Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Authorization Assurance for Tool-Using LLM Agents: A Structured Review of Enforcement Semantics and Effect-Level Security Evidence

Mohamed Abbas Elmasry

malmasry@eelu.edu.eg

Egyptian e-Learning University

Research Article

Keywords: LLM agents, authorization assurance, access control, tool use, security evaluation, prompt injection

Posted Date: September 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10911223/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

Tool-using large language model (LLM) agents exercise authority over protected resources through APIs, tools, credentials, and external services. Existing evaluations often measure prompt resistance, tool selection, call admission, or benchmark success, but those observations do not necessarily justify stronger claims about preventing unauthorized protected-resource effects. This structured review introduces authorization assurance as a claim-centered lens relating asserted security properties to their required obligations, enforcement semantics, evidence, observation points, and residual assumptions. A 75-reference evidence library supported a closed analytical corpus of 30 full-text sources decomposed into 92 Source–Claim Records: a 15-source strict agent core, a six-source sensitivity layer, and nine authorization/assurance comparators. In the strict core, all three sources relevant to least privilege reduced available authority rather than measuring effective privilege across the execution trajectory; mediation claims spanned materially different enforcement and observation boundaries; and every source retained at least one material residual assumption. No evaluable revocation-assurance claim appeared among the 21 agent-focused sources, and no evaluable delegation-containment claim appeared in the 30-source corpus. The synthesis also shows that final-state correctness is not equivalent to authorization correctness and that assurance is compositional rather than determined by a single evidence modality. On this empirical basis, the review contributes the Minimum Authorization-Assurance Profile (MAAP), a novel evidence-derived analytical reporting profile that operationalizes claim scope, enforcement boundaries, effect-level observation, and residual assumptions for transparent comparison. MAAP is intended for diagnostic reporting and comparative appraisal; validation as a predictive metric or certification scheme remains future work.

1. Introduction

Large language models increasingly operate as action-capable components rather than text generators. Tool-using agents can retrieve private data, invoke APIs, change application state, communicate with external parties, maintain persistent memory, and coordinate multi-step actions [1, 11, 13, 14, 16, 35]. The security consequence is qualitative: an incorrect output may remain informational, whereas an agent action can exercise authority and create a persistent effect in a protected system.

Authorization is therefore a system property, not only a model-behavior property. The model proposing an action, the component holding credentials, the policy decision point (PDP), the policy enforcement point (PEP), the executor, and the protected resource may be different components. Classical access-control principles identify the principal, resource, operation, policy, enforcement boundary, and relevant context [66, 68–73]. Emerging protocols and operational guidance increasingly acknowledge the same boundaries for agentic systems [74, 75].

Prompt injection, persistent-state manipulation, unsafe tool use, privacy leakage, and compromised control flow can redirect the causal path from model input to external effect [10–13, 15, 17, 40, 41, 43, 48, 51, 65]. Instruction hierarchies, structured interfaces, information-flow control, tool gates, and agent

firewalls reduce parts of that risk [20, 47, 53, 60–64], but a mechanism label or benchmark score does not determine the strength of the resulting authorization claim. Hiding a tool may leave equivalent operations available through a shared tool; rejecting a call may leave another path to the resource; a correct final state may conceal a missing confirmation; and central revocation may precede actual downstream denial [7, 23, 46, 52, 57].

We call the resulting problem **authorization assurance**: justified confidence that authorization controls mediate the in-scope executable paths capable of producing specified protected-resource effects, under an explicit system boundary, threat model, and set of trusted assumptions. The core question is claim–evidence proportionality—whether the enforced semantics and reported observations justify the property at the scope in which it is stated. This adapts established security-assurance reasoning, which links claims, requirements, assumptions, verification conditions, and evidence [2, 3, 18, 19, 25], to the execution chain of tool-using agents.

Publication-level categorization alone cannot answer this question. One paper may provide a formal guarantee for an interpreter, an empirical benchmark result for a particular attack, and a broader discussion claim about the system as a whole. Treating the paper as uniformly “secure” or “effect-level” would erase the different boundaries of those claims. The review therefore changes the unit of synthesis from the publication to the individual source claim and preserves the relationship between that claim and the evidence used to support it.

The review addresses four research questions:

1. **RQ1**: What authorization-security properties are claimed, and under which principals, system boundaries, threat models, protected resources, and assumptions?
2. **RQ2**: Where and when are those properties enforced across delegation, planning, tool admission, execution, and protected-resource interaction?
3. **RQ3**: To what extent does reported evidence support the obligations implied by each claimed property?
4. **RQ4**: Which recurring claim–enforcement–evidence mismatches limit assurance, and what minimum reporting profile follows from them?

The review makes four novel, evidence-grounded contributions. First, it introduces the Authorization Assurance Claim Model, which relates the claimed property, principal, delegated authority, protected effect, PDP/PEP, evidence, and residual assumptions. Second, it operationalizes property-specific assurance obligations without imposing a universal hierarchy of evidence. Third, it decomposes 30 full-text sources into 92 Source–Claim Records (SCRs), allowing distinct claims within the same publication to receive appropriately bounded evidentiary interpretations. Fourth, it synthesizes the recurring corpus-level gaps into MAAP, a novel analytical reporting profile for transparent and comparable authorization-assurance reporting.

2. Scope and Positioning Against Prior Reviews

The surrounding literature already contains extensive survey activity; novelty cannot rest on claiming that authorization, agent security, or evaluation has not been reviewed. Broad LLM-security and safety surveys organize attacks, privacy risks, robustness failures, and defenses [10, 12, 15]. Agent-focused reviews cover threats, architectures, defensive mechanisms, evaluation frameworks, case studies, and real-world deployment concerns [11, 13, 14, 16, 17]. Domain and governance reviews extend this picture to autonomous cyber defence, trust/risk/security management, and regulatory anchoring [35–37]. These sources establish the field but generally retain the publication, threat, mechanism, or lifecycle stage as the principal unit of synthesis.

Related work on human–AI integration in cybersecurity emphasizes the operational setting in which AI-supported decisions and incident workflows are embedded [38]. Focused security surveys in the target journal also demonstrate that a bounded corpus can support a useful synthesis when the interfaces, components, and security properties under review are defined explicitly [39]. Neither precedent, however, replaces claim-level appraisal of agent authorization evidence.

Authorization research provides the semantic foundations used here. Contemporary studies address compound objects and temporal constraints [5], conflict and redundancy in policy models [6], dynamic risk state [8], cryptographic revocation and traceability [9], policy evaluation and mining [22, 26, 27], RBAC and information-flow structures [28, 29], and privacy, cloud, IoT, and spatiotemporal authorization [30–32]. Formal and cryptographic mechanisms show how policy can be bound to protocol or resource acceptance [33, 34]. These contributions do not evaluate heterogeneous LLM-agent claims; they supply comparators for interpreting privilege, mediation, delegation, and revocation.

Prompt-injection and tool-security research spans model-output attacks, online adversarial interaction, persistent memory, executable agents, structured interfaces, and realistic web environments [40, 41, 43, 48, 51, 58, 59, 65]. More recent mechanisms move selected controls outside unconstrained model reasoning through tool-dependency constraints, capability/provenance tracking, information-flow policy, or independent interception [47, 50, 53, 54]. Recent syntheses, system studies, and design proposals foreground delegated authority, inter-agent communication, security patterns, and system-level control [56, 57, 60–64]. Prompt injection can therefore be a cause of authorization failure, but influence, tool selection, call submission, execution, and attacker-defined resource state remain different observations.

Agent benchmarks make this distinction concrete. AgentDojo [40], ToolEmu [42], ASB [44], AgentHarm [45], τ-bench [46], ToolSandbox [52], and WASP [58] observe progressively richer combinations of calls, trajectories, and environment state. Yet richer state does not automatically encode identity, consent, delegation, revocation, or every process condition that makes a transition authorized. Security-assurance research provides the closest methodological precedent by treating claims, arguments, assumptions, verification, and evidence as traceable but non-interchangeable artifacts [2, 18, 19, 24, 25]. Table 1 positions this review at their intersection.

Table 1
Positioning against representative prior-review families

Prior-review family	Primary organizing unit	Distinctive role of this review
Broad LLM security and safety [10, 12, 15]	Attack, vulnerability, privacy, robustness, defense	Restricts synthesis to authorization claims in action-capable systems
LLM-agent security and architecture [11, 13, 14, 16, 17]	Threat, architecture, lifecycle, mechanism, evaluation	Uses the individual authorization claim as the analytical unit
Cyber-defence, TRiSM, and safety reviews [35–37]	Domain, governance, operational risk	Focuses on technical claim–enforcement–evidence alignment
Security-assurance reviews [3, 18, 19]	Claim, requirement, argument, verification evidence	Adapts assurance reasoning to tool-mediated protected effects
Frontier agent-security synthesis [56]	Threat surface, delegated authority, state, evaluation	Used for qualitative positioning, not peer-reviewed denominators

The review is therefore neither a general agent-security taxonomy nor a new access-control architecture. Its distinctive contribution is a claim-level authorization-assurance appraisal that connects each asserted property to its obligations, enforcement boundary, observed protected effect, and residual assumptions.

3. Review Methodology

3.1 Review design and search boundary

We conducted a structured, claim-centered security review of authorization assurance in tool-using LLM agents. The search and eligibility window closed on 15 July 2026. Search concepts intersected four families: (i) tool-using, autonomous, web, computer-use, and multi-agent systems; (ii) access control, authorization, privilege, delegation, capability, credentials, least privilege, confused deputy, and revocation; (iii) PDP/PEP, gateways, resource-side controls, policy enforcement, and information flow; and (iv) security assurance, formal guarantees, benchmarks, sandbox execution, final-state evaluation, and deployment evidence. Sources were located through major computer-security, machine-learning, and information-systems indexes and publisher repositories, supplemented by backward and forward citation tracing.

The 75-reference evidence library supports framing, prior-review positioning, standards, foundational semantics, mechanisms, benchmarks, and discussion. It is not a quantitative denominator. Pure text-only chatbot studies, generic content-safety or jailbreak studies without an execution/authorization dimension, tool-learning studies without security relevance, and governance commentary without extractable technical semantics were excluded from claim coding.

Eligibility was assessed against the analytical function of the source rather than venue prestige alone. Agent papers had to expose enough of the action path to identify at least a protected resource, an enforcement or evaluation boundary, and evidence relevant to a security claim. Authorization comparators had to contribute a transferable obligation—such as path coverage, effective privilege, resource-side revocation, or claim–verification traceability—rather than merely share access-control terminology. This rule allowed classical and adjacent sources to inform interpretation without allowing them to inflate claims about contemporary agent practice. Reviews, standards, and horizon literature could enter the 75-reference library while remaining outside the 30-source claim-coded corpus.

3.2 Layered analytical corpus

Thirty sources were selected for full-text coding and frozen after a 12-source pilot. Layer A required a peer-reviewed, agent-specific source with extractable security claims and sufficient enforcement or evidence semantics to support claim-level appraisal. Layer B included peer-reviewed technical sources relevant to the analysis but weaker on at least one strict dimension—agent specificity, authorization specificity, or an effect-level security objective. Layer C contained non-agent or adjacent authorization/assurance comparators used to derive or test transferable obligations. Preprints were retained only for horizon positioning.

Table 2
Analytical corpus and inferential roles

Layer	Sources	SCRs	Inferential role
A—strict agent evidence core	15	47	Primary source-level denominator for agent findings
B—expanded sensitivity evidence	6	18	Added only in explicitly identified A + B sensitivity analyses
C—authorization/assurance comparators	9	27	Derives or tests obligations; excluded from agent proportions
Total	**30**	**92**	Closed claim-coded evidence base

Pilot coding was conducted from 16–28 July 2026 and generated 38 SCRs across six property families, exceeding the predefined feasibility gate of 25 records across four families. The corpus was frozen on 28 July; the remaining 18 sources were coded from 29 July–15 August 2026.

3.3 Source–Claim Records and appraisal

The SCR, rather than the publication, is the unit of analysis. One source can yield multiple records when it advances claims with different properties, enforcement boundaries, or evidence. Each SCR records the paraphrased source claim and location; normalized property and scope; system boundary and threat model; principal and delegation relation; protected resource and privileged effect; PDP, PEP, and enforcement timing; bypass, failure, and revocation semantics; evidence type and observation point; supported and unsupported obligations; residual assumptions; and a qualitative appraisal note. Claim

scope was retained descriptively rather than converted into an unimplemented explicit/implied classification.

Extraction separated what authors stated from the review's normalized interpretation. The source-claim field preserved the claim in paraphrase and recorded its section or page location; normalization mapped heterogeneous terminology to the property whose obligations were being evaluated. Mechanism functionality was not silently promoted to a stronger guarantee. For example, tool filtering could be coded as authority reduction without being treated as proof of operation-level least privilege, and a final-state benchmark could be coded as effect-proximate without being treated as complete authorization-process evidence. This distinction was essential because several publications offered multiple relevant claims under one architecture.

For claim (c), the required obligations, evidenced obligations, and residual assurance gap are:

$G(c) = O(c) - E(c)$

O(c) is the set of obligations required by the claimed property; **E(c)** is the set supported by reported evidence; and **G(c)** is the residual assurance gap—the obligations in O(c) that are not supported in E(c).

