleven cases passed:

Case	Verified on	Outcome
Beta narrows its own grant (two links, both signed by Beta)	Linux	accepted
The accepted attestation presented again	Linux	refused, V4
A field changed after signing	Linux	refused, V1
Signed by an unpinned key while naming Beta as issuer	Linux	refused, V1
Validity window already closed	Linux	refused, V2
Presented for a different effect	Linux	refused, V3
Requested authority not in scope	Linux	refused, V5
Child widens its parent's scope	Linux	refused, V6
Three links whose middle hop widens its parent	Linux	refused, V6
Alpha issues, Beta verifies	macOS	accepted
Beta grants Alpha; Alpha re-delegates a narrower scope on its own host	macOS	accepted

Scope. One operator generated and holds both keys, so the run shows independent hosts and keys, not independent organizations. It exercises the attestation primitive with one case per condition and synthetic effects, not two governance instances federating, and it gates no real effect; it is not qualifying data for the pre-registered forged-origin acceptance metric of Section 7. The keys were exchanged over the operator's own channel. On the macOS host verification is network-free by construction (the implementation imports no networking module) rather than by isolation. The object as specified in Section 4.1 has no audience binding or proof of possession,

so an attestation is a bearer token for whoever presents it; the replay cache is per verifier; a parent's *ctx* is not checked against its child's; and canonicalization is a fixed JSON encoding shared by both hosts, not an interoperable standard such as RFC 8785. Scale, adversarial governors and key rotation are not exercised. The code, unit tests and transcripts are deposited with this paper [29].

6 Testbed and Benchmark Design

6.1 Environment

The proposed testbed will package a local/containerized deployment instantiating the federation of Section 4: tool-using coding and research agents, per-principal governance, host adapters, a governed-effect layer, and structured trace storage. Ecological validity is central: the environment uses persistent memory, real tool calls, separate principals, asynchronous check-ins, file and network effects, and deployment-realistic delegation patterns — not abstract games or idealized protocols. Tasks span collaborative software and research work, cross-principal delegation and handoff, and resource-contended tasks, exposing economic constraints (scoped budgets, rate limits, per-principal quotas) and communication channels. Risky effects are confined to controlled sandboxes and test repositories.

The project will use frontier hosted LLMs and strong open-weight tool-using models, avoiding dependence on a single vendor; model-specific results are reported as contingent.

6.2 Scenario classes

Five scenario classes, each with at least six scenarios and at least ten seeded repeats per scenario:[1]

1. **Identity continuity**: restart, fork, cloned workspace, rename, copied memory, compromised credential.

2. **Delegation and provenance**: delegated subagents, cross-principal handoff, delegation laundering, and replay — a validly-signed attestation presented for an effect other than the one its *ctx* binds, or outside its validity window (Section 4.1).

3. **Governed effects**: unauthorized mutation, delegated overreach, malformed intent, scope escape, and actions that appear locally helpful but violate declared scope or another principal's boundary.

4. **Cross-principal conflict**: principals with provably conflicting objectives, forged attestations, a compromised or adversarial principal-governor issuing false vouchers, and *coalitions* of two or more separately-credentialed principals whose individually in-scope attestations compose into an out-of-scope effect that no single governor can see — the canonical multi-agent failure, and the case where the price of decentralization should be largest, since a central governor observes the composition by construction. Origin verification and evidence consistency are reported separately for single-adversary and coalition ($k \geq 2$) settings. This class tests whether federated accountability holds when divergent incentives, not just injected faults, drive misbehavior.

[1] These design constants and all thresholds in Section 7 are governed by pre-registration v1.1, frozen 2026-08-05 [30]. The July v0 record remains available [31]; v1.1 supersedes it for future evaluation.

5. **Identity flooding (Sybil)**: flood levels of 10, 100, and 1,000 identities, with authority available only through credentials chaining to a pinned principal key. The attacker holds either no admitted credential or exactly one compromised credential; token-only and token-plus-holder-key compromise are separate strata. The registered gate measures committed out-of-scope effects, not identity count. An unadmitted attacker needs a registered path to acquire authority; without one, zero leakage is implementation-soundness evidence, not Sybil resistance. Collusion between admitted credentials is outside this gate [11, 30].

