BEYOND PROMPT FILTERING: WHAT RUNTIME SAFETY MECHANISMS FOR LLM AGENTS OBSERVE, DECIDE, AND CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Runtime defenses for tool-using LLM agents operate at different observation, decision and execution interfaces, making their safety claims difficult to compare. We present a source-anchored comparative synthesis of 77 version-pinned comparison records spanning runtime mechanisms, evaluations and related syntheses. A Horizon–Modality (H/M) map characterizes the evidence available to named checking stages and how they decide across 25 core mechanisms and supporting components. Mechanisms grouped under information-flow defenses can rely on learned behavioral judgments or enforced runtime restrictions, requiring different validation evidence. The synthesis distinguishes task alignment, authorization and effect correctness, and identifies validation obligations for policy updates and cross-authority handoffs. Two external human coders independently annotated 68 core/extension records, agreeing on all 25 core H/M assignments; agreement was lower for decision targets and inferred premises. We map evaluations by whether they measure judgments, executed effects or legitimate completion, clarifying what evidence supports a learned guard's deployment claim. A comparison-record schema, validator and worked examples support reuse, alongside a handoff checklist, reporting template and an executable handoff test with 80 fixtures and expected outcomes.

1 INTRODUCTION

An agent can give a helpful answer through an unauthorized tool, or send the wrong document to an authorized recipient. Indirect injection exploits the instruction/data boundary (Greshake et al., 2023); conversation classifiers such as Llama Guard (Inan et al., 2023) address a different target from action authorization.

Consider a workflow: A reads an internal memo, B summarizes it, and C sends it to a partner. Useful content, credentials and release permission are distinct facts; identical sends can have opposite authorization states (Siu et al., 2026, Sections 4.1, 4.4). Enforcement must preserve source restrictions and bind release authority to current state.

We organize executable policies, provenance-aware runtimes, learned monitors and isolation around evidence *horizon* and decision *modality*, with intervention, trust and evaluation records.

Three questions guide the synthesis. **RQ1:** What do mechanisms observe, decide, and control? **RQ2:** Which assumptions must survive agent and authority boundaries? **RQ3:** Which outcomes do evaluations measure? We contribute: **(1) A traceable comparison resource:** 77 version-pinned records, a 25-core HM matrix and independent human coding of 68 core/extension records; **(2) A cross-system synthesis:** three distinctions linking decision targets, inferred premises and authority scope to the evidence needed for composition; **(3) Reusable reporting instruments:** a comparison-record schema, validator and three non-core worked examples, plus a separate handoff checklist and execution-reporting template. Sections 4–6 answer these questions in order; one bounded handoff illustration demonstrates the record's use (Section 6.2).

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

2 SCOPE AND EVIDENCE PROTOCOL

Selection and scope. We study software-agent actions that change state or transfer information across trust boundaries. The purposive comparison contains 25 core records, 43 bounded-extension records and nine named sources; RvLLM and ProbGuard are contextual boundary cases. Appendix H records queries, eligibility, versions, supporting sources and retrospective table allocation; Appendix H.5 checks four additional scope challenges.

Comparison procedure. We extract property, observations, intervention, trust and evaluation, then map horizon, modality and authority scope. Source statements and transfer inferences remain separate; results retain their original tasks, denominators and approval semantics. Section 7 consolidates the coverage and review scope.

Position among related syntheses. Kim et al. (2026, Table 2) maps defense categories to covered risks, grouping MELON, CaMeL and Fides under IFC/taint tracking. Our **checker-level H/M/D3 mappings** distinguish learned call-similarity decisions from runtime label propagation: the former needs detector validation, the latter label and propagation validation; neither supplies a missing release grant (Section 4). Siu et al. (2026, Sections 7–8) connects contextual properties and ideal oracles while leaving multi-agent delegation and guarantee composition open. We make the corresponding deployment questions inspectable through **version-pinned records** with source/inference separation and independent coding statistics (Appendix B.1), plus **executable fixtures** with policy-specific expected outcomes, not a general composition guarantee. Specification and trajectory surveys (Dantas et al., 2026; Lotfi et al., 2026), architectural/security syntheses (Beurer-Kellner et al., 2025; Ling et al., 2026), and MCP, authorization, harness and guardrail reviews (Hou et al., 2025; Surapani et al., 2026; Li et al., 2026b; Wang et al., 2025e) provide complementary scopes; Appendix H.7 records overlaps.

Use for machine learning research. The distinction matters when learning a guard, policy or world model. Task Shield, MELON and ToolSafe make learned judgments; ProbGuard checks an estimated model, while permissive IFA combines utility estimation with restricted-context generation. Table 2 and Figure 3 help match a learned guard's claim to its evaluation: judgment accuracy, an intervention's executed effects, or legitimate completion. This separates improvement of the learned component from evidence for protection by the deployed agent.

3 WHAT DOES AN ENFORCEMENT CLAIM MEAN?

3.1 SEPARATE OBSERVATION, DECISION, AND EXECUTION

Let s_t be the environment state before step t, h_t the executed history, and a_t a proposed action. Monitor i receives $o_{i,t} = O_i(s_t, h_t, a_t)$ and returns

$$d_{i,t} = m_i(o_{i,t}) \in \{\mathsf{allow}, \mathsf{block}, \mathsf{clarify}\}. \tag{1}$$

A gate uses this decision to determine whether a_t reaches the environment. Clarification must suspend the protected action until a new decision is made; otherwise it is advice, not a blocking outcome. Complete mediation requires that every relevant execution path, including delegated calls and retries, pass through the gate. This is an application of a classical protection requirement (Saltzer & Schroeder, 1975), not a property conferred by placing a checker in a diagram.

For an action gate, let Φ be a prefix-closed set of permitted finite histories: once a prohibited event occurs, later actions cannot erase that violation. A conditional safety claim has the form

$$\mathcal{A} \implies \forall \tau \in \mathrm{Exec}(\text{agent}, \text{gate}, \text{environment}), \quad \tau \in \Phi, \tag{2}$$

where $\mathcal{A}$ requires faithful encoding, adequate observations, sound decisions and complete execution control (Schneider, 2000). This excludes arbitrary liveness; noninterference compares multiple executions (Sabelfeld & Myers, 2003). Information-flow control (IFC) may enforce conservative flow restrictions without proving every semantic confidentiality requirement.

3.2 Comparison Records and HM Coding

We describe a mechanism by $\langle P, O, I, T, E \rangle$: protected property, observation scope, intervention, trusted assumptions, and supporting evidence. Formal rules do not establish faithful policy translation or complete mediation. Checking a learned transition model likewise establishes a statement about that model under its assumptions, not an unconditional deployment guarantee.