G(c) is a set, not a scalar score. Evidence type and observation point were coded separately because formal proof, classification, call observation, sandbox execution, final state, and deployment observation answer different questions. The record-level appraisal labels summarize supported scope or the main evidentiary boundary; they are non-ordinal and are not used as an aggregate security score. Primary findings instead rely on the normalized property, enforcement locus, observation point, O(c), E(c), G(c), residual assumptions, and explicit source-level denominators. An assurance gap indicates insufficient evidence for the broader claim, not demonstrated insecurity.

3.4 Quality assurance and synthesis

After primary coding, a stratified sample of 20/92 SCRs (21.7%) was independently double-coded using the frozen framework. Four judgment-bearing fields were evaluated. Pre-adjudication agreement was 100% for normalized property ($\kappa = 1.000$), 90% for enforcement locus ($\kappa = 0.861$), 90% for observation stage ($\kappa = 0.885$), and 95% for effect-level evidence ($\kappa = 0.899$). Five disagreements concerned adjacent enforcement/observation boundaries or the treatment of benchmark-bounded effects; adjudication did not change corpus-level denominators or the principal findings.

Layer A is the primary agent denominator; A + B is reported only as expanded sensitivity. Layer C never enters agent-level proportions. Multiple SCRs from one source are not treated as independent studies, so source-level denominators are used for prevalence statements. A result is elevated from description to security finding only when the coding supports the chain: pattern, denominator, property-specific obligation, observed evidence, assurance gap or alignment, and security implication.

No pooled inferential statistic was calculated over the 92 SCRs. Claim-level counts describe the composition of the evidence base; they are not estimates obtained from independent observations. Zero counts are likewise interpreted as absence of an evaluable claim in the closed corpus, not proof that the property is absent from every agent system. The review reports ratios only when the numerator and denominator refer to the same property and corpus layer.

Use of generative AI. OpenAI's ChatGPT and Codex were used to assist with language editing, structural refinement, and document preparation. The author reviewed all AI-assisted output, verified the final manuscript against the coded evidence base, and accepts full responsibility for the content.

4. Authorization Assurance Claim Model

Authorization assurance is interpreted relative to an operational execution chain and an evidentiary claim chain. A **principal** is the entity whose authority is recognized; an **actor** is the agent, sub-agent, service, or executor exercising it. Delegation is the bounded exercise or transfer of authority on behalf of another principal, as made explicit in OAuth subject–actor and token-exchange relationships [71, 72]. A **protected resource** is data, a service, account, API, file, credential, infrastructure, or authorization state. A **privileged effect** is a security-relevant operation or state transition over that resource—such as reading, disclosing, modifying, deleting, communicating, executing, transferring, or granting access.

This endpoint matters because intermediate behavior is not the effect. Generating SQL is not executing it against a database; selecting an email tool is not sending a message; and producing transaction parameters is not committing the transaction. The PDP determines whether an operation should be permitted, while the PEP ensures that the decision governs execution [69, 70]. Either may appear at the model, orchestrator, tool registry, gateway, credential broker, executor, API, or protected resource. A correct decision does not provide resource-level assurance if the operation can travel through an unmediated path.

Authorization is also temporal. Controls may operate at provisioning, task start, planning, tool admission, execution, resource access, or after revocation. Pre-task capability reduction can reduce authority without constraining arguments or capabilities acquired later. Similarly, a revocation event establishes a control-state change; cessation of authority requires subsequent denial at the protected resource [7].

The analytical chain is:

Principal → Delegation → Actor/Agent → PDP decision → PEP enforcement → Tool/Executor → Protected resource → Privileged effect

Table 3
Operational elements and inferential boundaries

Element	Coding question	Inferential boundary
Claim and property	What is asserted, prevented, or guaranteed?	A mechanism label is not itself an assurance claim
Principal and delegation	Whose authority is exercised, by whom, and under what scope/duration?	User intent does not automatically bind downstream execution
Protected resource/effect	What security-relevant state transition matters?	Tool selection or submission may precede the realized effect
PDP and PEP	Where is permission decided and where is it enforced?	Decision correctness does not prove path coverage
Enforcement timing	Is authority constrained at provisioning, task, call, execution, resource, or revocation time?	Static reduction is not per-operation authorization
Evidence and observation	What is proved/tested, and at which stage?	Later stages are not universally superior; fit depends on the claim
Residual assumptions	What remains trusted, synchronized, non-bypassable, or outside attacker control?	Assumptions bound assurance but do not automatically invalidate it

Evidence is located at six stages: plan, tool selection, tool call, execution, final state, and deployment effect. Plan evidence can support planning compliance but not attempted execution. Tool-call evidence can support interception but not successful or prevented downstream state. Final-state evidence can establish a tested state transition while remaining silent about consent, identity, sequencing, or delegation. Deployment evidence extends to persistent, cross-step, post-revocation, or operational consequences. These stages locate evidence; they do not form a universal quality ladder.

Residual assumptions are consequently first-class elements of the model. Typical assumptions include complete visibility of relevant tools, correct capability labels, non-bypassable enforcement, faithful tool execution, synchronized policy state, non-copyable credentials, honest external services, and reliable human approval. An assumption does not automatically weaken a guarantee: a narrow formal property can be strong under an explicit model, and a controlled resource experiment can be strong for the tested operation. The assurance boundary becomes ambiguous only when unreported assumptions are needed to extend evidence beyond what was modeled or observed.

5. Property-Specific Assurance Obligations

The model becomes appraisable only when the claimed property determines the obligations in (O(c)). These obligations are claim-contingent: a policy-classification study need not demonstrate revocation unless it claims revocable authority, whereas complete mediation or least privilege requires evidence

beyond ordinary decision accuracy. The obligation set synthesizes classical least privilege, complete mediation, fail-safe defaults, and confused-deputy reasoning [66, 67]; RBAC/ABAC and PDP–PEP semantics [68–70]; delegated authorization and sender-constrained authority [71–73]; and assurance traceability [2, 18, 19, 24].

Three distinctions carry most of the analytical weight. First, least privilege concerns effective authority, not only the number of visible tools:

Task-required(c) ≤ Exercised(c) ≤ Available(c) ≤ Granted(c)

Second, complete mediation requires every in-scope path to the specified effect to traverse the PEP; interception of one gateway proves only mediation of that path. Third, evidence must match the claimed effect. Model classification may fully support a classification claim, call rejection may support an interception claim, protected-state observation is normally necessary for a state-prevention claim, and post-revocation access is necessary for a claim that authority has ceased.

Authorization correctness requires an identifiable policy relation among principal, resource, operation, and relevant context. When a study claims only decision accuracy, policy labels or allow/deny predictions may be sufficient; when it claims enacted compliance, the evidence must also show that the decision governed the corresponding operation. This separates policy understanding from enforced policy behavior.

Non-bypassability concerns the integrity of the PEP rather than only the number of paths it covers. A defensible claim must explain why the agent cannot disable the control, detach credentials from the authorized decision, mutate security-relevant arguments after approval, or reach an equivalent interface. Fail-safe enforcement adds a different obligation: missing identity, stale policy, timeouts, malformed calls, partial execution, or unavailable authorization services must resolve to denial or another demonstrably safe state within the stated boundary.

Delegation containment requires the downstream actor to remain within the authority granted by the originating principal. Nominally smaller permission sets are insufficient if a sub-agent can reproduce the same effect through credential forwarding, actor substitution, shared tools, or another delegation path. Temporal validity similarly requires evidence after authorization changes. Revocation-list insertion, policy update, or central token invalidation establishes that a control event occurred; it does not by itself show when the protected resource stopped accepting the authority [7, 34].

Accountability is supporting rather than preventive assurance. A log can help establish that a decision was applied and reconstruct a failure, but a tool-call record without principal, policy, credential, resource, and resulting effect is not a complete authorization trace. These distinctions keep the obligation model property-specific while allowing formal, empirical, and operational evidence to contribute complementary support.

Table 4
Property-specific obligations and minimum claim-aligned evidence

Claimed property	Core obligations	Evidence needed for the stronger claim	Insufficient proxy
Authorization correctness	Policy, principal, resource, operation, context, correct allow/deny, decision bound to action	Decision evidence for decision claims; execution/state evidence for enacted compliance	Policy QA or classifier accuracy without enforced execution
Least privilege	Task-required versus granted/available/exercised authority; argument/resource granularity; adjacent or dynamic authority	Effective-authority comparison across in-scope operations	Smaller tool list without showing task-minimal remaining authority
Complete mediation	Protected effect; all in-scope paths; mandatory PEP traversal; alternative paths; active enforcement	Path-coverage argument plus observation appropriate to the effect	Rejection at one gateway with untested routes
Non-bypassability	PEP outside agent discretion; decision bound to execution; credential and alternate-route resistance	Formal argument or adversarial bypass testing within the threat model	Correct behavior only through the intended interface
Fail-safe enforcement	Defined failures; deny/safe default; safe partial operations and recovery	Failure injection or formal coverage of material failure modes	Nominal-operation accuracy
Delegation containment	Original principal; delegated scope; attenuation; chain integrity; credential binding; redelegation and expiry	Downstream authority/effects remain within delegated scope	Fewer named permissions without effect-level containment
Temporal validity/revocation	Revocation event and effective time; stale/cached authority; propagation; latency; in-flight operations	Post-revocation execution or resource-side denial	Central invalidation without subsequent access test
Accountability/traceability	Link principal, delegation, policy, PEP, operation, credential, time, and effect	Reconstructable and integrity-protected principal–policy–action–effect trace	Tool-call log without effect or provenance

The table does not require every source to address every property. It prevents both over-claiming—inferring an effect-level guarantee from an upstream proxy—and over-penalization—judging a narrow, correctly evidenced claim against a stronger property it never asserted.

6. Landscape of Authorization Claims

The 21 agent-focused sources comprised nine benchmark/evaluation sources [40, 42–46, 51, 52, 58], ten mechanism, control, authorization, or policy-engineering sources [4, 20, 21, 47–50, 53–55], and two threat/risk studies [1, 41]. These roles overlap conceptually but describe why each source entered the analysis. Layer C supplied comparators only.

Table 5
Property coverage in the agent-focused corpus

Property/dimension	Layer A sources	Layer A + B sources	Principal interpretation
Policy compliance	1	4	Decision and process semantics are sparsely explicit
Least privilege/data minimization	3	3	Pre-execution authority or data narrowing
Complete mediation/non-bypassability	5	5	Different enforced boundaries and observation points
Fail-safe behavior	1	2	Rarely evaluated as an independent property
Delegation containment	0	0	No evaluable agent-focused claim
Revocation/temporal validity	0	0	No evaluable agent-focused claim
Confidentiality/integrity information flow	4	4	Wrapper, guard, interpreter, or structured interface
Evidence-to-effect alignment	15	21	Cross-cutting dimension represented in every agent source

Across the 47 Layer-A SCRs, 14 concerned preventive properties—five complete mediation/non-bypassability, four information-flow control, three least privilege/data minimization, one policy compliance, and one fail-safe behavior. The remaining 33 concerned evidence-to-effect alignment, threat/boundary coverage, reliability, or claim–enforcement traceability. Expansion to A + B produced 65 agent-focused SCRs but no revocation or delegation-containment record.

The imbalance is substantively informative but must be interpreted carefully. It shows that the selected agent evidence base says much more about attacks, trajectories, calls, and evaluation boundaries than about several classical authorization properties. It does not show that every paper should have claimed

revocation, delegation containment, or fail-safe behavior. Rather, it identifies which assurance questions cannot be answered from the claims and observations currently represented in the corpus.

Systems ranged from model-facing structured interfaces to wrappers, tool registries, runtime guards, persistent memory, executable tools, and stateful web environments. Delegation was usually implicit in a user assigning a task while the agent exercised application capabilities, rather than represented as an authorization object with scope, attenuation, redelegation, and revocation. Threat models also differed: injected content [43, 48, 50, 51, 53, 58], poisoned memory [41], underspecified benign tasks [42], direct malicious requests [45], and trusted planner, metadata, interpreter, or human assumptions [47, 53, 54]. These differences preclude treating all SCRs as measurements of one homogeneous security construct.

Protected resources included private information, memory and retrieval state, tool/application databases, authenticated web resources, authorization labels, and policy representations. Corresponding effects ranged from disclosure to message transmission, record modification, harmful multi-step completion, and attacker-defined web state. This diversity is not noise to be normalized away: it determines whether a reported observation is semantically close enough to the effect named by the claim.

7. Enforcement Semantics

Enforcement ranged from early authority reduction to runtime interception and resource/cryptographic acceptance. AgentDojo filters tools before untrusted content is processed [40]; AirGapAgent minimizes data disclosed to an untrusted conversational component [49]; and IPIGuard constrains a precomputed command-tool dependency graph while permitting limited query-tool expansion [50]. These are meaningful reductions, but tool availability, accessible data, and operation-level authority are not equivalent.

Runtime mechanisms place the control closer to execution. The IFC guard in [47] checks consequential calls and permits, blocks, or escalates them. CaMeL mediates calls through a capability/provenance interpreter [53]. The design in [54] requires an independent interceptor and evaluates modeled traces. Moving control outside unconstrained agent reasoning improves the non-bypassability argument, but each guarantee remains bounded to instrumented tools, registered capabilities, correct metadata, or the modeled architecture.

Tool granularity is a recurring boundary. A shared tool may perform authorized and unauthorized operations depending on target, argument, state, or time. AgentDojo notes the limitation when benign and malicious goals require the same tool [40]; IPIGuard remains dependent on tool classification and graph completeness [50]; and ASB’s simplified calls can omit argument semantics [44]. Conventional access control makes these dimensions explicit and demonstrates that a valid policy can retain residual over-grant [5, 23, 69].