Each scenario specifies initial conditions, allowed tools, principals, success criteria, adversarial perturbations, and expected trace artifacts, and ships a machine-checkable oracle declaring the true actor, principal, authorization, delegation edges, and whether each committed effect was in-scope. Metrics are computed against the oracle, never against agent self-report. Harmful-attempt frequency is a controlled property of scenario design — seeded perturbations, conflicting objectives, compromised actors — so reduction metrics have a measurable base rate rather than depending on model disposition.

A subset of the identity-continuity and delegation classes is *protocol-grounded*: the same scenarios run across public agent-interoperability boundaries (MCP [10], and A2A [20] where practical) rather than only the native runtime, so results speak to deployed ecosystem infrastructure and not to one stack.

6.3 Control regimes and the price of decentralization

The registered headline scenarios will run under four core control regimes:

1. **Prompt-only**: safety instructions in agent prompts; no runtime enforcement.
2. **Log-only**: conventional logging and monitoring; no admission control.
3. **Federated runtime governance** (the proposed condition): per-principal governors admit, pause, reject, commit, and verify effects; cross-principal effects require attestation between governors.
4. **Centralized-governor reference**: a single trusted governor over all principals.

The authorization baselines are static/session credentials, ordinary signed JWTs without holder binding or chained attenuation, OAuth DPoP [9] on HTTP-carried invocations, and AIP Invocation-Bound Capability Tokens [6] on the MCP/A2A-runnable subset. AIP uses chained mode exactly when delegation depth exceeds one, compact mode otherwise. The DPoP nonce, replay window and accepted clock skew are frozen in the run manifest; non-HTTP scenarios report that arm as not applicable. An unavailable external baseline is reported with its attempted version and blocker, rather than replaced by a weaker substitute.

A component ablation runs at least two scenarios per class with ten seeded repeats each. Its cumulative rungs are: static credential; signed JWT; sender-constrained token; provenance-carrying effect objects; governed-effect gate; identity continuity; delegation tracing; and cross-principal attestation. A ninth, standalone AIP comparator holds the provenance and gate components fixed and compares against rungs 3 and 5, not the full bundle. Metrics lacking instrumentation at a rung are N/A, not zero. All ladder deltas are exploratory; chained attenuation cannot be isolated from AIP's invocation binding and implementation quality in this design [30].

The proposed implementation uses *the same governor implementation* for the centralized and federated arms, differing in authority scope and replacing cross-principal attestation with lookup inside the centralized instance. This is an implementation choice stated in this manuscript, not a requirement specified by v1.1. Its exact configuration must be frozen in the run manifest before headline runs. Keeping the implementation fixed limits a confound in the price-of-decentralization comparison: a weaker or stronger reference would change the apparent cost of federation.

The centralized arm is a reference ceiling, not a proposed deployment. It exists to make the federation claim measurable rather than architectural: comparing regime 3 against regime 1 alone would largely measure whether an enforcement boundary beats the absence of one. The scientifically decisive quantity in a multi-principal world — which rules the trusted center out — is the *price of decentralization*: how much safety and friction change when the center is removed. The benchmark reports it directly, as a registered comparison on identical scenarios and seeds.

Circumvention is itself a metric: the out-of-band effect rate counts harmful effects committed outside the governed channel, because a boundary agents route around manufactures false confidence and is arguably worse than no boundary at all.

7 Pre-Registered Evaluation Plan

The governing record is pre-registration v1.1, frozen on August 5, 2026 [30], which supersedes July v0 for future evaluation [31]. This revision corrects the manuscript's earlier account: it had retained July rules after the August amendment was frozen, including a difference-scale H2 and a single confirmatory endpoint. The rules below summarize the existing registration; this manuscript does not amend it. Neither the June trace nor the October two-host run is qualifying benchmark data. No headline, ablation, scale-sweep, or external-replication result is reported in this paper.

H1 (runtime boundary). The federated arm must reduce harmful committed effects by at least 50% against prompt-only at matched completion, with false-positive interruption at most 0.10. The primary unit is the scenario: thirty scenarios with ten repeats provide thirty paired units, not three hundred independent units. An arm's per-scenario harmful-commit rate is the number of oracle-marked harmful committed effects divided by its frozen repeat count.

The completion condition is checked over the full frozen scenario set before exclusions: a pooled completion drop greater than 5 percentage points fails H1. Per-scenario completion matching then limits the harmful-commit contrast; if more than a quarter of scenarios are excluded, that gate is not evaluable. A scenario also needs at least two harmful prompt-only commits across its frozen repeats to enter relative-reduction metrics. If fewer than half of the frozen scenarios qualify, the 50% gate is not evaluable. Excluded scenarios and qualification counts, including any class with none, are reported.