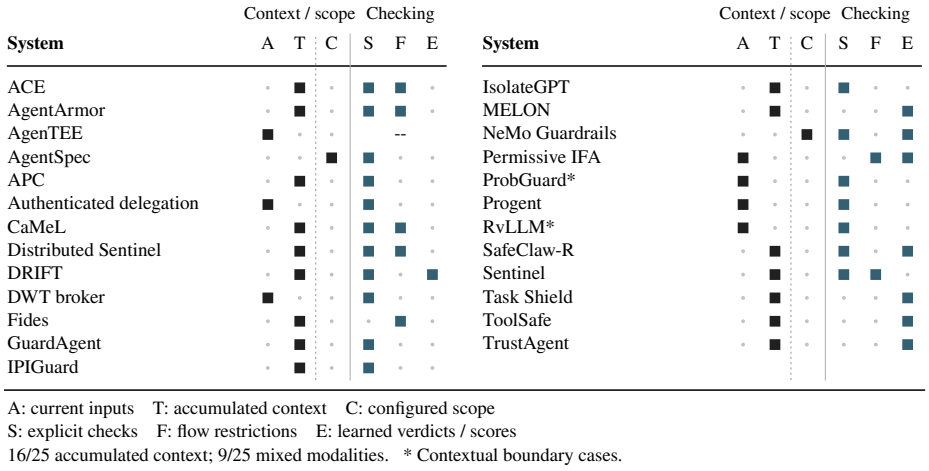

Figure 1: H/M profiles of all 25 core records. Squares mark active codes; multiple M marks identify combined checking stages. C denotes configured scope, not a third temporal horizon. AgenTEE's dash denotes substrate protection without a semantic action-policy check. Table 5 adds targets, premises and authority scope.

HM codes the *named checking stages*, not an entire system. The notation below separates record fields, stage codes and evidence claims.

Notation	Meaning
$\langle P, O, I, T, E \rangle$	Property, observations, intervention, trust premises, evaluation evidence.
H: A / T / C	Present input / accumulated context (including labels) / configured scope.
M: S / F / E	Explicit rule/model check / flow restriction / learned verdict or score.
D1 / D2 / D3	Target / inferred premise / authority scope.
D3: W / D / X	Workflow-local / delegated / cross-agent or provider authority.
RQ1 / RQ2 / RQ3	Mechanism interface / transfer obligations / measured outcomes.
Eqs. (1)–(3); Obs. 1	Verdict; conditional safety; non-composition; indistinguishable observations.

Record T/E denote trust/evidence, not the H=T or M=E codes. Thresholding a learned score remains E. Combined codes denote distinct stages; a dash in Table 5 denotes no semantic check.

Generated policies, programs, extracted facts and estimated models do not make their downstream symbolic checker E: Progent and CaMeL follow this rule. Core H/M/D3 codes agree between coders; D1/D2 inform individual comparisons rather than quantitative taxonomy claims. Decision units and rationales appear in Appendix J.

Independent coding. Two external human coders independently annotated 68 core/extension records. An author adjudicated all 35 disputed records, including 13 core D1/D2 cases. Appendix B.1 reports applicability, all dimensional statistics and final labels separately.

3.3 Three Diagnostic Failure Modes

Specification non-composition: local policies hold without implying the global property. *Monitor error*: acceptance fails to establish the checked property. *Execution-control failure*: execution skips a check or ignores denial. Reading permission need not grant release; an unobservable grant instead

creates an evidence deficit (Appendix G). Distinguishing these causes requires observations and effects, not just verdict logs.

4 Comparative Synthesis

Figure 1 maps 25 core records; Tables 5, 6–7 give their interfaces and evidence. NeMo flows (Rebedea et al., 2023) and TrustAgent inspection (Hua et al., 2024) already combine prompting with runtime control, not prompt-only filtering.

Landscape finding (RQ1). Sixteen core records use accumulated context, seven current inputs and two configured scope. Ten designated checking stages use S alone; nine records combine modalities. An explicit check can use current or accumulated evidence and depend on inferred premises (Table 5). These counts describe the purposive core, not end-to-end learning independence.

4.1 Rules and Structured Verification

AgentSpec (Wang et al., 2026a) joins triggers, predicates and enforcement. Section 3.3 defines predicates over user input/trajectory, supplied with current state and pending actions for action events. Interventions include stopping, user inspection and LLM reconsideration.

GuardAgent (Xiang et al., 2025) generates executable checks from guard requests, specifications and logs using LLM planning and retrieved demonstrations. Translation fidelity and effect interception remain separate obligations.

Progent v3 (Shi et al., 2026) gates tool names/arguments with symbolic policies. LLM-proposed updates require approval or SMT-verified non-expansion. Initial-policy and approver correctness remain premises; approval need not be human.

RvLLM (Zhang et al., 2025b) forward-chains expert rules over LLM-extracted propositions, checking output rather than tool execution. The authors require faithful encoding (Section 3.4) and acknowledge incomplete inequality-domain knowledge (Sections 1 and 4.3).

AgentArmor (Wang et al., 2025b) checks LLM-inferred dependencies, propagates labels using a trusted registry, and blocks violating calls. Its "transfer execution" study (Section VI-E) exposes a utility/protection trade-off when allowing/disallowing instructions from a user-designated file.

IPIGuard (An et al., 2025) plans dependencies before external responses, resolves arguments, permits query-only expansion and redirects injected instructions through fake tool invocations. Query/Command classification is unspecified. Our inference: argument fidelity and classification remain premises; read-only status does not establish confidentiality or recipient authorization.

DRIFT v3 (Li et al., 2025) combines LLM-generated call plans, parameter checklists, dynamic validation and tool-output masking before memory insertion. Read deviations are allowed; Write/Execute deviations require intent alignment or user approval. Evaluation counts approval requests as rejection, not observed user decisions.

4.2 Information Flow and Authority Separation

CaMeL (Debenedetti et al., 2025) separates trusted control from untrusted data and checks security policies over capability metadata before tool execution. Section 5.4 proposes user confirmation for deployment-time violations instead of hard blocking. Fides (Costa et al., 2025) tracks confidentiality and integrity, selectively hiding and revealing values to preserve planner expressiveness. Runtime propagation and checks, not model-reproduced tags, implement its policy; a coarse taint wrapper does not capture the full mechanism.

Earlier f-secure systems (Wu et al., 2024) protect planning integrity through labeled context and structured execution. RTBAS (Zhong et al., 2025) couples learned dependency screening with actual pre-generation redaction and tool-policy checks. AirGapAgent (Bagdasarian et al., 2024) minimizes data before third-party conversation and isolates the minimizer.

Conservative IFC can block acceptable outputs (Denning, 1976; Sabelfeld & Myers, 2003). Permissive IFA (Siddiqui et al., 2025) searches labeled subcontexts, then *regenerates* the returned output

from the selected subcontext, rather than relabeling it. Monotonic utility conditions search completeness; the flow argument requires restricted-context generation and correct input labels. This is not a complete tool executor; release still needs authorization.

At a handoff, the question is whether the receiving runtime preserves restrictions and their authority, not whether an LLM can delete a textual tag. Authorized declassification differs from accidental weakening; semantic similarity alone proves neither authorization nor complete removal of sensitive content.