Five Layer-A sources produced mediation/non-bypassability claims [47, 49, 50, 53, 54]. All provided controls stronger than prompt-only instruction, but none supports an unbounded field-wide conclusion. Uninstrumented APIs, unregistered tools, hidden resource channels, mutable metadata, or trusted-component compromise remain outside some boundaries. Fail-safe behavior was even sparser: [20] assigns missing labels the most restrictive value, while [55] addresses ambiguity in policy translation rather than runtime fail-closed execution. No agent source evaluated downstream revocation.

The PDP–PEP distinction also separates policy specification from enforced execution. PPLLA establishes privacy-preserving attribute matching within an LLM gateway under an explicit cryptographic threat model [21], but downstream tool effects remain outside that protocol. The natural-language policy study in [55] evaluates extraction of entities and relations without executing the resulting rules against a protected resource. Conversely, Layer-C mechanisms such as distributed version-control access control [33] bind decisions more directly to cryptographic read/write acceptance. The relevant appraisal question is therefore not whether a policy exists, but whether its decision is bound to the operation whose effect is claimed.

Enforcement timing creates another non-equivalence. A control applied before task execution may prevent unnecessary capabilities from entering the interaction, while a call-time PEP can incorporate current arguments and context. Resource-time enforcement can remain independent of the agent's reasoning, but it may know less about task intent or upstream delegation. No location is automatically sufficient: assurance depends on whether the location has the information and control required by the claimed property. Robust designs may therefore compose early minimization, call-time checks, and resource-side authorization rather than asking one layer to carry the entire assurance argument.

Table 6
Representative enforcement patterns and assurance boundaries

Pattern	Sources	PEP location	Supported scope	Main boundary
Pre-task tool reduction	[40]	Tool registry	Bounded authority reduction	Shared tools and arguments remain
Pre-interaction data minimization	[49]	Trusted data boundary	Bounded disclosure/least privilege	Minimizer correctness and hidden access
Tool-dependency restriction	[50]	Agent executor	Bounded mediation/least privilege	Classification and graph completeness
Structured instruction/data separation	[48]	Front end plus model	Interface-level integrity	Model remains part of the control
Runtime IFC guard	[47]	Instrumented calls	Call-time information-flow mediation	Consequential effects must use instrumented tools
Capability/provenance interpreter	[53]	Interpreter	Formal and implemented bounded mediation	Registered tools, metadata, interpreter trusted
Independent formal interceptor	[54]	Modeled call boundary	Non-bypassability in the model	Runtime implementation pending
Gateway/cryptographic comparator	[21, 33, 34]	Gateway or resource/protocol boundary	Authorization or revocation within protocol scope	Downstream agent-tool effects may be outside scope

The central enforcement result is therefore semantic rather than numerical: policy specification, decision correctness, capability reduction, call interception, and resource-side denial are distinct security propositions.

8. Security Evidence and Observation Points

Evidence in the corpus included conceptual risk analysis, formal reasoning, offline model or policy evaluation, call-level benchmarks, emulated execution, stateful environments, and rule-verified effects. Formal evidence can strongly justify an in-model property without establishing implementation correspondence; executable evidence can demonstrate concrete behavior without proving universal path coverage [2, 18, 19].

Offline evidence terminated at prompt classification, model response, or generated policy in [4, 43, 48, 55]. ASB and InjecAgent observed tool selection or parsed invocation [44, 51]. ToolEmu executed stateful emulated trajectories and validated realism through human judgment [42]. AgentDojo, τ-bench, and ToolSandbox inspected environment or database state [40, 46, 52], while WASP separated an intermediate hijack judge from a rule-based attacker-goal predicate [58]. AgentPoison similarly separated poisoned retrieval, target action, and end-to-end impact [41].

No Layer-A or Layer-B source provided longitudinal production-deployment authorization evidence. This absence bounds generalization but does not make controlled evidence weak. A rule-based final-state oracle can strongly establish whether a defined effect occurred in the tested environment while remaining silent about other environments, causal paths, or unencoded authorization conditions.

Effect proximity and semantic completeness must also be separated. A final-state checker can directly observe whether a database changed, but it may not encode which principal initiated the action, whether the user confirmed it, whether the correct policy version applied, or whether the change followed a permitted sequence. Conversely, a formal proof at a call boundary can provide strong coverage of modeled traces while remaining upstream of real service behavior. Security evidence is strongest when these distinct contributions are connected rather than treated as substitutes.

The same distinction applies within empirical studies. A parsed model output may be a reliable representation of an attempted call yet remain dependent on the assumption that the call would execute. An emulated environment can test stateful trajectories while depending on fidelity to external services. A self-hosted application can verify a concrete final state while remaining bounded to enumerated actions and attacker goals. These are not defects in the studies; they are boundaries that determine which authorization claim the evidence can justify.

Evidence adequacy therefore depends on the claim. Detector output can support detector accuracy [4]; call admission can support interception [47, 53]; protected-state observation is necessary for a claim about record modification [46, 52, 58]; and post-revocation resource access is necessary for a claim that authority ended [7]. This claim-relative interpretation replaces a universal evidence ladder.

9. Cross-Cutting Authorization-Assurance Findings

Finding 1: mediation evidence does not uniformly terminate at the protected effect

Five strict-agent sources contained complete-mediation or non-bypassability claims [47, 49, 50, 53, 54]. Their controls constrained information flow, disclosure, tool graphs, calls, or modeled traces, but their observation boundaries differed. CaMeL provides a comparatively strong formal-plus-implementation argument within its interpreter and registered-tool boundary [53]; [54] provides modeled intercepted-trace evidence without runtime implementation. The sources therefore support different bounded claims, not one binary conclusion that complete mediation was demonstrated. **Implication**: call-level interception

justifies mediation of the tested path; effect-level complete mediation additionally requires an exhaustive, non-bypassable enforced boundary.

This finding is not an argument that effect-level evidence is mandatory for every mediation statement. A precisely scoped claim that all registered calls traverse an interpreter can be supported through architectural and formal evidence at that boundary. The mismatch arises only when this bounded result is extended to every path capable of producing the protected-resource effect without evidence that the interpreter is unavoidable. Claim language, not a universal preference for later-stage evidence, determines the required observation.

Finding 2: least privilege is operationalized as authority narrowing

Exactly three Layer-A sources generated least-privilege/data-minimization SCRs [40, 49, 50]. All three reduced tools, data, or command paths before or early in execution. Thus, 3/3 relevant sources implemented reduction of available authority or information rather than direct measurement of effective privilege across the complete trajectory. This is a meaningful contribution, but shared tools, arguments, dynamic expansion, and adjacent resource instances can retain unnecessary authority. **Implication**: stronger claims should compare task-required, granted, available, and exercised authority and test the adjacent operations that remain reachable.

The denominator is deliberately property-specific and small. The result should not be paraphrased as “agent research generally implements least privilege through tool filtering.” Its defensible scope is that every evaluable least-privilege claim in the strict corpus relied on early authority narrowing, and none supplied a trajectory-wide effective-authority measure. That narrower finding is still valuable because it identifies the measurement needed to distinguish surface reduction from task-minimal authority.

Finding 3: revocation and delegation containment are coverage gaps

No evaluable revocation-assurance claim appeared in Layer A or after expansion to all 21 agent-focused sources (0/21). No source in the complete 30-source analytical corpus generated an evaluable delegation-containment or attenuation claim (0/30). These are corpus-bounded coverage gaps, not evidence that deployed systems lack such controls. Layer-C evidence clarifies the missing test: [7] measures the interval between central revocation and downstream denial, while [8, 34] formalize temporal or cryptographic revocation under different boundaries. OAuth, token exchange, and sender-constrained credentials distinguish resource owner, subject, actor, and client [71–73]. **Implication**: future agent claims require post-revocation attempts, stale-credential tests, propagation to sub-agents, actor substitution, and downstream resource observation.

The gaps are particularly important because delegation is already implicit in most agent tasks: a user authorizes work, the agent selects actions, and services execute them using user- or application-controlled credentials. What is missing is not delegated action as an operational fact, but a claim and

evaluation showing that authority remains attenuated across handoffs and ceases when the upstream grant changes. This distinction keeps the finding empirical rather than speculative.

Finding 4: final-state correctness is not authorization correctness

τ-bench demonstrates that the expected database state can be reached through a policy-violating process, including omission of required confirmation [46]. ToolSandbox supports milestone and minefield constraints but does not generally encode mandatory authentication and confirmation [52]. Therefore:

Final-state(c) = Pass ⇏ Authorization-correct(c) = Pass

Effect proximity and semantic completeness are independent. Final state cannot establish an identity, consent, ordering, temporal, or delegation obligation absent from the oracle; call evidence can establish admission without confirming the final effect. **Implication**: authorization benchmarks need both protected-state predicates and process predicates that define why the transition was authorized.

The converse matters as well. A process trace can show that the expected confirmation was obtained and the correct call was admitted, but an external service may fail, retry, partially commit, or create a different final state. Authorization assurance therefore benefits from two linked oracles: one over the legitimacy of the decision and process, and one over the realized resource consequence.

Finding 5: residual assumptions bound every strict-agent argument

At least one material residual assumption was identified in every Layer-A source (15/15). Examples included correct labels and complete wrapper visibility [20], fidelity of benchmark predicates and tool semantics [40, 44, 45, 58], emulator realism [42], trusted planner or human approval [47], preserved structured channels [48], absence of hidden access [49], side-effect-free query tools [50], accurate simulated execution [51], sound capability metadata and a trusted interpreter [53], and correspondence between a bounded model and runtime implementation [54]. Assumptions are not evidence of weak security; every assurance argument has a trusted boundary. **Implication**: the material distinction is whether the assumption is explicit and bounded or silently required to extend the evidence.

This finding also prevents a misleading comparison between formal and empirical work. A formal study may state its trusted model explicitly, while a benchmark may embed comparable assumptions in an emulator, rubric, or goal predicate. Recording assumptions symmetrically makes both forms of evidence interpretable without penalizing one simply for being explicit.

Finding 6: strong assurance is compositional

The corpus does not support a ranking in which formal proof, sandbox execution, or final-state evidence is always superior. CaMeL combines formal semantics, implemented enforcement, and benchmark testing [53]; [47] combines a runtime guard with benchmark evidence; [54] offers meaningful formal

evidence with implementation pending; WASP offers strong rule-verified effects in defined environments without a universal proof [58]. **Implication**: the strongest argument is a traceable correspondence among property, obligations, enforcement, evidence, protected effect, and residual assumptions. No modality substitutes for links it does not observe.

Compositional assurance does not require every study to conduct every evaluation. It requires that the supported claim remain within the combined boundary of its evidence and that missing links be visible. Later work can then add implementation correspondence, bypass testing, resource-side observation, or deployment evidence without discarding the valid contribution of the earlier result.

Table 7
Cross-cutting findings and denominators

Finding	Basis	Main inference
Mediation evidence is not uniformly effect-level	Five Layer-A sources	Do not generalize beyond the enforced boundary
Least privilege is authority narrowing	3/3 relevant Layer-A sources	Tool/data reduction supports a bounded claim
No agent revocation-assurance claim	0/21 A + B sources	Coverage gap, not demonstrated system failure
No delegation-containment claim	0/30 analytical sources	Coverage gap requiring direct evaluation
Final state ≠ authorization correctness	Counterexample synthesis [46, 52]	State and process predicates are both required
Residual assumptions are universal	15/15 Layer-A sources	Guarantee scope depends on explicit assumptions
Assurance is compositional	[47, 53, 54, 58] plus [2, 18, 19, 24]	Formal, runtime, and effect evidence are complementary

10. Benchmarks, Auditability, and Reproducibility

Benchmark meaning depends on the oracle. AgentDojo combines deterministic task checks, environment mutations, attacker success, and benign utility [40]. ASB operationalizes many attacks through attack-specific tool use and simplified calls [44]. InjecAgent parses harmful or exfiltration-oriented calls and simulates responses [51]. AgentHarm scores multi-step completion through proxy tools and task-specific rubrics [45]. ToolEmu adds stateful emulation and realism validation [42]. These designs answer different questions and should not be collapsed into a single attack-success construct.

Stateful benchmarks improve effect observability but can omit authorization semantics. τ-bench checks final database state and selected outputs and uses (pass^k) to expose repeated-run instability [46]; ToolSandbox represents multiple valid trajectories through milestones and prohibited intermediate conditions [52]; WASP distinguishes intermediate diversion from a rule-verified attacker goal [58]. A

benchmark can therefore be close to the effect yet incomplete with respect to identity, consent, delegation, or policy version.

Table 8
Benchmark assurance coverage

Benchmark	Primary oracle	Effect proximity	Key authorization boundary
AgentDojo [40]	Deterministic task checks, environment mutations, targeted ASR	High within benchmark	Goal predicates and tool semantics encode the effect
ToolEmu [42]	Emulated stateful trajectory plus evaluator and human validation	Medium–high, simulated	Emulator fidelity to real tool semantics
ASB [44]	Attack-specific tool use and simplified calls	Call-level	Arguments and realized effects may be omitted
AgentHarm [45]	Multi-step proxy trajectory and task rubrics	High within proxy	Synthetic rather than production effect
τ-bench [46]	Final database/output conditions and (pass^k)	High	Final state can omit consent and sequence
InjecAgent [51]	Parsed harmful/exfiltration calls and simulated responses	Call/simulated execution	No independent effect confirmation
ToolSandbox [52]	Intermediate/final milestones and minefields	High	Authentication/confirmation not generally encoded
WASP [58]	Rule-based final attacker state plus intermediate judge	High	Bounded to tested sites and attacker goals

Trace availability is not itself an auditability guarantee. A complete authorization trace would link principal, delegation, policy and version, PDP decision, PEP execution, credential, operation arguments, and protected effect. Existing benchmarks expose fragments of this chain, but none in A + B claims complete authorization auditability; the only strict-corpus claim–enforcement traceability record arose from [54]. Assurance and certification comparators show why linking claims to verification evidence and later vulnerability information matters [2, 24].