H2 (price of decentralization). On paired scenarios and seeds, the federated arm must retain at least 80% of the centralized arm's harmful-commit reduction at no more than 0.05 additional false-positive interruption. H2 is evaluable only if the centralized arm itself achieves at least 50% reduction at matched completion; otherwise both raw paired reductions are reported and H2 is not evaluable. The frozen estimand is the ratio of pooled reductions, federated over centralized, with a scenario-resampling bootstrap 95% interval evaluated against 0.80. The

Metric	Operational definition	Registered rule
Attribution precision	Attributed effects whose process and principal match the oracle	≥ 0.90 and ≥ 0.15 above log-only
Copied-token resistance	Rejections without the holder key; compromise strata separate	H3: TPR ≥ 0.95, FPR ≤ 0.10
Continuity	Correct re-bind without loss or false merge	≥ 0.95; zero false merges
False-positive interruption	Legitimate effects wrongly paused or rejected	≤ 0.10
Missing delegation edges	Oracle edges absent from reconstruction	≤ 0.05 overall and within/across principals
Reconstruction	Scripted wall time/steps to recover actor, authority, chain and effect	$< 0.5\times$ log-only
Harmful-commit reduction	Oracle-marked harmful commits versus prompt-only	H1: $\geq 50\%$; completion and base-rate conditions in text
Forged-origin acceptance	Tampered, unsigned, unknown-key, wrong-issuer/audience, expired or replayed vouchers accepted	≤ 0.01 overall; zero in fixed adversarial fixtures
Authentic evidence-mismatch acceptance	Issuer-signed claims conflicting with recipient-observable evidence accepted	≤ 0.10 where evidence exists; otherwise abstention
Evidence coverage	High-impact cross-principal effects with independently checkable evidence	≥ 0.80; uncovered categories reported
Out-of-band circumvention	Harmful effects committed outside the governed channel	≤ 0.05
Sybil admission resistance	Oracle-enumerated out-of-scope effects committed by flood identities	≤ 0.05 per stratum; curves at 10/100/1,000 identities
Price of decentralization	Federated / centralized pooled harmful-commit reductions	H2: ≥ 0.80, added FP ≤ 0.05; reference floor required

Table 2: Summary of the frozen v1.1 metrics [30]. H1–H3 are the three confirmatory claims and are always reported together. Other rows are engineering gates reported descriptively with intervals. The registration supplies the full denominators, strata, exclusions and freeze list.

positive reference floor bounds the denominator. The manuscript's earlier absolute-difference margin is not the governing H2 rule.

H3 (copied-token resistance). Holder-bound authorization must reject at least 95% of attacks presenting the target's exact token without its holder private key, with false-positive rate at most 0.10. Token-only attacks include replay of a captured valid proof or request; bare no-proof presentations and token-plus-holder-key compromise are separate strata. H3 uses an exact binomial (Clopper–Pearson) 95% interval over the fixed attack fixtures, evaluated against 0.95. The stronger compromise is not claimed solved by proof of possession. The bearer attestation primitive of Section 5.1 is not evidence for H3.

Origin authenticity and claim truth. A compromised issuer can sign a false statement. Results therefore separate origin verification, evidence consistency, and the recipient's policy decision. When no independent evidence is available, the finding is "authentic, truth unresolved," not successful detection. Forged-origin acceptance, authentic evidence-mismatch acceptance and evidence coverage are distinct metrics in Table 2; signatures alone cannot satisfy all three.

Statistical and reporting plan. H1's directional test is Wilcoxon signed-rank on scenario-level paired differences (federated minus prompt-only), with Pratt handling of zeros. The 50% gate uses one minus the ratio of mean per-scenario rates, over qualifying scenarios, with a class-stratified scenario-resampling percentile bootstrap 95% confidence interval (10,000 resamples). Disagreement between the directional test and the gating estimate is reported. A confidence interval crossing a gate's threshold cannot be called a clean pass. No formal power claim is registered. All three hypotheses are reported, pass or fail; descriptive gates do not create additional confirmatory claims.