4.3 Task-Centered and Learned Checks

Task Shield (Jia et al., 2025) uses LLM instruction extraction and contribution judgments for task alignment. Its tool-call path blocks misaligned calls; tool-output processing adds feedback. This runtime intervention checks contribution, not necessarily resource-owner permission.

MELON (Zhu et al., 2025) terminates before execution when original and masked-run calls are similar. Its ensemble analysis assumes detector-score separation. The analysis uses bounded similarity scores; our concern is the unstated sampling independence needed for its Hoeffding bound. Reported experiments test single-prompt variants rather than the ensemble (Sections 3.4, 4.3). Detection remains a behavioral proxy, not an authorization grant.

ToolSafe (Mou et al., 2026) supplies TS-Guard with the request, history, candidate call and available tool specifications. TS-Flow blocks unsafe-rated calls and returns feedback for continuation. Whether "controversial" ratings trigger online blocking is unspecified; strict/loose benchmark scoring is separate.

LlamaFirewall (Chennabasappa et al., 2025) layers input scanning, trace-based alignment checks, and static code analysis. Its AgentDojo evaluation incrementally rescans fixed traces and credits flagged attacks as defended (Section 4.3); that measures a different interface from live post-intervention recovery.

4.4 Isolation and Delegated Authority

IsolateGPT (Wu et al., 2025) isolates app processes and contexts, with non-LLM operators and a trusted hub mediating exchanges. Within-app compromise is excluded. Its protection evaluation reports permission-dialog appearance and warning rates, not users' eventual decisions. We distinguish containment from consent quality and allowed-message semantics.

ACE (Li et al., 2026a) combines trusted planning, app matching, static flow checks and isolated workers. Its metadata attack exceeds IsolateGPT's trusted-schema assumption; execution-integrity/availability attacks use app outputs within that threat model (Table I; Section III). Injection benchmarks score app calls; Tool Usage also scores final outputs/states (Section V-C). Comparisons retain single-query scope, trusted-label and executor premises.

AgenTEE (Abdollahi et al., 2026) protects component assets with attested confidential VMs and channels, trusting Arm CCA hardware/firmware and user input delivery. Its OpenCCA evaluation measures agent–model latency rather than action-policy compliance; software attestation does not certify semantic decisions.

South et al. (2025) propose identity-linked scope tokens, mutual authentication, and token validation by receiving services, leaving translation validation open. We distinguish authenticated grants from faithful interpretation and enforcement.

Bounded Agents' Agentic Principal Chain (APC) (Muruaga, 2026) attenuates delegated scope and budgets and checks session action combinations outside the model. Its composition theorem assumes complete, static effective restrictions and serialized admission; tool-to-action classification remains a premise. It directly addresses delegated composition, while parameter-level misuse and cross-session coordination remain outside that guarantee.

Delegation Without Trust (DWT) (Dantuluri & Sundi, 2026) describes a broker issuing attenuated, workload-bound grants and mediating calls, including revocation. Sender binding requires authenticated workload identity. Its unreleased demonstrator tests abstract tool/resource actions; live-model

integration remains future work. Our inference is that confinement to a valid grant does not certify every permitted action's semantic effects.

4.5 Prediction and Cross-Agent Enforcement

ProbGuard (Wang et al., 2026b) learns a predicate-abstracted Markov chain for stopping or re-prompting. Its driving/embodied evaluations are a boundary case here. As the authors note (Sections 5.2, 7), the budget of 30 traces per driving scenario is below certification bounds and PAC guarantees exclude abstraction misspecification. The driving instantiation uses a reachability bound rather than the stronger uniform CTL guarantee.

SafeClaw-R (Wang et al., 2026c) mediates functional nodes, explicitly assuming faithful skill execution (Section 3.3, Figure 5). Its approximately 2,000-case Workspace study includes benign controls. For third-party skills, the authors report excessive REVIEW and absent benign controls; the code-execution study leaves advanced semantic evasion untested. Skill detection includes review; code cases cosmetically expand 20 seeds. We do not equate these verdicts with independent exploit families or complete mediation.

Sentinel and Distributed Sentinel (Wu & Gong, 2026b;a) use graph checks and taint; the latter adds Semantic Taint Tokens (STTs) and remote predicate queries. Fixed-call/constructed-workflow evidence supports conditional claims requiring faithful state/grounding, local witnesses and complete propagation (Appendix C).

4.6 What the Mechanism Comparison Reveals

Target choice changes the required evidence. Task Shield checks task contribution; Progent checks privileges; CaMeL/Fides propagate restrictions. A useful send may be unauthorized; an authorized, flow-compliant send may contain the wrong summary. Match the target to task evidence, a resource-specific grant, provenance or an effect oracle. Our selection rule is to combine complementary obligations: task-alignment judgments do not supply a missing resource-owner grant.

Similar gates have different policy-update obligations. Progent's unapproved updates face an SMT non-expansion check, whereas DRIFT's write/execute deviations involve learned intent judgment or approval. Fixed privileges require non-expansion; scope-expanding exceptions need an explicit expansion authority. HM locates the check; inferred premises identify what to revalidate when a policy generator, approver or state source changes. Conseca provides a compact worked example below.

Adding agents need not change authority. Progent evaluates agents serving one user under unified control (Shi et al., 2026); APC and DWT add delegated confinement. APC already supplies conditional composition; the comparison question is which premises a concrete handoff preserves. A shared policy service may supply these facts locally; an independently administered recipient needs agreed grant scope and policy interpretation, artifact-bound provenance, and current authority (Table 1). Our deployment inference is that crossing authorities can change the evidence contract even with an unchanged local checker. These source-anchored contrasts do not rely on coder-sensitive D1/D2 label frequencies.

Worked record: Conseca. The non-core Conseca example (Tsai & Bagdasarian, 2025, Sections 3.2–4.1) makes the policy-generation/enforcement boundary inspectable:

Field	Filled comparison record
P	Task-specific API and argument admissibility.
O	Trusted context for policy generation; proposed API and arguments for checking.
I	Admit/reject calls using API and regex-argument restrictions.
T	Faithful generated policy, trusted-context boundary and complete mediation.
E	Linux prototype; 20 selected tasks; five-trial averages (source Figure 3).
H/M/D3	A/S/W: current-call deterministic check within one workflow.
D1/D2	Authorization target; policy-generation fidelity is an inferred premise.

Table 1: Proposed handoff-contract checklist, paired with the execution-level reporting record in Appendix E. Each interface supplies a concrete verification question for composed workflows.

Interface	Representative mechanisms	Required check for composition
Policy to check	AgentSpec; GuardAgent; Progent; DRIFT	Do checks capture the policy, and who may approve its expansion?
Objective to action	Task Shield; MELON; ToolSafe; IPIGuard	Does the plan or safety judgment also satisfy the resource owner's authorization rule?
State to observation	AgentArmor; ProbGuard; RTBAS	Are relevant facts, dependencies, and abstractions faithful and current?
Source to derived data	CaMeL; Fides; permissive IFA	Are provenance and restrictions preserved or explicitly declassified by a trusted authority?
Agent to agent	IsolateGPT; APC; DWT broker	Do recipient identity, delegated authority, and policy meaning survive the handoff?
Decision to effect	Every blocking architecture	Can retries, alternate tools, concurrent updates, or direct calls bypass the checked decision?