Reproducibility likewise depends on preserved security semantics, not only released code. Tool schemas, simulated responses, database state, model versions, predicates, rubrics, and evaluator prompts can change what a score means. Portability therefore requires re-establishing the link between the benchmark oracle and the protected effect. The benchmark synthesis yields three conclusions: task success, attack success, and authorization compliance are distinct; effect proximity and semantic completeness are orthogonal; and benchmark evidence is an assurance component, not an assurance conclusion.

Repeated-run behavior deserves separate attention. Only τ-bench generated a dedicated strict-corpus reliability record through (pass^k) [46]. A control that blocks an unauthorized effect in 95% of runs but intermittently allows it has a different security interpretation from an agent that merely fails benign tasks in the remaining runs. Mean task or attack scores can obscure this asymmetry; security-sensitive evaluation should report failure distributions and worst-case or repeated-run behavior where feasible.

Auditability also affects reproducibility. If a benchmark records only a final score, later users cannot determine whether a difference arose from planning, call admission, tool execution, evaluator behavior, or resource state. Traces that preserve policy version, arguments, ordering, and state transitions allow both security interpretation and experimental replication. This does not require production logging in every benchmark, but it does require enough provenance to reconstruct why the oracle classified a trajectory as authorized, unauthorized, successful, or failed.

12. Discussion and Research Implications

The review shifts attention from policy presence to defensible authorization claims. Mechanism designers should separate behavioral compliance from mandatory enforcement and make PDP–PEP binding, argument/resource granularity, alternative execution paths, and the trusted computing base explicit. Prompt-level training and structured prompting can reduce unsafe behavior [43, 48], but they should not be equated with an external PEP unless the architecture prevents the model from bypassing the decision. Runtime IFC, capability/provenance enforcement, and independent interception show promising patterns [47, 53, 54].

For least privilege, design should move from enumerating tools toward modeling effective authority. Relevant questions include which resource instances and argument values remain reachable, whether read-oriented tools have hidden side effects, whether a tool registry can expand during a task, and whether another credential or sub-agent can reproduce the same operation. This is the agent analogue of residual-overgrant analysis in conventional access control [23].

Benchmark designers should distinguish influence, attempted operation, executed operation, and realized protected effect. AgentPoison and WASP illustrate the value of decomposing the causal chain [41, 58]. Authorization-sensitive oracles should encode identity, consent, ordering, policy version, time, delegation, and state transition where applicable. Repeated-run reliability also matters: intermittent unauthorized effects have a different interpretation from failures that only reduce benign utility [46].

Deployers should compare published guarantees with the architecture actually operated. Fixed tool registries, trusted metadata, human approvals, simulated services, and benchmark-specific state predicates may not survive dynamic tools, concurrency, retries, partial failures, changing credentials, or external SaaS semantics. Security evaluators should therefore appraise the property rather than the mechanism label: which obligations are required, which are evidenced, which remain assumptions, and whether the paper’s language stays within that supported scope.

MAAP enables structured comparison across mechanisms that use similar security labels but provide materially different enforcement and evidence. For example, two mechanisms described as “least privilege” may differ because one constrains only visible tools whereas another constrains operations and resource instances; similarly, two prompt-injection defenses may differ because one measures model output whereas another observes protected state. These comparisons make the basis of each assurance claim explicit and identify the additional evidence required to support broader scope.

Formal assurance and runtime validation are complementary. Formal work defines what follows inside a model [21, 53, 54]; executable environments show how an implementation behaves under concrete attacks [40, 42, 46, 52, 58]. A mature argument connects formal property, implementation correspondence, runtime enforcement, adversarial evaluation, and protected effect rather than selecting one modality as a universal proxy.

The resulting research agenda has six priorities: (1) measure effective privilege at operation, argument, value, resource, and temporal granularity; (2) test revocation during active tasks, cached credentials, persistent sessions, in-flight operations, and sub-agent propagation; (3) model subject–actor relations, attenuation, redelegation, and credential forwarding; (4) combine protected-state predicates with identity-, consent-, sequence-, and policy-aware process predicates; (5) compose formal, implementation, adversarial, and effect-level evidence; and (6) collect longitudinal deployment evidence under changing tools, credentials, policy, concurrency, retries, and long-lived state. These priorities follow from the coded gaps rather than from an attempt to design a universal agent architecture.

These implications distribute responsibility across the system lifecycle. Mechanism designers define enforceable boundaries; benchmark designers encode the relevant effect and process semantics; deployers verify that research assumptions hold in the operated architecture; and evaluators constrain public claims to the evidence observed. Treating authorization assurance as shared traceability across these roles is more realistic than expecting model alignment, a policy engine, or a single benchmark to carry the complete security argument.

13. Limitations and Threats to Validity

The 30-source corpus is analytically selected and deliberately bounded. The 15-source strict denominator supports corpus-specific findings, not field-wide prevalence. The 15 July 2026 cutoff is material in a rapidly evolving field, and later work may address revocation or delegation directly. Layer B broadens sensitivity but differs from Layer A on at least one strict criterion; Layer C clarifies obligations but cannot describe agent practice. Horizon sources support positioning only.

The structured search was complemented by backward and forward citation tracing to accommodate variation in agent-authorization terminology. Terminology for agent authorization remains unstable, and eligible mechanisms may be described through information flow, capability safety, tool security, or policy compliance without using “authorization.” Citation tracing and intersecting search concepts reduce but

do not eliminate this retrieval risk. The results are therefore anchored to the frozen analytical corpus and should be updated as the field evolves.

SCR construction requires judgment about claim normalization, system boundary, PDP/PEP location, privileged effect, observation stage, and residual assumptions. The double-coding results show strong reproducibility for the tested fields but do not eliminate alternative readings, especially at adjacent observation boundaries. The review therefore reports source-level denominators where prevalence is discussed and does not treat 92 SCRs as independent studies.

The reliability sample covered 20 records and four judgment-bearing fields rather than a second independent coding of the entire corpus. Agreement estimates are consequently evidence about application of the frozen codebook to that sample, not proof that every record has a uniquely determined interpretation. Adjudication may also introduce reviewer judgment even when disagreements are resolved against explicit text and predefined definitions.

Several properties have sparse or zero denominators. Results such as 3/3 least-privilege sources, 0/21 revocation claims, and 0/30 delegation-containment claims are coverage statements within the corpus, not estimates of the complete field. Heterogeneous adversaries and environments further prevent direct equivalence between a failed sandbox attack, a formal protocol result, and a resource-side denial.

Finally, insufficient assurance evidence is not evidence of insecurity. A call-level experiment may leave a final effect unobserved without implying that the effect occurred; a formal result may leave implementation correspondence outside its model without implying an implementation vulnerability. MAAP should therefore be interpreted within the analytical boundary of this review; future validation should examine its completeness, usability, inter-rater behavior, and relationship to deployed security.

14. Conclusion

Tool-using LLM agents transform authorization from an implicit capability concern into a system-level assurance problem. Across 30 full-text sources and 92 SCRs, the review found different enforcement boundaries behind similar labels, limited effective-privilege measurement, no evaluable agent-level revocation or delegation-containment claim in the selected corpus, universal residual assumptions in the strict core, and a persistent distinction between final-state and authorization correctness. The evidence does not support a universal hierarchy; assurance is strongest when property, obligations, enforcement, observation, protected effect, and assumptions correspond.

MAAP operationalizes that correspondence as a novel, review-derived reporting profile for transparent authorization-assurance appraisal. The review's central contribution is therefore the separation of authorization functionality from authorization assurance: the presence of a policy or successful benchmark outcome is not enough unless the evidence justifies the claimed protected effect within an explicit boundary. As agents increasingly exercise delegated authority, security evaluation must ask not

only whether the model behaved safely, but whose authority was exercised, which effect was mediated, where the control operated, what the evidence observed, and what remained assumed.

That distinction permits strong but bounded claims. It does not demand that every paper prove every property; it demands that the scope of the claim remain commensurate with the mechanism and evidence actually supplied.

Declarations

Competing interests.

The author declares no competing interests.

AI Statement.

AI-assisted tools were used to improve language and clarity; the author is fully responsible for the content.

Ethics approval.

Not applicable.

Funding.

No funding was received for conducting this study or preparing this manuscript.

Author Contribution

M A E conceived and designed the review, developed the methodology, curated and analyzed the evidence, prepared the figures and tables, and wrote and revised the manuscript.Since this is a single-author paper, this is a complete and appropriate statement — it covers all the standard CRediT-style roles (conceptualization, methodology, formal analysis, visualization, writing) in one sentence, which is exactly what's expected for a sole-author submission.

Data Availability

The source–claim record dataset and coding materials supporting the findings are available from the corresponding author on reasonable request.

References

1. Leo, M., Tan, F., Miao, T., Anand, G.: From threat to trust: assessing security risks of agentic AI systems. Int. J. Inf. Secur. (2026). https://doi.org/10.1007/s10207-025-01185-y
2. Wen, S.-F., Sharma, A.: Structuring trust: a quantitative and traceable framework for hardware security assurance. Int. J. Inf. Secur. (2026). https://doi.org/10.1007/s10207-026-01284-4
3. Wen, S.-F., Shukla, A., Katt, B.: Artificial intelligence for system security assurance: A systematic literature review. Int. J. Inf. Secur. (2025). https://doi.org/10.1007/s10207-024-00959-0
4. Dzhaliuk, N., Sabodashko, D., Khoma, V., Khoma, Y., Kolchenko, V., Podpora, M.: Comparative evaluation of machine learning methods for protecting LLMs from prompt injection attacks. Int. J. Inf. Secur. (2026). https://doi.org/10.1007/s10207-026-01264-8
5. Bertolissi, C., Anton, A.M., Zannone, N.: Fine-grained access control for multi-user data in cooperative systems: managing compound objects and temporal constraints. Int. J. Inf. Secur. (2026). https://doi.org/10.1007/s10207-026-01227-z
6. Alkhorem, A.H., de Leon, C., Jillepalli, D., Song, A.A.: Graph-based formal modeling and implementation of access control policies with automated conflict and redundancy detection. Int. J. Inf. Secur. (2026). https://doi.org/10.1007/s10207-025-01130-z
7. Rasori, M., Saracino, A., Mori, P., Tiloca, M.: Using the ACE framework to enforce access and usage control with notifications of revoked access rights. Int. J. Inf. Secur. (2024). https://doi.org/10.1007/s10207-024-00877-1
8. Ma, Y.-W., Chiu, P.-H.: A novel risk-based access control engine in zero trust architecture for IoT network. Int. J. Inf. Secur. (2025). https://doi.org/10.1007/s10207-025-01030-2
9. Wang, W., Zhong, N., Xiao, J., Wei, J., Chen, X.: Efficient revocable attribute-based encryption with black-box traceability. Int. J. Inf. Secur. (2026). https://doi.org/10.1007/s10207-026-01247-9
10. Das, B.C., Amini, M.H., Wu, Y.: Security and Privacy Challenges of Large Language Models: A Survey. ACM Computing Surveys (2025). https://doi.org/10.1145/3712001
11. He, F., Zhu, T., Ye, D., Liu, B., Zhou, W., Yu, P.S.: The Emerged Security and Privacy of LLM Agent: A Survey with Case Studies. ACM Computing Surveys (2025). https://doi.org/10.1145/3773080
12. Sheng, Z., Chen, Z., Gu, S., Huang, H., Gu, G., Huang, J.: LLMs in Software Security: A Survey of Vulnerability Detection Techniques and Insights. ACM Computing Surveys (2025). https://doi.org/10.1145/3769082
13. Xu, Y., Zhuang, Y., Liu, X., Zhang, T., Xiao, B., Xu, X., Jiang, D., Wang, J., Hu, H.: LLM agents security duality: a comprehensive survey of self-security and empowered cybersecurity. Artif. Intell. Rev. (2026). https://doi.org/10.1007/s10462-026-11563-0
14. Chowa, S.S., Alvi, R., Rahman, S.S., Rahman, M.A., Raiaan, M.A.K., Islam, M.R., Hussain, M., Azam, S.: From language to action: a review of large language models as autonomous agents and tool users. Artif. Intell. Rev. (2026). https://doi.org/10.1007/s10462-025-11471-9