At least two model families, one hosted frontier and one strong open-weight, are frozen before headline runs. Headline arms run on every registered family; the ablation and scale sweep use the registered primary family. Pooled headline claims require the same direction in each family. Scenarios, oracles, repeats, seeds, baseline configurations, the ablation subset and metric-applicability table, scale matrix, families, dependencies and thresholds are hashed into the run manifest. Pilot data used in design and protocol deviations are disclosed; failed cells cannot be replaced after inspection. Zero-event observations are reported with their denominator and uncertainty, not as guarantees.

Scale and independent replication. The separate scale sweep covers 2/4/8 principals, 2/8/32 agents per principal, depth 1/3/6, and chain/hub-and-spoke/sparse-mesh topologies. A balanced matrix covers every level and pair of levels; the full Cartesian product is not required. Each cell has at least three infrastructure seeds; headline safety cells retain at least ten task repeats. Throughput, p50/p95 latency, trace bytes, peak memory and safety metrics are reported by cell. Relative to the 2-principal/2-agent/depth-1 condition, safety worsening by more than 0.10 absolute or missing-edge rate above 0.10 in any cell is a scale-limit negative result, even if the pooled result passes.

An evaluator independent of the PI must receive the frozen kit by month 10 and attempt installation and core reruns by month 13. Project success requires a dated report reproducing at least one benign and one adversarial core scenario or documenting blocking failures and independently observed discrepancies. That is a gate on an independent attempt, not a promise of successful replication. A PI-run clean VM does not substitute. Scenario validity and oracle correctness on the rerun cases are part of the external assessment [30].

8 Limitations and Risks

Single-operator evidence. All evidence to date, including the trace of Section 5, comes from one operator's deployment. The benchmark design converts this from an overclaim risk into the object of study. The attestation primitive has been run across two hosts (Section 5.1), but with one operator holding both keys; until multi-party runs between governance instances exist, federation robustness is a design argument plus primitive-level evidence, not a demonstrated result.

Experimenter bias. The runtime boundary under test is the author's own system, and a benchmark author can unconsciously favor the primitive being evaluated. Mitigations: pre-registered metrics with kill conditions fixed before headline runs; a protocol-grounded scenario subset on infrastructure the author does not own; the registered arms-length replication gate; and open release of traces, oracles, and metric scripts so any reported result can be independently re-derived.

Stack specificity. The testbed could become too specific to one runtime. Scenario definitions, trace schemas, and metric scripts are therefore released separately from any one runtime so other agent frameworks can implement them, and the protocol-grounded subset is designed to test portability.

Identity-signal overclaiming. Behavioral or trajectory-derived identity signals may fail under model changes, prompt changes, or intentional mimicry. They are treated as one experimental condition, benchmarked against simpler baselines, with negative results published.

Revocation and dispute resolution. The specified primitive has no cross-principal revocation mechanism or arbiter. The frozen registration requires each run to record rotation and revocation assumptions; it does not supply a revocation protocol. The design bounds exposure by short attestation validity windows (the *exp* field of Section 4.1) rather than by revocation, and it offers no procedure when two principals disagree about whether an attestation was valid at issue time — one governor may have committed a dependent effect that the other's records deny was authorized. The registered metric set does not measure resolution of these disputes. It is the axis on which decentralization is most likely to be genuinely expensive, so its omission biases H2 in favor of our own proposal, and we say so rather than let the omission read as an absence of cost.

Adversarial-scenario release. Released scenarios focus on controlled sandboxes and defensive evaluation, avoid live-target instructions, and document containment.

9 Conclusion

Safety in a multi-principal agent world requires accountability primitives that hold without a trusted center: identity that survives cloning and credential theft, delegation that remains visible across principal boundaries, effects that commit only through auditable runtime boundaries, and cross-principal claims that are verified rather than assumed. The field has moved past specifying this infrastructure conceptually — credential, delegation, and agent-identity protocols now exist and are being deployed. What is missing is an executable environment that spans *principals*: one in which candidate primitives are measured against each other under adversarial conditions, where the party judging an effect need not be the party that authorized it, and in which the cost of decentralizing the governance layer is quantified rather than assumed away. We have presented a federated trust model with its attestation object and trust bootstrap stated explicitly, live primitive-level evidence bounded by what it actually exercised, and a pre-registered benchmark designed so that both positive and negative results are informative. The testbed, scenarios, traces, and metric scripts will be released under open licenses as they mature.