Separating T from I requires validating policy fidelity separately from execution's obedience to it. This record passes the companion validator's checks of schema consistency, not semantic correctness. These review annotations are outside the 68-record independent coding exercise.

5 From Local Mechanisms to Multi-Agent Workflows

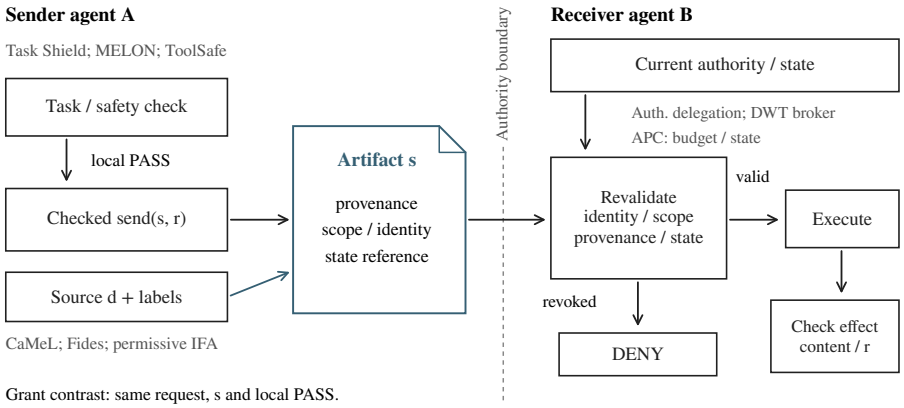

Figure 2: Proposed handoff structure, synthesized from the annotated mechanisms. Artifact-bound evidence crosses the authority boundary; the receiver combines it with current authority and state before execution. Under the grant-dependent policy, valid and revoked grants lead to different branches despite identical local evidence. Actual effects are checked separately.

5.1 When Can Local Claims Compose?

Drawing on compositional reasoning (McMillan, 1999), we propose a deployment test: local contracts must entail the global invariant, establish successor assumptions and survive state-consistent execution. The memo's send needs authenticated provenance, propagated restrictions and a current release grant; missing evidence requires suspension.

AIOracles (Villa et al., 2026) gives a positive generator/filter composition result under component completeness and security premises (v1, Theorem 1). Applying such a result requires matching

those premises to the deployed interfaces; Appendix H.3 records this and four other author-selected follow-up sources.

5.2 What Must Survive a Handoff?

Synthesis answer (RQ2): transfer policy meaning, authority and state assumptions, not just a verdict (Figure 2; Table 1). Authority scope depends on who controls policy-relevant facts, not model count: several agents can share one authority, while one agent can cross authorities.

Policy meaning and delegated authority. Macaroons restrict credential context (Birgisson et al., 2014); agent tokens and brokers bind grants to identities and calls (South et al., 2025; Dantuluri & Sundi, 2026). SAGA already enforces contact policies between agents (Syros et al., 2026). Transfer requires matching the grant's operation, recipient, resource and policy interpretation.

Provenance through derivation. Runtime IFC propagates restrictions (Costa et al., 2025; Debenedetti et al., 2025); restricted-context regeneration (Siddiqui et al., 2025) returns a new artifact. Our transfer obligation is to bind provenance to the released artifact and enforce its release policy, not an observed mechanism defect.

Shared state and commit-time validity. APC's serialized admission and cumulative accounting (Muruaga, 2026) address within-session composition. Our deployment inference is to preserve these premises through retries, revocation, and shared-state updates. A trajectory log records events; enforcing a budget also requires state-consistent admission.

6 What Do Existing Evaluations Establish?

6.1 Different Benchmarks Answer Different Questions

R-Judge (Yuan et al., 2024), TraceSafe (Chen et al., 2026c) and TS-Bench score judgments on supplied records; TS-Flow also exercises blocking and feedback (Mou et al., 2026). AgentDojo (Debenedetti et al., 2024) checks model output and environment state deterministically. AgentHarm (Andriushchenko et al., 2025) uses effect-free synthetic tools and benign capability controls. Table 2 separates these measurement interfaces.

EU-Agent-Bench (Lichkovski et al., 2025) evaluates legally grounded single-turn choices, excluding trials without the necessary tool call; its conditional legality rate is not all-request safe completion. The benchmark map distinguishes judgments, executed effects, and legitimate completion.

MAST (Cemri et al., 2025) labels 14 failure modes, including in successful trajectories, not authorization. ConVerse (Gomaa et al., 2026) evaluates multi-turn privacy/security using simulated interactions and judges. Table 9 adds tool-risk and computer-use settings without pooling scores.

ST-WebAgentBench (Levy et al., 2024) and The Verifier Tax (Sah et al., 2026) already measure joint safe completion; the latter includes recovery. τ-bench (Yao et al., 2024) tests final states and repeated-trial reliability; AI Control (Greenblatt et al., 2024) tests intentional model subversion. Adaptive Progent evaluation (Narisetty et al., 2026) covers one defense/attack family with an open-weight model.

The gap is *evidence transfer*: different policies and execution interfaces preclude pooled safety rankings. ShieldAgent conditions compliance on task success (Chen et al., 2025); ClawGuard's security setting denies all approval requests (Zhao et al., 2026); Prudentia's benign traces assume approval (Kolluri et al., 2026). These are distinct from observed legitimate safe completion. Tables 7 and 20 preserve the denominators and conditions.

Coverage finding (RQ3). Our source-linked extraction of 11 selected evaluation settings records five directly measuring task completion, two joint safe completion and one conditional recovery; four inspect executed state. These are setting-specific counts, not field-wide prevalence. Figure 3 preserves each measurement's scope and denominator.

Table 2: Evaluation map: measured outcomes and additional evidence needed when extending the claim. Core and named-source anchors are in Appendices A and I.

Evaluation	What the setting measures	Evidence needed for transfer
R-Judge; TraceSafe; TS-Bench	Risk judgment over supplied records, traces, or candidate steps	Live execution and post-block recovery
AgentDojo	Injection resistance and task utility in interactive environments	Organization-specific policies and deployment sampling
AgentHarm	Harmful task completion with synthetic tools; benign capability controls	Validated real effects and cross-agent policy checks
EU-Agent-Bench	Rubric-based legality among necessary-tool-calling trials	All-request and multi-step outcome accounting
ST-WebAgentBench	Completion with zero measured policy violations	Policy/evaluator coverage beyond the tested suite
τ-bench	Final database outcomes; repeated-trial reliability	Intermediate-action checks for trajectory constraints
MAST	Failure modes in multi-agent execution traces	A separate oracle for policy violations

6.2 An Illustrative Handoff Contract

We provide a reusable handoff test oracle, exercised through a configured CaMeL interpreter and Fides tutorial receiver in an isolated AgentDojo mailbox. Supplied reader labels govern subject and body; Appendix F gives 432 local policy-translation checks.