15. Chen, C., Gong, X., Liu, Z., Jiang, W., Goh, S.Q., Lam, K.-Y.: AI safety landscape for large language models: taxonomy, state-of-the-art, and future directions. Artif. Intell. Rev. (2026). https://doi.org/10.1007/s10462-026-11590-x
16. Khapre, S., Mersha, M.A., Olalekan, Z., Shakil, H., Iskander, S., Moustafa, A., Kalita, J.: Agentic AI systems: A systematic survey of multi-agent architectures, cognitive foundations, interaction, explainability, security, and performance evaluation. Neurocomputing (2026). https://doi.org/10.1016/j.neucom.2026.134049
17. Moia, V.H.G., Sanz, I.J., Rebello, G.A.F., de Meneses, R.D., Hitaj, B., Lindqvist, U.: LLM in the middle: A systematic review of threats and mitigations to real-world LLM-based systems. Comput. Sci. Rev. (2026). https://doi.org/10.1016/j.cosrev.2026.100916
18. Shukla, A., Katt, B., Nweke, L.O., Yeng, P.K., Weldehawaryat, G.K.: System security assurance: A systematic literature review. Comput. Sci. Rev. (2022). https://doi.org/10.1016/j.cosrev.2022.100496
19. Mohamad, M., Steghöfer, J.-P., Scandariato, R.: Security assurance cases—state of the art of an emerging approach. Empir. Softw. Eng. (2021). https://doi.org/10.1007/s10664-021-09971-7
20. Siddiqui, S.A., Gaonkar, R., Köpf, B., Krueger, D., Paverd, A., Salem, A., Tople, S., Wutschitz, L., Xia, M., Zanella-Béguelin, S.: Permissive Information-Flow Analysis for Large Language Models. Trans. Mach. Learn. Res. (2025). https://openreview.net/
21. Li, Z., Xu, J., He, H., Li, H., Wang, Q., Zhou, F.: PPLLA: Privacy-preserving attribute-based LLM authorization. Inf. Sci. (2026). https://doi.org/10.1016/j.ins.2026.123623
22. Madkaikar, M., Yelisetty, A., Sural, S., Vaidya, J., Atluri, V.: Performance analysis of dynamic ABAC systems using a queuing theoretic framework. Computers Secur. (2025). https://doi.org/10.1016/j.cose.2025.104432
23. Wang, X., Guo, Y., Sun, Z., Yu, M., Li, F., Fang, L.: OPMonitor: Continuously monitoring residual over-granted permissions in verified access control policies. Computers Secur. (2025). https://doi.org/10.1016/j.cose.2025.104623
24. Janovsky, A., Jancar, J., Svenda, P., Chmielewski, Ł., Michalik, J., Matyas, V.: sec-certs: Examining the security certification practice for better vulnerability mitigation. Computers Secur. (2024). https://doi.org/10.1016/j.cose.2024.103895
25. Ali, M., Ullah, A., Islam, M.R., Hossain, R.: Assessing of software security reliability: Dimensional security assurance techniques. Computers Secur. (2025). https://doi.org/10.1016/j.cose.2024.104230
26. Shang, S., Wang, X., Liu, A.: ABAC policy mining method based on hierarchical clustering and relationship extraction. Computers Secur. (2024). https://doi.org/10.1016/j.cose.2024.103717
27. Guo, Y., Wang, X., Yu, M., Li, F., Pang, Z., Fang, L.: An on-the-fly framework for usable access control policy mining. Computers Secur. (2025). https://doi.org/10.1016/j.cose.2024.104211
28. Logrippo, L.: Data flow security in Role-based access control. J. Inform. Secur. Appl. (2025). https://doi.org/10.1016/j.jisa.2025.103997

29. Zhu, X., Zou, S., Xu, G., Xi, J.: Group-Capability-Based Access Control with Ring Signature. J. Inform. Secur. Appl. (2025). https://doi.org/10.1016/j.jisa.2025.104014
30. Said, S., Hajlaoui, J.E., Omri, M.N.: New privacy-respecting access control-based approach for data placement in an IoT environment. J. Inform. Secur. Appl. (2025). https://doi.org/10.1016/j.jisa.2025.104192
31. Baseri, Y., Hafid, A., Firoozjaei, M.D., Cherkaoui, S., Ray, I.: Statistical privacy protection for secure data access control in cloud. J. Inform. Secur. Appl. (2024). https://doi.org/10.1016/j.jisa.2024.103823
32. Fu, C., Shen, G., Huang, Z., Xie, J., Fu, J.: Automatic IoT permission assignment with transformer models under spatiotemporal constraints. J. Inform. Secur. Appl. (2025). https://doi.org/10.1016/j.jisa.2025.104099
33. Xu, X., Yang, Z., Cai, Q., Lin, J., Ren, L., Chen, B., Huang, Y.: Enforcing cryptographic distributed-VCS access control with no trust on servers. J. Inform. Secur. Appl. (2025). https://doi.org/10.1016/j.jisa.2025.104103
34. Zhai, Y., Yang, H., Yao, J., Wang, T., Zhou, Y., Zhu, F., Yang, B.: DRAC: A dynamic fine-grained access control scheme for cloud storage with censorship-coerced resistance. J. Inform. Secur. Appl. (2025). https://doi.org/10.1016/j.jisa.2025.104123
35. Alqahtani, H., Ahuja, P.: Secure autonomous cyber defense with LLM agents: A systematic review of autonomy, tool-augmented reasoning, and governance constraints. Comput. Electr. Eng. (2026). https://doi.org/10.1016/j.compeleceng.2026.111184
36. Raza, S., Sapkota, R., Karkee, M., Emmanouilidis, C.: TRiSM for Agentic AI: A review of Trust, Risk, and Security Management in LLM-based Agentic Multi-Agent Systems. AI Open (2026). https://doi.org/10.1016/j.aiopen.2026.02.006
37. Valenta, T., Rozinek, O., Horálek, J.: Agentic AI Safety: A Structured Review of Open Problems and Their Regulatory Anchoring. AI (2026). https://doi.org/10.3390/ai7080298
38. Kincl, J., Adam, M.T.P., Pavleska, T.: Human-AI Integration in Cybersecurity: An Industry-Aligned Perspective on Incident Management. Int. J. Inf. Secur. (2026). https://doi.org/10.1007/s10207-026-01259-5
39. Kröger, N., Zeidler, O., Kellerer, W.: 5G Open Radio Access Network Security - Vulnerabilities on O-RAN Interfaces and Components. Int. J. Inf. Secur. (2026). https://doi.org/10.1007/s10207-026-01255-9
40. Debenedetti, E., Zhang, J., Balunović, M., Beurer-Kellner, L., Fischer, M., Tramèr, F.: AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. NeurIPS 2024 (2024). https://doi.org/10.52202/079017-2636
41. Chen, Z., Xiang, Z., Xiao, C., Song, D., Li, B.: AgentPoison: Red-teaming LLM Agents via Poisoning Memory or Knowledge Bases. NeurIPS. **2024** (2024). https://doi.org/10.52202/079017-4136
42. Ruan, Y., Dong, H., Wang, A., Pitis, S., Zhou, Y., Ba, J., Dubois, Y., Maddison, C.J., Hashimoto, T.: Identifying the Risks of LM Agents with an LM-Emulated Sandbox. ICLR 2024 (2024).

https://arxiv.org/abs/2309.15817

43. Toyer, S., Watkins, O., Mendes, E.A., Svegliato, J., Bailey, L., Wang, T., Ong, I., Elmaaroufi, K., Abbeel, P., Darrell, T., Ritter, A., Russell, S.: Tensor Trust: Interpretable Prompt Injection Attacks from an Online Game. ICLR (2024). (2024) https://arxiv.org/abs/2311.01011
44. Zhang, H., Huang, J., Mei, K., Yao, Y., Wang, Z., Zhan, C., Wang, H., Zhang, Y.: Agent Security Bench (ASB): Formalizing and Benchmarking Attacks and Defenses in LLM-based Agents. ICLR 2025 (2025). https://arxiv.org/abs/2410.02644
45. Andriushchenko, M., Souly, A., Dziemian, M., Duenas, D., Lin, M., Wang, J., Hendrycks, D., Zou, A., Kolter, J.Z., Fredrikson, M., Gal, Y., Davies, X.: AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents. ICLR 2025 (2025). https://arxiv.org/abs/2410.09024
46. Yao, S., Shinn, N., Razavi, P., Narasimhan, K.: τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. ICLR 2025 (2025). https://arxiv.org/abs/2406.12045
47. Kolluri, A., Sharma, R., Costa, M., Köpf, B., Niessen, T., Russinovich, M., Tople, S., Zanella-Béguelin, S.: Optimizing Agent Planning for Security and Autonomy. ICLR 2026 (2026). https://arxiv.org/abs/2602.11416
48. Chen, S., Piet, J., Sitawarin, C., Wagner, D.: StruQ: Defending Against Prompt Injection with Structured Queries. USENIX Security (2025). (2025) https://arxiv.org/abs/2402.06363
49. Bagdasarian, E., Yi, R., Ghalebikesabi, S., Kairouz, P., Gruteser, M., Oh, S., Balle, B., Ramage, D.: AirGapAgent: Protecting Privacy-Conscious Conversational Agents. ACM CCS 2024 (2024). https://doi.org/10.1145/3658644.3690350
50. An, H., Zhang, J., Du, T., Zhou, C., Li, Q., Lin, T., Ji, S.: IPIGuard: A Novel Tool Dependency Graph-Based Defense Against Indirect Prompt Injection in LLM Agents. EMNLP 2025 (2025). https://doi.org/10.18653/v1/2025.emnlp-main.53
51. Zhan, Q., Liang, Z., Ying, Z., Kang, D.: InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents. Findings of ACL 2024 (2024). https://doi.org/10.18653/v1/2024.findings-acl.624
52. Lu, J., Holleis, T., Zhang, Y., Aumayer, B., Nan, F., Bai, H., Ma, S., Ma, S., Li, M., Yin, G., Wang, Z., Pang, R.: ToolSandbox: A Stateful, Conversational, Interactive Evaluation Benchmark for LLM Tool Use Capabilities. Findings of NAACL 2025 (2025). https://doi.org/10.18653/v1/2025.findings-naacl.65
53. Debenedetti, E., Shumailov, I., Fan, T., Hayes, J., Carlini, N., Fabian, D., Kern, C., Shi, C., Terzis, A., Tramèr, F.: Defeating Prompt Injections by Design. IEEE SaTML (2026). (2026) https://arxiv.org/abs/2503.18813
54. Doshi, A., Hong, Y., Xu, C., Kang, E., Kapravelos, A., Kästner, C.: Towards Verifiably Safe Tool Use for LLM Agents. ICSE NIER 2026 (2026). https://doi.org/10.1145/3786582.3786839
55. Lawal, S., Zhao, X., Rios, A., Krishnan, R., Ferraiolo, D.: Translating Natural Language Specifications into Access Control Policies by Leveraging Large Language Models. IEEE TPS-ISA 2024 (published 2025 record) (2025). https://doi.org/10.1109/TPS-ISA62245.2024.00048

56. Ling, Y., Yu, S., Chen, Z., Fang, C.: Toward Secure LLM Agents: Threat Surfaces, Attacks, Defenses, and Evaluation. arXiv:2606.10749 (2026). https://arxiv.org/abs/2606.10749
57. Mellafe Zuvic, D.: Capability Gates Are Not Authorization: Confused-Deputy Failures in LLM Agent Frameworks. arXiv:2606.28679 (2026). https://arxiv.org/abs/2606.28679
58. Evtimov, I., Zharmagambetov, A., Grattafiori, A., Guo, C., Chaudhuri, K.: WASP: Benchmarking Web Agent Security Against Prompt Injection Attacks. NeurIPS **2025** (2025). https://mlanthology.org/neurips/2025/evtimov2025neurips-wasp/
59. Abdelnabi, S., Fay, A., Salem, A., Zverev, E., Liao, K.-C., Liu, C.-H., Kuo, C.-C., Weigend, J., Manlangit, D., Apostolov, A., Umair, H., Donato, J., Kawakita, M., Mahboob, A., Bach, T.H., Chiang, T.-H., Cho, M., Choi, H., Kim, B., Lee, H., Pannell, B., McCauley, C., Russinovich, M., Paverd, A., Cherubin, G.: LLMail-Inject: A Dataset from a Realistic Adaptive Prompt Injection Challenge. arXiv:2506.09956 (2025). https://arxiv.org/abs/2506.09956
60. Abdelnabi, S., Gomaa, A., Bagdasarian, E., Kristensson, P.O., Shokri, R.: Firewalls to Secure Dynamic LLM Agentic Networks. arXiv:2502.01822 (2025). https://arxiv.org/abs/2502.01822
61. Beurer-Kellner, L., Buesser, B., Creţu, A.-M., Debenedetti, E., Dobos, D., Fabian, D., Fischer, M., Froelicher, D., Grosse, K., Naeff, D., Ozoani, E., Paverd, A., Tramèr, F., Volhejn, V.: Design Patterns for Securing LLM Agents against Prompt Injections. arXiv:2506.08837 (2025). https://arxiv.org/abs/2506.08837
62. Costa, M., Köpf, B., Kolluri, A., Paverd, A., Russinovich, M., Salem, A., Tople, S., Wutschitz, L., Zanella-Béguelin, S.: Securing AI Agents with Information-Flow Control. arXiv:2505.23643 (2025). https://arxiv.org/abs/2505.23643
63. Wang, S., Zhang, G., Yu, M., Wan, G., Meng, F., Guo, C., Wang, K., Wang, Y.: G-Safeguard: A Topology-Guided Security Lens and Treatment on LLM-based Multi-agent Systems. Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, 7261–7276 (2025). https://doi.org/10.18653/v1/2025.acl-long.359
64. Wallace, E., Xiao, K., Leike, R., Weng, L., Heidecke, J., Beutel, A.: The Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions. arXiv:2404.13208 (2024). https://arxiv.org/abs/2404.13208
65. Liu, Y., Deng, G., Li, Y., Wang, K., Wang, Z., Wang, X., Zhang, T., Liu, Y., Wang, H., Zheng, Y., Zhang, L.Y., Liu, Y.: Prompt Injection Attack against LLM-Integrated Applications. arXiv:2306.05499 (2023). https://arxiv.org/abs/2306.05499
66. Saltzer, J.H., Schroeder, M.D.: The Protection of Information in Computer Systems. Proceedings of the IEEE (1975). https://doi.org/10.1109/PROC.1975.9939
67. Hardy, N.: The Confused Deputy (or why capabilities might have been invented). ACM SIGOPS Operating Syst. Rev. (1988). https://doi.org/10.1145/54289.871709
68. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-Based Access. Control Models IEEE Comput. (1996). https://doi.org/10.1109/2.485845