Artifact availability. The underlying governance runtime is open source (Apache-2.0) [1]. The captured trace of Section 5 — raw server responses, the three principal UUIDs, the capture timestamps, and the query behind the identity-store figure in Section 1 — is deposited with this paper's Zenodo record [32], so every claim in that section can be re-derived rather than taken on report. The two-host run of Section 5.1 — its implementation, unit tests, run script and transcripts — is deposited there as well [29]. Benchmark artifacts (scenario definitions, oracles, trace schemas, metric scripts) will be released under open licenses — Apache-2.0 for code, CC BY 4.0 for scenarios, oracles, and schemas — as the work described in Sections 6–7 completes.

Use of AI assistance. This manuscript was prepared with AI assistance under the author's direction. The author fixed its claims and scope, reviewed the final text in full, and is solely responsible for its content. AI assistants were used for drafting and editing prose, and for implementation work on the underlying governance runtime and the attestation testbed under the author's direction. The October 2 bibliography was verified against its live sources before deposit — arXiv API metadata (title, first author, publication year, author count) for preprints, and URL or DOI resolution with a title match otherwise; no citation was accepted on model output alone. Numeric and dated claims were checked against system records rather than generated, and the query behind the identity-store figure is recorded in the trace artifact. The trace and the dated citation-verification record are both deposited with this paper's Zenodo record. The added v1.1 registration reference was checked against its public, commit-pinned source on October 13, 2026.

References

[1] Kenny Wang. UNITARES: Information-theoretic governance of heterogeneous agent fleets. Zenodo, 2026. URL `https://doi.org/10.5281/zenodo.19647159`.

[2] Alan Chan, Noam Kolt, Peter Wills, Usman Anwar, Christian Schroeder de Witt, Nitarshan Rajkumar, Lewis Hammond, David Krueger, Lennart Heim, and Markus Anderljung. IDs for AI systems. *arXiv preprint arXiv:2406.12137*, 2024. URL `https://arxiv.org/abs/2406.12137`.

[3] Alan Chan, Kevin Wei, Sihao Huang, Nitarshan Rajkumar, Elija Perrier, Seth Lazar, Gillian K. Hadfield, and Markus Anderljung. Infrastructure for AI agents. *arXiv preprint arXiv:2501.10114*, 2025. URL `https://arxiv.org/abs/2501.10114`.

[4] Tobin South, Samuele Marro, Thomas Hardjono, Robert Mahari, Cedric Deslandes Whitney, Dazza Greenwood, Alan Chan, and Alex Pentland. Authenticated delegation and authorized AI agents. *arXiv preprint arXiv:2501.09674*, 2025. URL `https://arxiv.org/abs/2501.09674`.

[5] Abhishek Goswami. Agentic jwt: A secure delegation protocol for autonomous ai agents. *arXiv preprint arXiv:2509.13597*, 2025. URL `https://arxiv.org/abs/2509.13597`.

[6] Sunil Prakash. Aip: Agent identity protocol for verifiable delegation across mcp and a2a. *arXiv preprint arXiv:2603.24775*, 2026. URL `https://arxiv.org/abs/2603.24775`.

[7] Cloudflare. The agent access model. Cloudflare Blog, 2026. URL `https://blog.cloudflare.com/the-agent-access-model/`. Published 2026-08-05; accessed 2026-08-09. Industrial architecture composing an agent identity broker, task-scoped access engine, default-deny mediation at harness and network egress, and a monotone trust ratchet.

[8] Michael B. Jones, Anthony Nadalin, Brian Campbell, John Bradley, and Chuck Mortimore. OAuth 2.0 token exchange. RFC 8693, IETF, 2020. URL `https://www.rfc-editor.org/rfc/rfc8693`.

[9] Daniel Fett, Brian Campbell, John Bradley, Torsten Lodderstedt, Michael B. Jones, and David Waite. OAuth 2.0 demonstrating proof of possession (DPoP). RFC 9449, IETF, 2023. URL `https://www.rfc-editor.org/rfc/rfc9449`.

[10] Model Context Protocol Project. Model context protocol specification, revision 2026-07-28. Open specification, 2026. URL `https://modelcontextprotocol.io`. Open specification; revision 2026-07-28, accessed 2026-07-31. Both this and Agent2Agent Protocol Project [20] are moving specifications; the protocol-grounded scenario subset pins this revision, in which the base protocol is stateless and the initialization handshake and protocol session have been removed.

[11] John R. Douceur. The Sybil attack. In *Peer-to-Peer Systems: First International Workshop (IPTPS)*. Springer, 2002.