Sixteen fixtures cross readers, receiving identity and revocation; four requests remain authorized. Five bridges supply 80 fixtures total: (i) current evidence, (ii) missing-evidence rejection, (iii) missing-as-public, (iv) stale-unchecked, and (v) revalidation. Arms (iii) and (iv) are our permissive negative controls, not upstream defaults or defects. Each fixture has a policy-derived expected outcome, computed separately from runtime decisions.

All 80 executions match those expectations. Current or revalidated evidence permits four safe deliveries per arm; the negative controls produce 12 and four unauthorized effects. Missing-evidence rejection denies all four authorized calls. These are conformance counts, not autonomous-agent success rates; no recovery is attempted. The companion's expected-outcome file and checker support regression testing of a replacement bridge (Figure 4, Appendix F.3).

7 Research Agenda and Limitations

Three priorities follow: **handoff contracts** binding identity, provenance and current authority; **evidence acquisition** measuring the cost of trusted facts and clarification; and **change and recovery** testing revocation, retries and shared state alongside legitimate completion. Task-level evaluation should distinguish incompatible contracts, checker errors and unmediated execution.

Limits of this synthesis. Purposive selection, bounded retrieval and unavailable versions limit coverage. Supplementary retrieval is a discovery reserve, not a fully eligibility-assessed corpus (Appendix H.2). Source verification and independent coding retain their stated scopes. The handoff test checks supplied-label conformance; autonomous attacks and recovery require task-level evaluation. Timing/covert channels and arbitrary liveness are outside scope; synthetic permissions are not legal conclusions.

8 Conclusion

Our checker-level map and source-linked records connect runtime mechanisms to composition and evaluation claims. The worked records and executable handoff test support reuse. Learned-guard evaluation should report current authority, executed effects and legitimate completion alongside judgment accuracy.

AI USE STATEMENT

Generative AI assisted literature inspection, comparison-framework development, the elementary observation, manuscript and bibliography revision, and figure and document tooling. It also assisted experimental code and tests, case construction, record auditing, agreement analysis and interpretation of the finite policy-translation and handoff tests. An author checked all main-text source assertions and section/page anchors against pinned sources and verified artifact counts. Separately, A and B manually and independently assigned applicability and H/M/D labels to the 68 core/extension records without consulting each other or existing classifications. Those label assignments were human; AI assisted the statistical tooling and reporting. An author then adjudicated the 35 disputed records; AI validated the returned worksheet's structure and codebook consistency and integrated the decisions, preserving the original annotations and agreement statistics. An author also completed six follow-up selection decisions after consulting AI-prepared source notes. An author reviewed the AI-assisted screening decisions at the title and abstract level, including follow-up checks of source identity and relevance to the paper's scope. AI integrated the returned decisions. Broader exploratory model calls are not reported as performance results in this paper. Human authors are responsible for the final claims, citations and artifacts. Section 7 identifies the review scope.

ETHICS STATEMENT

The finite conformance example uses authored fixtures and public benchmark data with fictional identities; effects remain in memory and no condition calls a model or external tool service. The supplementary comparison data exclude private exploratory records and service identifiers. Upstream code is imported from pinned checkouts, not redistributed here. This work certifies neither legal compliance nor deployment safety.

REPRODUCIBILITY STATEMENT

Appendices A and B pin sources and coding rules; Appendix H gives exact search queries, eligibility rules, reconciled selection counts and extended source anchors. The ledger separates source claims from review inferences. Observation 1 includes its elementary proof. Appendix E gives a suggested evidence record, and Appendix F details the policy-translation and handoff tests. The reviewer supplementary material, `survey_companion.zip`, contains 77 version-pinned comparison records, six follow-up selection decisions, scoped retrieval accounting, named HM decision units, the selection register and historical decisions, four scope challenges, a reporting template, 432 local checks, 80 handoff fixtures, direct-database retrieval accounting, checksums and offline verifiers. Its README gives offline verification and native reproduction commands. The archive also includes both human annotation sets, their frozen codebook, pre-adjudication agreement statistics, author decisions, the final classification mapping and reproduction code (Appendix B.1). The verifier checks record integrity, accounting and numerical consistency. The archive excludes private exploratory logs and copied upstream code.

REFERENCES

Sina Abdollahi, Mohammad M Maheri, Javad Forough, Amir Al Sadi, Josh Millar, David Kotz, Marios Kogias, and Hamed Haddadi. AgenTEE: Confidential LLM agent execution on edge devices. In *Proceedings of the Sixth European Workshop on Machine Learning and Systems*, 2026. doi: 10.1145/3805621.3807660. URL https://arxiv.org/abs/2604.18231v2.

Robel Addis, Kiran Porter, Jayden Ryan, Micah Stull, Shengjie Xu, Paul Wagner, Robert Honomichl, and Steven Forsyth. LLM-Based Intelligent Agents for Cybersecurity: A Tutorial and Survey of Automated Vulnerability Discovery. *IEEE Access*, 14:100884–100917, 2026. doi: 10.1109/ACCESS.2026.3707573. URL https://doi.org/10.1109/ACCESS.2026.3707573.

Basavesh Ammanaghatta Shivakumar, Swarn Priya, and Peng Gao. AgentFlow: A flow-centric policy language and framework for securing LLM agent systems, 2026. URL https://arxiv.org/abs/2608.22868v1.

Hengyu An, Jinghuai Zhang, Tianyu Du, Chunyi Zhou, Qingming Li, Tao Lin, and Shouling Ji. IPIGuard: A novel tool dependency graph-based defense against indirect prompt injection in LLM agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1023–1039, 2025. doi: 10.18653/v1/2025.emnlp-main.53. URL https://aclanthology.org/2025.emnlp-main.53/.

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, Eric Winsor, Jerome Wynne, Yarin Gal, and Xander Davies. AgentHarm: A benchmark for measuring harmfulness of LLM agents. In *ICLR*, 2025. URL https://arxiv.org/abs/2410.09024v3. Cited version: arXiv:2410.09024v3.

Eugene Bagdasarian, Ren Yi, Sahra Ghalebikesabi, Peter Kairouz, Marco Gruteser, Sewoong Oh, Borja Balle, and Daniel Ramage. AirGapAgent: Protecting privacy-conscious conversational agents. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 3868–3882. ACM, 2024. doi: 10.1145/3658644.3690350. URL https://arxiv.org/abs/2405.05175v2.

Roy Betser, Amit Giloni, Shamik Bose, Sindhu Padakandla, Chiara Picardi, Lidor Erez, and Roman Vainshtein. AgenTRIM: Tool risk mitigation for agentic AI, 2026. URL https://arxiv.org/abs/2601.12449v2.