69. NIST: NIST SP 800 – 162: Guide to Attribute Based Access Control (ABAC) Definition and Considerations. NIST Special Publication: (2014). https://doi.org/10.6028/NIST.SP.800-162
70. NIST: NIST SP 800 – 207: Zero Trust Architecture. NIST Special Publication: (2020). https://doi.org/10.6028/NIST.SP.800-207
71. IETF: RFC 6749: The OAuth 2.0 Authorization Framework. IETF RFC: (2012). https://www.rfc-editor.org/rfc/rfc6749
72. IETF: RFC 8693: OAuth 2.0 Token Exchange. IETF RFC: (2020). https://www.rfc-editor.org/rfc/rfc8693
73. IETF: RFC 8705: OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens. IETF RFC: (2020). https://www.rfc-editor.org/rfc/rfc8705
74. Model Context Protocol project: Model Context Protocol (MCP) Authorization Specification. MCP Specification: (2025). https://modelcontextprotocol.io/specification/
75. OWASP: OWASP Top 10 for LLM Applications / Agentic security guidance. OWASP guidance: (2025). https://genai.owasp.org/

Figures

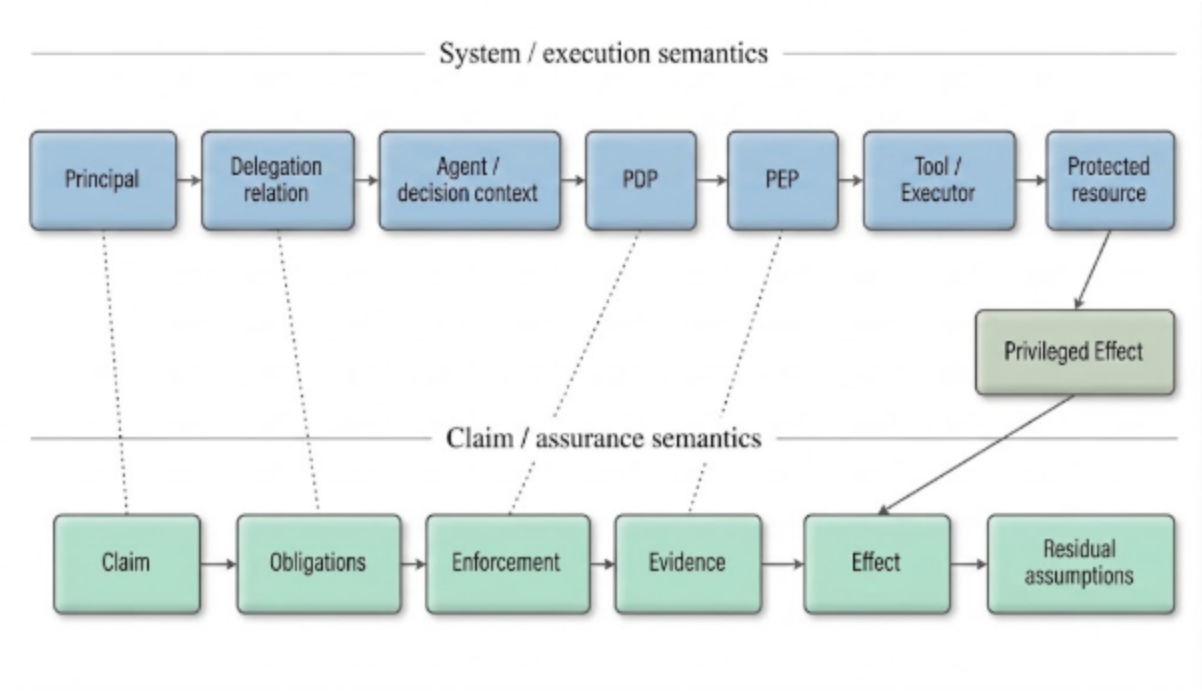


Figure 1

Authorization Assurance Claim Model. *The upper path shows how delegated authority reaches a protected resource; the lower path shows the assurance argument used to appraise the corresponding claim.*

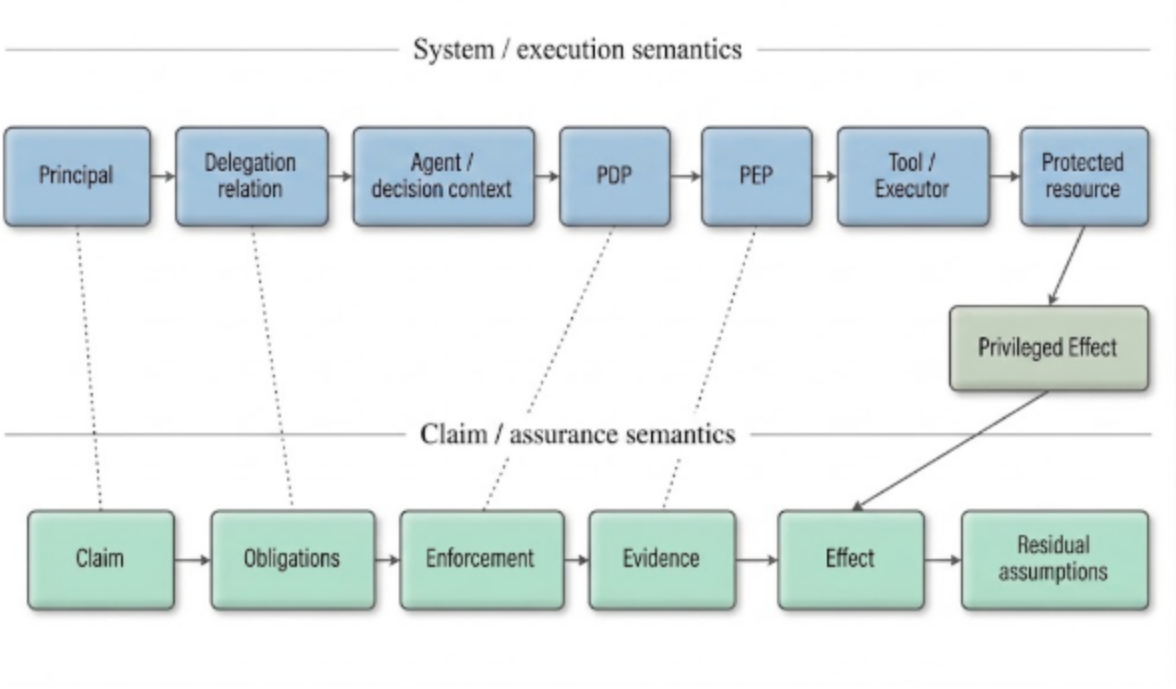


Figure 1

Authorization Assurance Claim Model. *The upper path shows how delegated authority reaches a protected resource; the lower path shows the assurance argument used to appraise the corresponding claim.*

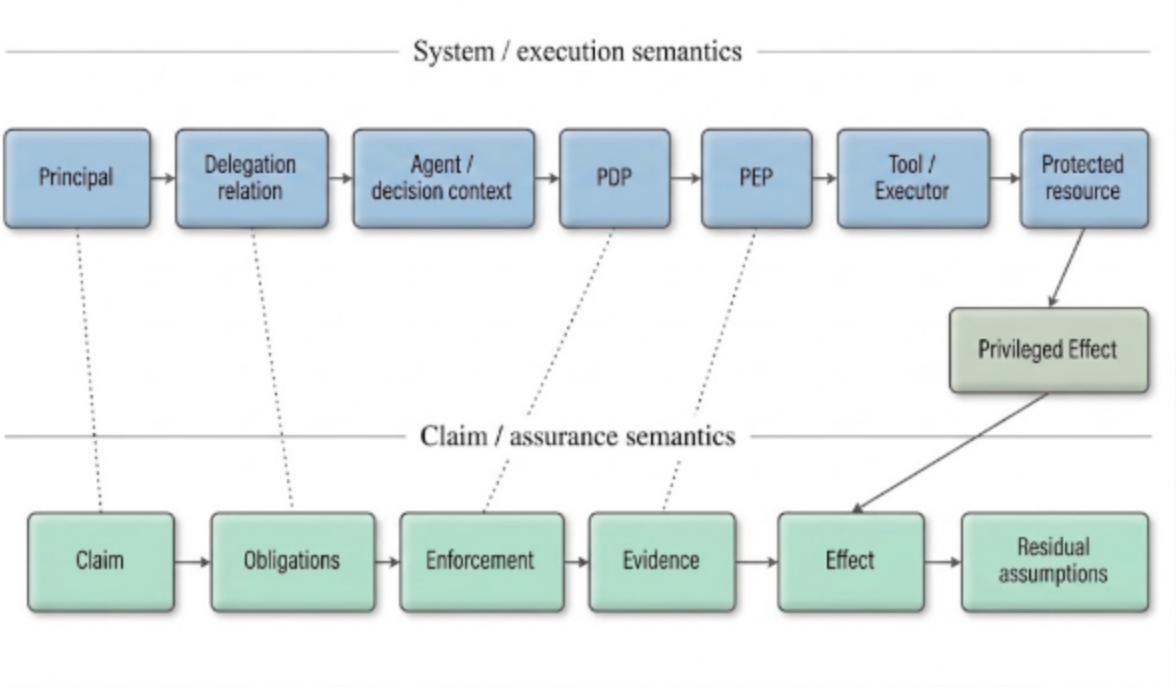


Figure 1

Authorization Assurance Claim Model. *The upper path shows how delegated authority reaches a protected resource; the lower path shows the assurance argument used to appraise the corresponding claim.*

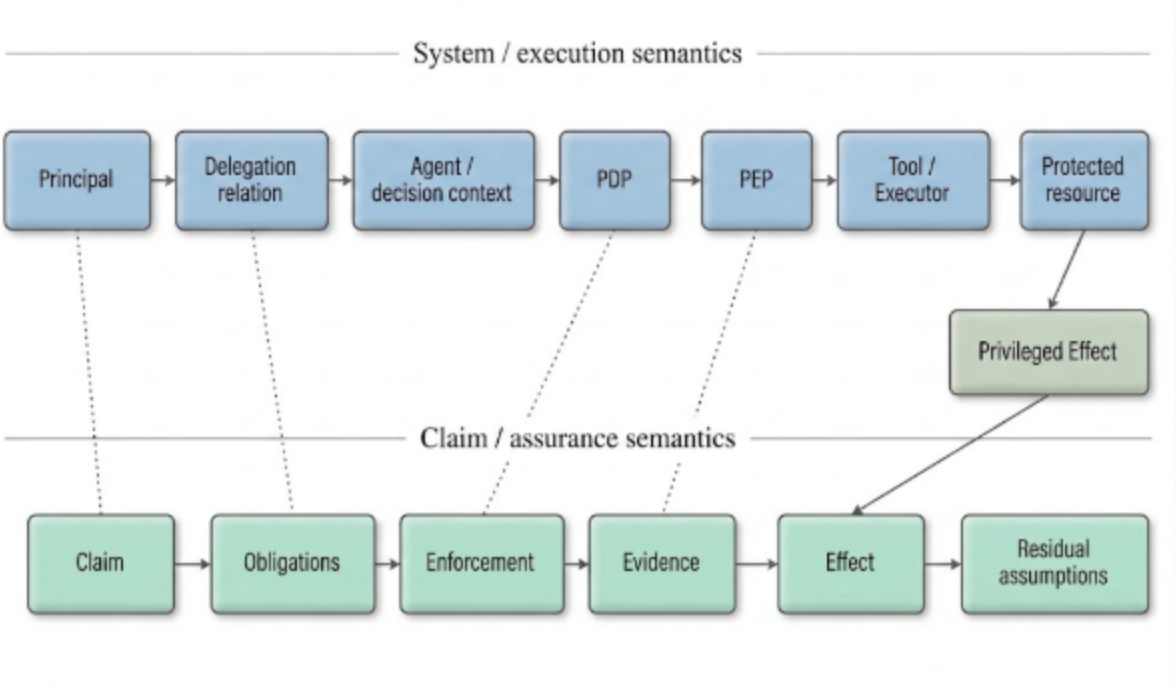


Figure 1

Authorization Assurance Claim Model. *The upper path shows how delegated authority reaches a protected resource; the lower path shows the assurance argument used to appraise the corresponding claim.*