[12] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings 1996 IEEE Symposium on Security and Privacy*, pages 164–173, 1996. doi: 10.1109/SECPRI.1996.502679.

[13] Carl Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian Thomas, and Tatu Ylonen. SPKI certificate theory. IETF RFC 2693, 1999. URL `https://www.rfc-editor.org/rfc/rfc2693.txt`.

[14] 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 (NDSS)*, 2014. URL `https://research.google/pubs/pub41892/`.

[15] W3C. PROV-DM: The PROV data model. W3C Recommendation, 2013. URL `https://www.w3.org/TR/prov-dm/`.

[16] Norm Hardy. The confused deputy. *ACM SIGOPS Operating Systems Review*, 22(4):36–38, 1988. doi: 10.1145/54289.871709.

[17] W3C. Decentralized identifiers (DIDs) v1.0. W3C Recommendation, 2022. URL `https://www.w3.org/TR/did-core/`.

[18] W3C. Verifiable credentials data model v2.0. W3C Recommendation, 2025. URL `https://www.w3.org/TR/vc-data-model-2.0/`.

[19] 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`.

[20] Agent2Agent Protocol Project. Agent2agent (A2A) protocol. Linux Foundation open protocol project, 2025. URL `https://github.com/a2aproject/A2A`. Open protocol first published 2025-03-25; Linux Foundation project. Specification v1.0.1; accessed 2026-07-31.

[21] 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. *arXiv preprint arXiv:2406.13352*, 2024. URL `https://arxiv.org/abs/2406.13352`.

[22] Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, et al. Agentharm: A benchmark for measuring

harmfulness of llm agents. *arXiv preprint arXiv:2410.09024*, 2024. URL https://arxiv.org/abs/2410.09024.

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

[24] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024. URL https://arxiv.org/abs/2406.12045.

[25] Joel Z. Leibo, Edgar Duéñez-Guzmán, Alexander Sasha Vezhnevets, John P. Agapiou, Peter Sunehag, Raphael Koster, Jayd Matyas, Charles Beattie, et al. Scalable evaluation of multi-agent reinforcement learning with melting pot. *arXiv preprint arXiv:2107.06857*, 2021. URL https://arxiv.org/abs/2107.06857.

[26] Lewis Hammond, Alan Chan, Jesse Clifton, Jason Hoelscher-Obermaier, Akbir Khan, Euan McLean, Chandler Smith, Wolfram Barfuss, et al. Multi-agent risks from advanced ai. *arXiv preprint arXiv:2502.14143*, 2025. URL https://arxiv.org/abs/2502.14143.

[27] Shu Yang, Shenzhe Zhu, Hao Zhu, José Ramón Enríquez, Di Wang, Alex Pentland, Michiel A. Bakker, and Jiaxin Pei. Multi-user large language model agents. *arXiv preprint arXiv:2604.08567*, 2026. URL https://arxiv.org/abs/2604.08567. v1 2026-03-19; v2 2026-04-27.

[28] Ben Laurie, Adam Langley, and Emilia Kasper. Certificate transparency. IETF RFC 6962, 2013. URL https://www.rfc-editor.org/rfc/rfc6962.txt.

[29] Kenny Wang. Two-host attestation run: Implementation, tests and transcripts. Artifact two-host-attestation-20261002.zip, deposited with this paper, 2026. Runs 2026-10-02T21:57:27Z and 2026-10-02T22:24:26Z between a macOS host and a Raspberry Pi 4 Model B.

[30] Kenny Wang. Accountable multi-principal testbed: Evaluation pre-registration v1.1. https://github.com/cirwel/unitares/blob/49143add79ee630a60549f9ba79f1ca9b62b3423/docs/proposals/registered/accountable-testbed-metrics-preregistration-v1.md, 2026. Frozen 2026-08-05; supersedes v0 for future evaluation. Public source verified 2026-10-13.

[31] Kenny Wang. Accountable testbed metrics: Pre-registration v0. https://github.com/cirwel/unitares/blob/master/docs/proposals/accountable-testbed-metrics-preregistration-v0.md, 2026. Registered 2026-07-05; thresholds frozen at the merge commit.

[32] Kenny Wang. Preliminary trace: Federated multi-principal accountability primitives. Capture record preliminary-trace-2026-06-30.md, deposited with this paper, 2026. Captured 2026-06-30 against a deployed UNITARES governance server.