Luca Beurer-Kellner, Beat Buesser, Ana-Maria Creţu, Edoardo Debenedetti, Daniel Dobos, Daniel Fabian, Marc Fischer, David Froelicher, Kathrin Grosse, Daniel Naeff, Ezinwanne Ozoani, Andrew Paverd, Florian Tramèr, and Václav Volhejn. Design patterns for securing LLM agents against prompt injections, 2025. URL https://arxiv.org/abs/2506.08837v3.

Rishika Bhagwatkar, Kevin Kasa, Abhay Puri, Gabriel Huang, Irina Rish, Graham W. Taylor, Krishnamurthy Dj Dvijotham, and Alexandre Lacoste. Indirect Prompt Injections: Are Firewalls All You Need, or Stronger Benchmarks? arXiv:2510.05244v1, 2025. URL https://arxiv.org/abs/2510.05244v1.

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*, 2014. doi: 10.14722/ndss.2014.23212. URL https://www.ndss-symposium.org/wp-content/uploads/2017/09/04_3_1.pdf.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2025. URL https://arxiv.org/abs/2503.13657v3.

Ada Chen, Yongjiang Wu, Junyuan Zhang, Jingyu Xiao, Shu Yang, Jen-tse Huang, Kun Wang, Wenxuan Wang, and Shuai Wang. JARVIS or ultron? a survey on the safety and security threats of computer-using agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45407–45441, 2026a. URL https://aclanthology.org/2026.acl-long.2106/.

Tianyu Chen, Chujia Hu, Ge Gao, Dongrui Liu, Xia Hu, and Wenjie Wang. LPS-Bench: Benchmarking Safety Awareness of Computer-Use Agents in Long-Horizon Planning under Benign and Adversarial Scenarios. arXiv:2602.03255v1, 2026b. URL https://arxiv.org/pdf/2602.03255v1.

Yen-Shan Chen, Sian-Yao Huang, Cheng-Lin Yang, and Yun-Nung Chen. Tracesafe: A systematic assessment of LLM guardrails on multi-step tool-calling trajectories. In *Conference on Language Modeling*, 2026c. URL https://arxiv.org/abs/2604.07223v2. To appear.

Zhaorun Chen, Mintong Kang, and Bo Li. ShieldAgent: Shielding agents via verifiable safety policy reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 2025. URL https://proceedings.mlr.press/v267/chen25ae.html.

Zhi-Yuan Chen, Shiqi Shen, Guangyao Shen, Gong Zhi, Xu Chen, and Yankai Lin. Towards tool use alignment of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1382–1400. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.82. URL https://aclanthology.org/2024.emnlp-main.82/.

Sahana Chennabasappa, Cyrus Nikolaidis, Daniel Song, David Molnar, Stephanie Ding, Shengye Wan, Spencer Whitman, Lauren Deason, Nicholas Doucette, Abraham Montilla, Alekhya Gampa, Beto de Paola, Dominik Gabi, James Crnkovich, Jean-Christophe Testud, Kat He, Rashnil Chaturvedi, Wu Zhou, and Joshua Saxe. LlamaFirewall: An open source guardrail system for building secure AI agents. *arXiv preprint arXiv:2505.03574*, 2025. URL https://arxiv.org/abs/2505.03574v1.

Michael R. Clarkson and Fred B. Schneider. Hyperproperties. *Journal of Computer Security*, 18(6): 1157–1210, 2010. doi: 10.3233/JCS-2009-0393. URL https://www.cs.cornell.edu/fbs/publications/Hyperproperties.pdf.

Jingyue Cong, Xinyuan Qiao, Yulin Dong, Yueheng Huang, Yang Yu, Estrid He, and Andy Song. IntentGuard: Safeguard LLM Agents via Intent Alignment. In *AAAI 2026 Workshop on Trustworthy Agentic AI*, 2026. URL https://trustagenticai.github.io/AAAI2026/AAAI-Workshop/47.pdf.

Manuel Costa, Boris Köpf, Aashish Kolluri, Andrew Paverd, Mark Russinovich, Ahmed Salem, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Securing AI agents with information-flow control. *arXiv preprint arXiv:2505.23643*, 2025. URL https://arxiv.org/abs/2505.23643v2.

Pierre Dantas, Lucas Cordeiro, Ehsan Nowroozi, and Norbert Tihanyi. Toward safe LLM agents: A survey of specification, verification, and enforcement. *arXiv preprint arXiv:2608.14590*, 2026. URL https://arxiv.org/abs/2608.14590v1.

Panduranga Sai Varma Dantuluri and Jyotirmoy Sundi. Delegation without trust: An empirical gap analysis of identity, authorization, and runtime governance in multi-agent LLM systems. *arXiv preprint arXiv:2609.00267*, 2026. URL https://arxiv.org/abs/2609.00267v1.

Debeshee Das, Luca Beurer-Kellner, Marc Fischer, and Maximilian Baader. CommandSans: Securing AI Agents with Surgical Precision Prompt Sanitization. arXiv:2510.08829v1, 2025. URL https://arxiv.org/pdf/2510.08829v1.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems*, 2024.

Edoardo Debenedetti, Ilia Shumailov, Tianqi Fan, Jamie Hayes, Nicholas Carlini, Daniel Fabian, Christoph Kern, Chongyang Shi, Andreas Terzis, and Florian Tramèr. Defeating prompt injections by design, 2025. URL https://arxiv.org/abs/2503.18813v2.

Zehang Deng, Yongjian Guo, Changzhou Han, Wanlun Ma, Junwu Xiong, Sheng Wen, and Yang Xiang. AI Agents Under Threat: A Survey of Key Security Challenges and Future Pathways. arXiv:2406.02630v2, 2024. URL https://arxiv.org/pdf/2406.02630v2.

Dorothy E. Denning. A lattice model of secure information flow. *Communications of the ACM*, 19 (5):236–243, 1976.

Ishaan Domkundwar, Mukunda N S, Ishaan Bhola, and Riddhik Kochhar. Safeguarding AI agents: Developing and analyzing safety architectures. *arXiv preprint arXiv:2409.03793*, 2024. URL https://arxiv.org/abs/2409.03793v3.

Shen Dong, Shaochen Xu, Pengfei He, Yige Li, Jiliang Tang, Tianming Liu, Hui Liu, and Zhen Xiang. Memory injection attacks on LLM agents via query-only interaction. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.52202/085713-1554. URL https://arxiv.org/abs/2503.03704v3.

Wenjie Fu, Xiaoting Qin, Jue Zhang, Qingwei Lin, Lukas Wutschitz, Robert Sim, Saravan Rajmohan, and Dongmei Zhang. CI-Work: Benchmarking Contextual Integrity in Enterprise LLM Agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 1483–1508. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-industry.103. URL https://aclanthology.org/2026.acl-industry.103.pdf.

Dylan Girrens and Guangjing Wang. SPA: Securing persistent LLM agents across queries with plan-first information-flow control, 2026. URL https://arxiv.org/abs/2608.27234v2.

Amr Gomaa, Ahmed Salem, and Sahar Abdelnabi. ConVerse: Benchmarking contextual safety in agent-to-agent conversations. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 3246–3268, 2026. doi: 10.18653/v1/2026.findings-eacl.170. URL https://aclanthology.org/2026.findings-eacl.170/.

Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. AI control: Improving safety despite intentional subversion. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16295–16336. PMLR, 2024. URL https://proceedings.mlr.press/v235/greenblatt24a.html.

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/2302.12173v2.

Feng He, Tianqing Zhu, Dayong Ye, Bo Liu, Wanlei Zhou, and Philip S. Yu. The Emerged Security and Privacy of LLM Agent: A Survey with Case Studies. arXiv:2407.19354v2, 2025. URL https://arxiv.org/html/2407.19354v2.

Yu He, Haozhe Zhu, Yiming Li, Shuo Shao, Hongwei Yao, Zhihao Liu, and Zhan Qin. AttriGuard: Defeating indirect prompt injection in LLM agents via causal attribution of tool invocations. In *35th USENIX Security Symposium*, 2026. URL https://www.usenix.org/conference/usenixsecurity26/presentation/he-yu.

Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. Model context protocol (MCP): Landscape, security threats, and future research directions, 2025. URL https://arxiv.org/abs/2503.23278v3.

Wenyue Hua, Xianjun Yang, Mingyu Jin, Zelong Li, Wei Cheng, Ruixiang Tang, and Yongfeng Zhang. Trustagent: Towards safe and trustworthy LLM-based agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10000–10016, 2024. doi: 10.18653/v1/2024.findings-emnlp.585. URL https://aclanthology.org/2024.findings-emnlp.585/.

Zheng Hui, Yijiang River Dong, Sanhanat Sivapiromrat, Ehsan Shareghi, and Nigel Collier. Privacy-R1: Privacy-aware multi-LLM agent collaboration via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45924–45937, 2026. doi: 10.18653/v1/2026.acl-long.2130. URL https://aclanthology.org/2026.acl-long.2130/.

Amjad Ibrahim and Yong Li. Overlaying governance: A compositional authorization framework for delegation and scope in agentic AI, 2026. URL https://arxiv.org/abs/2606.03518v1.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama guard: LLM-based input-output safeguard for human-AI conversations. *arXiv preprint arXiv:2312.06674*, 2023. URL https://arxiv.org/abs/2312.06674v1.

Rishi Jha, Harold Triedman, Justin Wagle, and Vitaly Shmatikov. Breaking and Fixing Defenses Against Control-Flow Hijacking in Multi-Agent Systems. arXiv:2510.17276v1, 2025. URL https://arxiv.org/abs/2510.17276v1.

Feiran Jia, Tong Wu, Xin Qin, and Anna Squicciarini. The task shield: Enforcing task alignment to defend against indirect prompt injection in LLM agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29680–29697, 2025. doi: 10.18653/v1/2025.acl-long.1435. URL https://aclanthology.org/2025.acl-long.1435/.

Maurits Kaptein, Vassilis-Javed Khan, and Andriy Podstavnychy. Runtime governance for AI agents: Policies on paths, 2026. URL https://arxiv.org/abs/2603.16586v1.

Juhee Kim, Wenbo Guo, and Dawn Song. SoK: Attack and defense landscape of agentic AI systems. In *35th USENIX Security Symposium*, 2026. URL https://www.usenix.org/conference/usenixsecurity26/presentation/kim-juhee-agentic.

Aashish Kolluri, Rishi Sharma, Manuel Costa, Boris Köpf, Tobias Nießen, Mark Russinovich, Shruti Tople, and Santiago Zanella-Béguelin. Optimizing Agent Planning for Security and Autonomy. arXiv:2602.11416v1, 2026. URL https://arxiv.org/abs/2602.11416v1.

Thomas Kuntz, Agatha Duzan, Hao Zhao, Francesco Croce, Zico Kolter, Nicolas Flammarion, and Maksym Andriushchenko. OS-Harm: A benchmark for measuring safety of computer use agents. *arXiv preprint arXiv:2506.14866*, 2025. URL https://arxiv.org/abs/2506.14866v1.

Ido Levy, Ben Wiesel, Sami Marreed, Alon Oved, Avi Yaeli, and Segev Shlomov. ST-WebAgentBench: A benchmark for evaluating safety and trustworthiness in web agents. *arXiv preprint arXiv:2410.06703*, 2024. URL https://arxiv.org/abs/2410.06703v1.

Evan Li, Tushin Mallick, Evan Rose, William Robertson, Alina Oprea, and Cristina Nita-Rotaru. ACE: A security architecture for LLM-integrated app systems. In *Network and Distributed System Security Symposium*, 2026a. doi: 10.14722/ndss.2026.230352. URL https://www.ndss-symposium.org/wp-content/uploads/2026-s352-paper.pdf.

Frank Li. OpenClaw PRISM: A zero-fork, defense-in-depth runtime security layer for tool-augmented LLM agents, 2026. URL https://arxiv.org/abs/2603.11853v1.

Hao Li, Xiaogeng Liu, Hung-Chun Chiu, Dianqi Li, Ning Zhang, and Chaowei Xiao. DRIFT: Dynamic rule-based defense with injection isolation for securing LLM agents. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.12104v3.

Junjie Li, Xi Xiao, Yunbei Zhang, Chen Liu, Lin Zhao, Xiaoying Liao, Yingrui Ji, Janet Wang, Yingqiang Ge, Weijie Xu, Xi Fang, Xiang Xu, Tianchen Zhao, Youngeun Kim, Jihun Hamm, Tianyang Wang, and Chandan K Reddy. Agent Harness Engineering: A Survey. Author-hosted preprint, repository revision 4884d874, 2026b. URL https://github.com/Picrew/LLM-Harness/blob/4884d8748b92ee375b8f460c232ab3ca96d186d7/docs/main.pdf.

Ying Li, Yanju Chen, Hongbo Wen, Bosi Zhang, Hanzhi Liu, Peiran Wang, Yu Feng, and Yuan Tian. VIGIL: Runtime enforcement of behavioral specifications in AI agent skills, 2026c. URL https://arxiv.org/abs/2606.26524v1.

Ilija Lichkovski, Alexander Müller, Mariam Ibrahim, and Tiwai Mhundwa. EU-agent-bench: Measuring illegal behavior of LLM agents under EU law. *arXiv preprint arXiv:2510.21524*, 2025.

Jay Ligatti, Lujo Bauer, and David Walker. Edit automata: Enforcement mechanisms for run-time security policies. *International Journal of Information Security*, 4(1–2):2–16, 2005. doi: 10.1007/s10207-004-0046-8.

Yuchen Ling, Shengcheng Yu, Zhenyu Chen, and Chunrong Fang. Toward secure LLM agents: Threat surfaces, attacks, defenses, and evaluation, 2026. URL https://arxiv.org/abs/2606.10749v2.

Yanming Liu, Xinyue Peng, Jiannan Cao, Xinyi Wang, Songhang Deng, Jintao Chen, Jianwei Yin, and Xuhong Zhang. ToolGate: Contract-grounded and verified tool execution for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 9653–9684, 2026. doi: 10.18653/v1/2026.findings-acl.470. URL https://aclanthology.org/2026.findings-acl.470/.