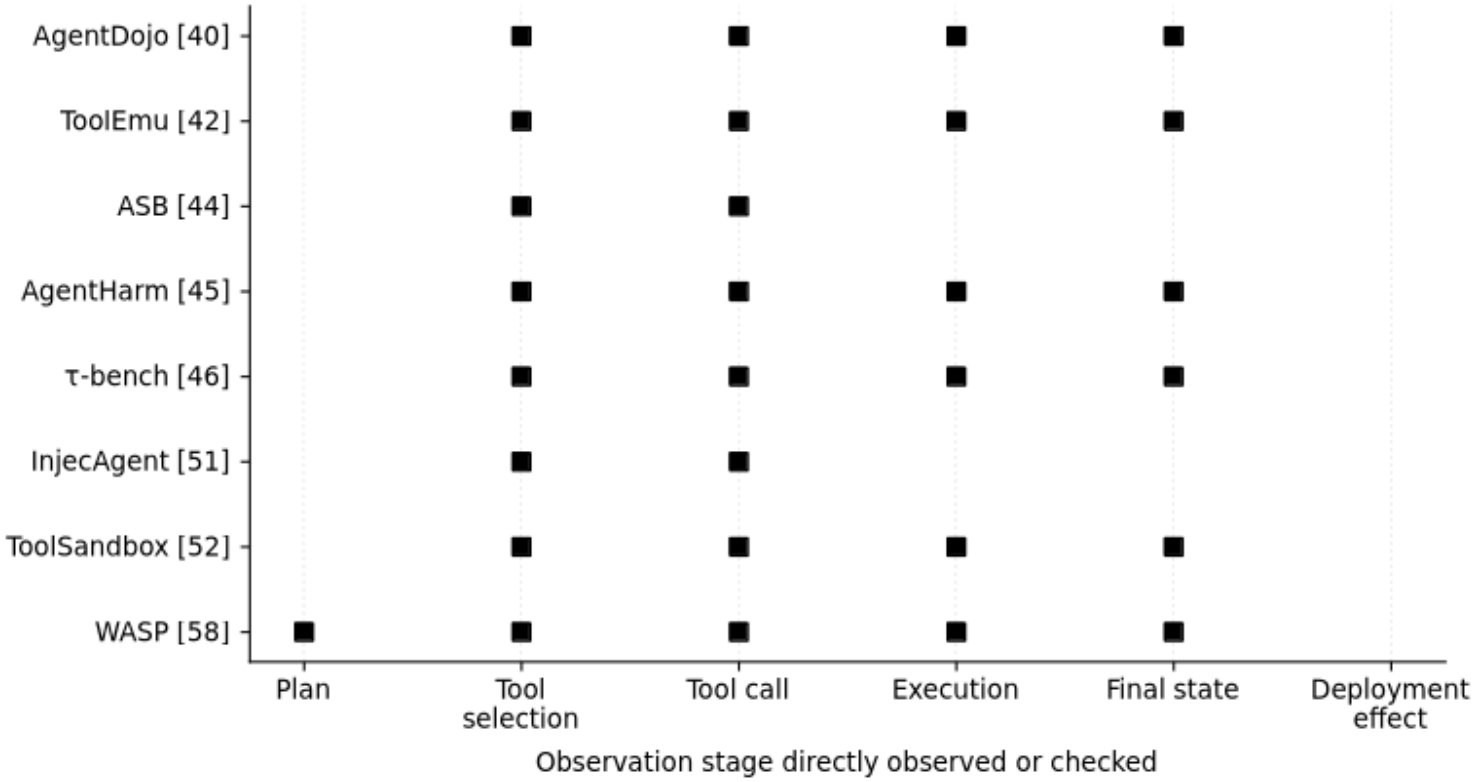


Figure 2

Observation coverage in representative agent-security benchmarks.*Dots identify stages directly observed or checked; the figure is descriptive, not a quality ranking.*

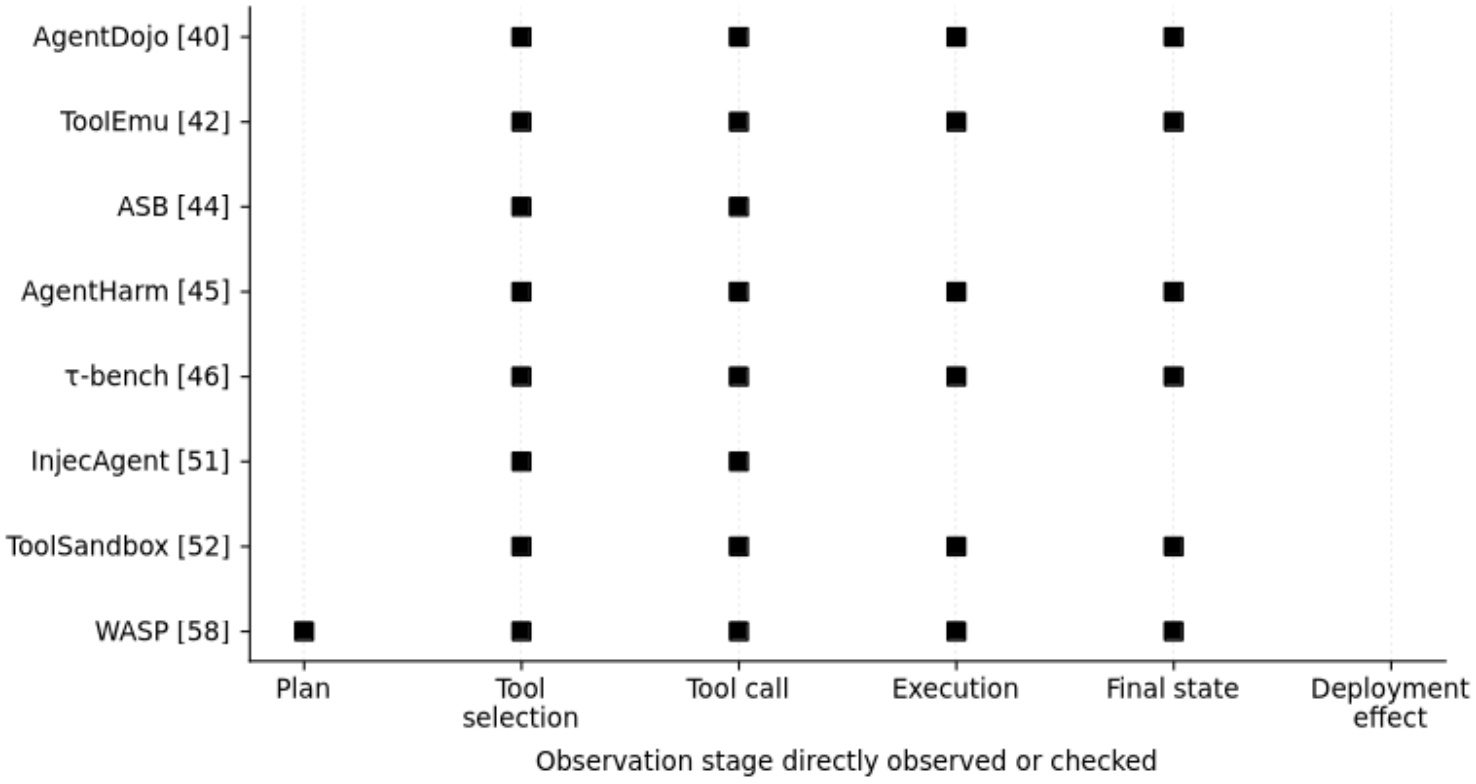


Figure 2

Observation coverage in representative agent-security benchmarks.*Dots identify stages directly observed or checked; the figure is descriptive, not a quality ranking.*

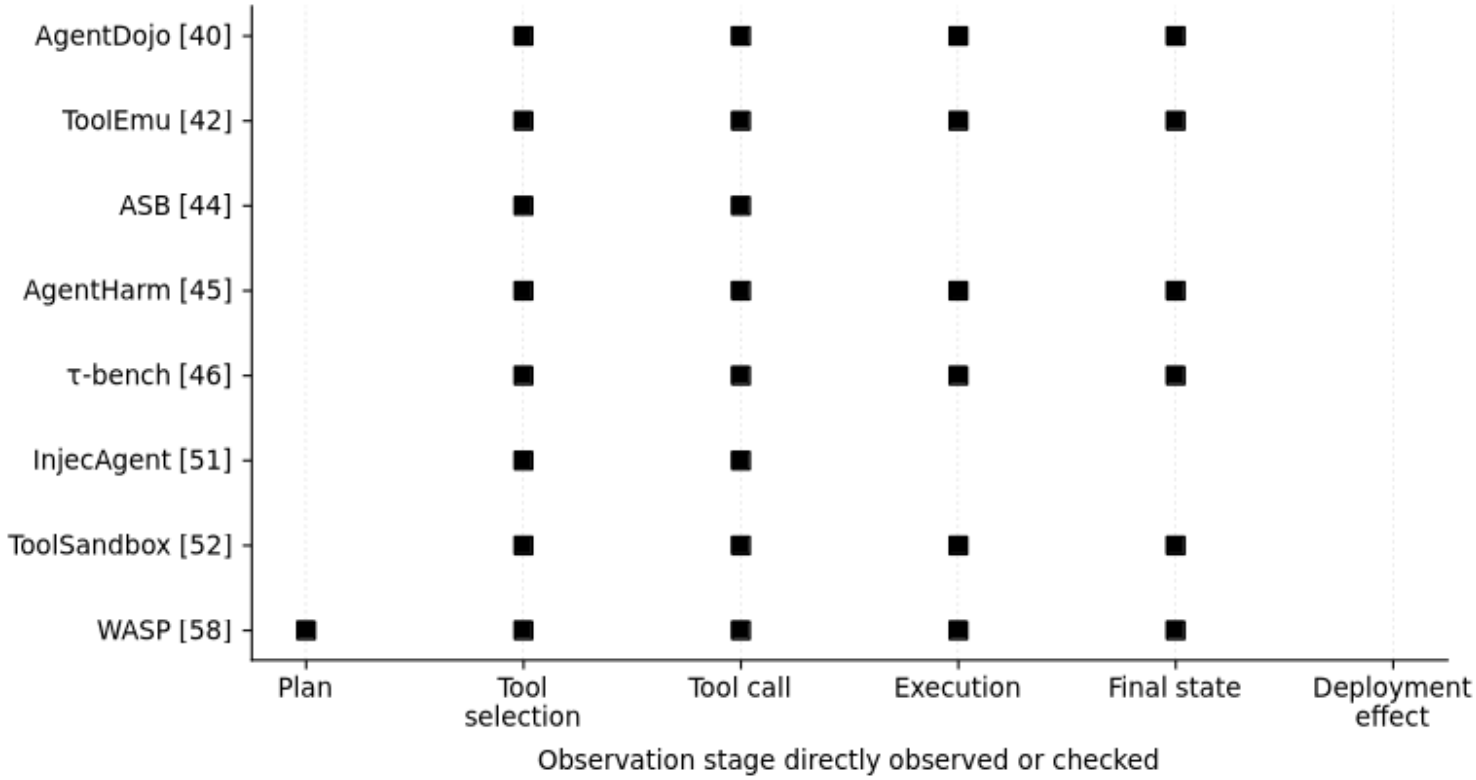


Figure 2

Observation coverage in representative agent-security benchmarks.*Dots identify stages directly observed or checked; the figure is descriptive, not a quality ranking.*

AgentDojo [40]
ToolEmu [42]
ASB [44]
AgentHarm [45]
τ-bench [46]
InjecAgent [51]
ToolSandbox [52]
WASP [58]
Plan
Tool selection
Tool call
Execution
Final state
Deployment effect
Observation stage directly observed or checked

Figure 2

Observation coverage in representative agent-security benchmarks.*Dots identify stages directly observed or checked; the figure is descriptive, not a quality ranking.*

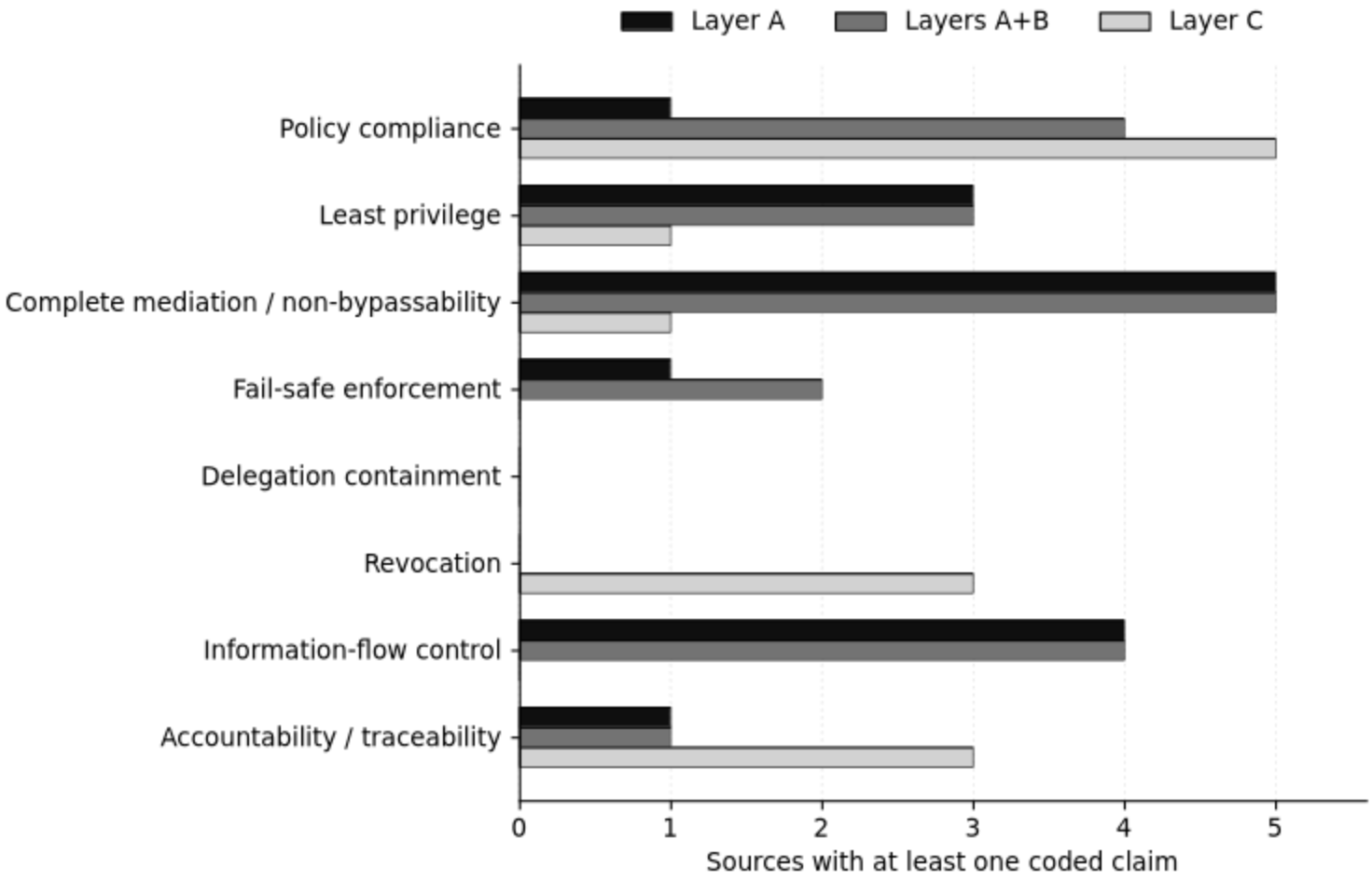


Figure 3

Property coverage across the closed analytical corpus. *Counts represent sources with at least one coded claim and should not be interpreted as field-wide prevalence.*

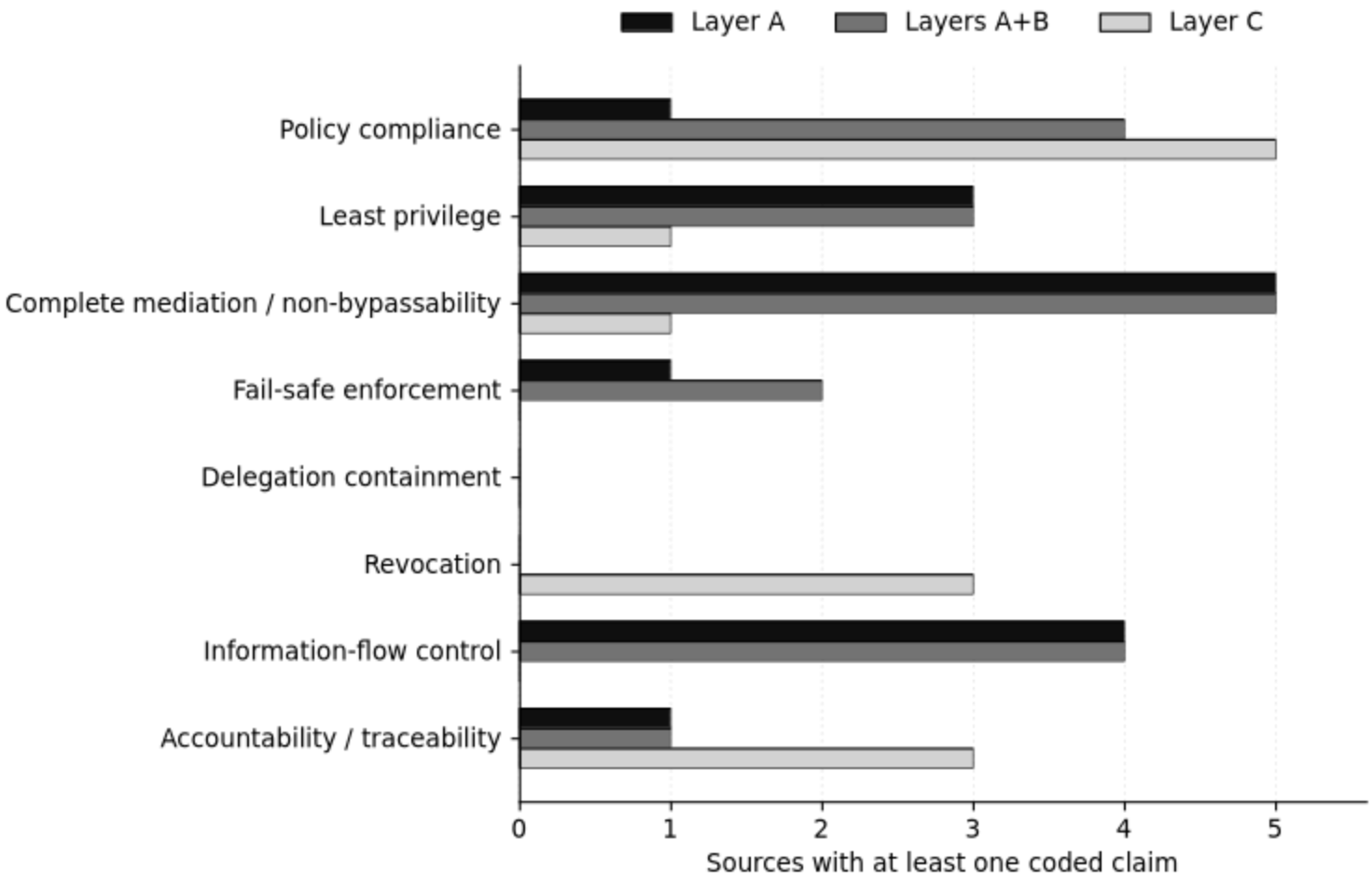


Figure 3

Property coverage across the closed analytical corpus. *Counts represent sources with at least one coded claim and should not be interpreted as field-wide prevalence.*

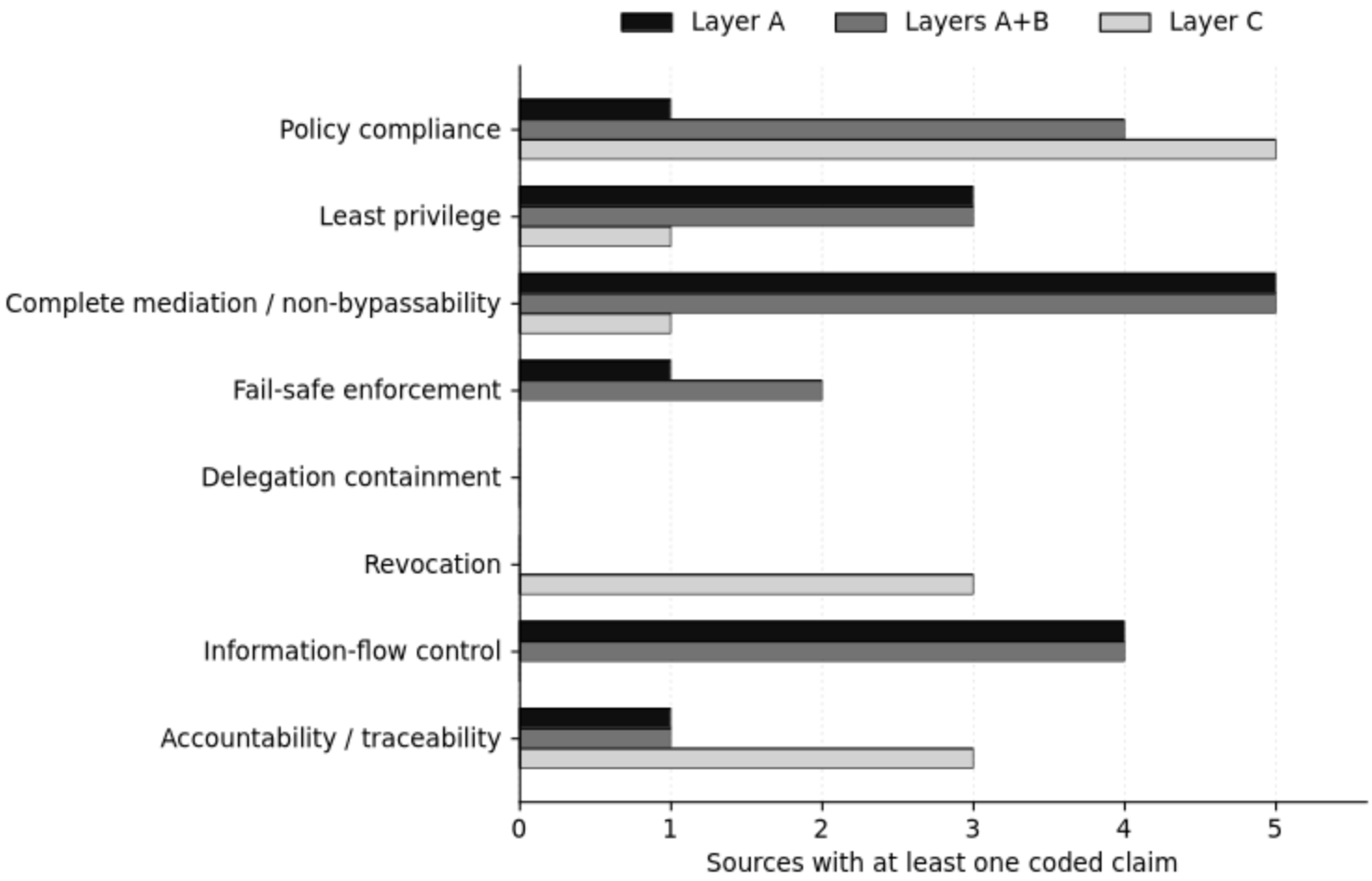


Figure 3

Property coverage across the closed analytical corpus. *Counts represent sources with at least one coded claim and should not be interpreted as field-wide prevalence.*

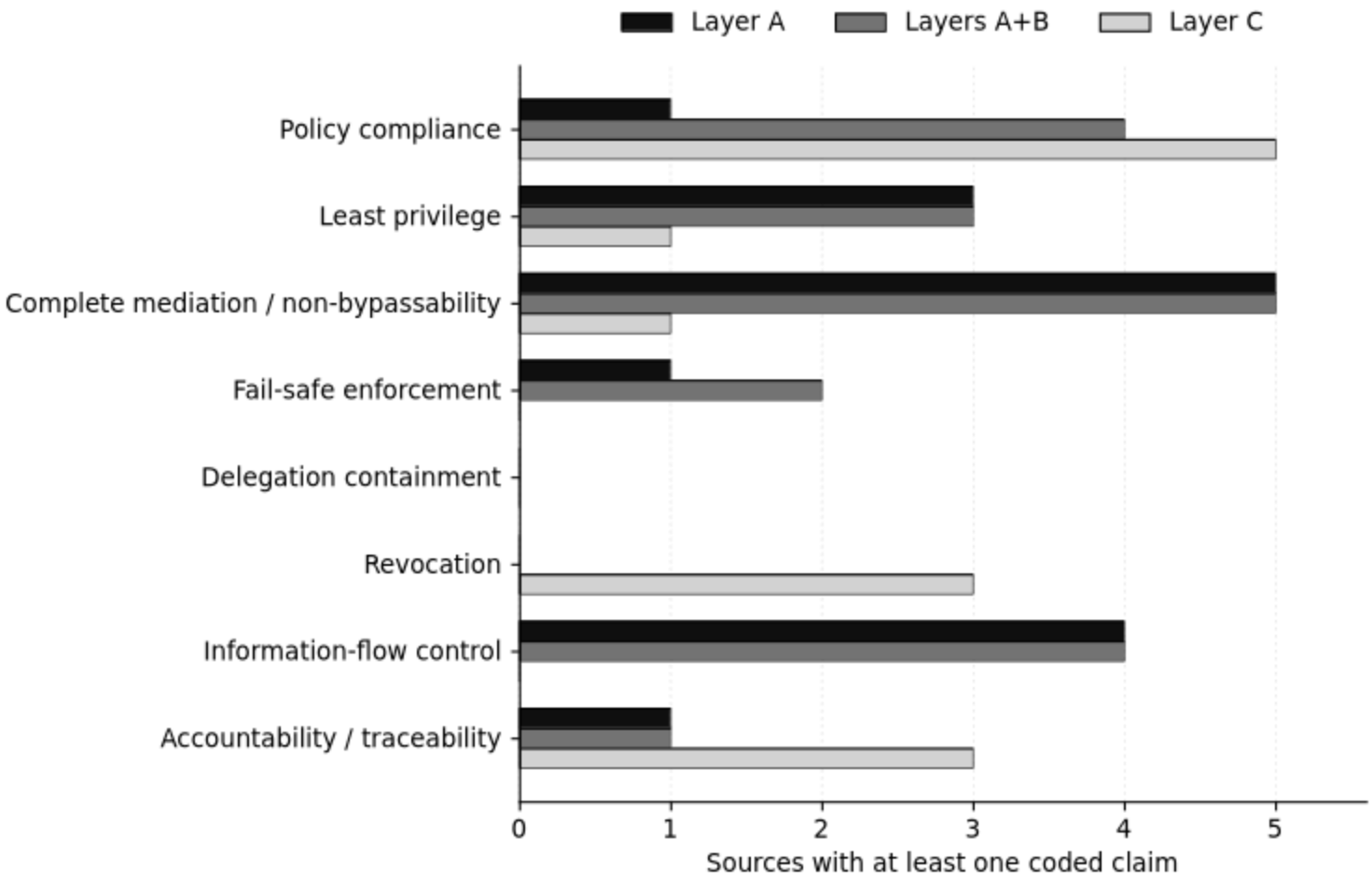


Figure 3

Property coverage across the closed analytical corpus. *Counts represent sources with at least one coded claim and should not be interpreted as field-wide prevalence.*