Alireza Lotfi, Subangkar Karmaker Shanto, Imtiaz Karim, and Elisa Bertino. Securing agentic AI: From per-action checks to trajectory assurance. *arXiv preprint arXiv:2608.01558*, 2026. URL https://arxiv.org/abs/2608.01558v1.

Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, Rongcheng Tu, Xiao Luo, Wei Ju, Zhiping Xiao, Yifan Wang, Meng Xiao, Chenwu Liu, Jingyang Yuan, Shichang Zhang, Yiqiao Jin, Fan Zhang, Xian Wu, Hanqing Zhao, Dacheng Tao, Philip S. Yu, and Ming Zhang. Large Language Model Agent: A Survey on Methodology, Applications and Challenges. arXiv:2503.21460v1, 2025. URL https://arxiv.org/abs/2503.21460v1.

Yingning Ma. RealSafe: Quantifying Safety Risks of Language Agents in Real-World. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 9586–9617. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.coling-main.642.pdf.

Kenneth L. McMillan. Circular compositional reasoning about liveness. In *Correct Hardware Design and Verification Methods (CHARME 1999)*, volume 1703 of *Lecture Notes in Computer Science*, pp. 342–346. Springer, 1999. doi: 10.1007/3-540-48153-2_30. URL https://link.springer.com/chapter/10.1007/3-540-48153-2_30.

Qianyu Meng, Yanan Wang, Liyi Chen, Yihang Li, Wei Wu, Wenyuan Jiang, Qimeng Wang, Chengqiang Lu, Yan Gao, Yi Wu, and Yao Hu. Agent Harness for Large Language Model Agents: A Survey. Preprints.org 202604.0428, version 3, 2026. URL https://www.preprints.org/manuscript/202604.0428.

Yutao Mou, Zhangchi Xue, Lijun Li, Peiyang Liu, Shikun Zhang, Wei Ye, and Jing Shao. ToolSafe: Enhancing tool invocation safety of LLM-based agents via proactive step-level guardrail and feedback. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 37125–37153, 2026. doi: 10.18653/v1/2026.findings-acl.1850. URL https://aclanthology.org/2026.findings-acl.1850/.

Xabier Muruaga. Bounded agents: Delegation security for multi-agent AI systems. *arXiv preprint arXiv:2608.15888*, 2026. URL https://arxiv.org/abs/2608.15888v1.

Praneeth Narisetty, Shiva Nagendra Babu Kore, Uday Kumar Reddy Kattamanchi, and Jayaram Kumarapu. Adaptive evaluation of out-of-band defenses against prompt injection in LLM agents. *arXiv preprint arXiv:2606.26479*, 2026. URL https://arxiv.org/abs/2606.26479v1.

Albus W. Ng, Yi Han, Jusheng Zhang, and Wenhao Wang. Agent safety should be a runtime contract, 2026. URL https://arxiv.org/abs/2608.11274v1.

Traian Rebedea, Razvan Dinu, Makesh Narsimhan Sreedhar, Christopher Parisien, and Jonathan Cohen. Nemo guardrails: A toolkit for controllable and safe LLM applications with programmable rails. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 431–445, 2023. doi: 10.18653/v1/2023.emnlp-demo.40. URL https://aclanthology.org/2023.emnlp-demo.40/.

Alex Remedios, Simon Storf, Fabien Roger, and John Hughes. Red-teaming auto mode: Improving blocking classifiers against malign coding agents, 2026. URL https://arxiv.org/abs/2609.19587v1.

Franziska Roesner and David Kohlbrenner. Agentic Browsers and the Same-Origin Policy. In *ICLR 2026 Workshop on Agents in the Wild*, 2026. URL https://homes.cs.washington.edu/%7Efranzi/pdf/roesner_kohlbrenner_2026_agentic_sop.pdf.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. Identifying the risks of LM agents with an LM-emulated sandbox. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GEcwtMk1uA.

Andrei Sabelfeld and Andrew C. Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications*, 21(1):5–19, 2003.

Tanmay Sah, Vishal Srivastava, Dolly Sah, and Kayden Jordan. The verifier tax: Horizon dependent safety success tradeoffs in tool using LLM agents. *arXiv preprint arXiv:2603.19328*, 2026. URL https://arxiv.org/abs/2603.19328v1.

Jerome H. Saltzer and Michael D. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, 1975. URL https://web.mit.edu/Saltzer/www/publications/protection/.

Fred B. Schneider. Enforceable security policies. *ACM Transactions on Information and System Security*, 3(1):30–50, 2000. URL https://www.cs.cornell.edu/fbs/publications/EnfSecPols.pdf.

Tianneng Shi, Jingxuan He, Zhun Wang, Hongwei Li, Linyu Wu, Wenbo Guo, and Dawn Song. Progent: Securing AI agents with privilege control. *arXiv preprint arXiv:2504.11703*, 2026. URL https://arxiv.org/abs/2504.11703v3.

Minsun Shim, Ramisha Raida Karim, Ruthwik Jakkula, Kaiwen Zhou, Xin Liu, Xin Eric Wang, and Zhou Li. Confuse the model, control the flow: Understanding and mitigating privacy leakage from LLM agents with information flow control, 2026. URL https://arxiv.org/abs/2609.14003v1.

Shoaib Ahmed Siddiqui, Radhika Gaonkar, Boris Köpf, David Krueger, Andrew Paverd, Ahmed Salem, Shruti Tople, Lukas Wutschitz, Menglin Xia, and Santiago Zanella-Béguelin. Permissive information-flow analysis for large language models. *Transactions on Machine Learning Research*, October 2025. URL https://openreview.net/forum?id=ufYRO8y3mr.

Vincent Siu, Jingxuan He, Kyle Montgomery, Zhun Wang, Neil Gong, Chenguang Wang, and Dawn Song. A framework for formalizing LLM agent security. *arXiv preprint arXiv:2603.19469*, 2026. URL https://arxiv.org/abs/2603.19469v1.

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/2501.09674v1.

Simon Storf, Rich Barton-Cooper, James Peters-Gill, and Marius Hobbhahn. Constitutional black-box monitoring for scheming in LLM agents, 2026. URL https://arxiv.org/abs/2603.00829v2.

Rakesh Kumar Surapani, Pradeep Kumar Dolabehera Kakitapelli, Arun Morampudi, and Praveena Padi. Authorization architectures for tool-using AI agents, 2026. URL https://arxiv.org/abs/2609.15906v1.

Georgios Syros, Anshuman Suri, Jacob Ginesin, Cristina Nita-Rotaru, and Alina Oprea. SAGA: A security architecture for governing AI agentic systems. In *Network and Distributed System Security Symposium*, 2026. doi: 10.14722/ndss.2026.230869. URL https://www.ndss-symposium.org/ndss-paper/saga-a-security-architecture-for-governing-ai-agentic-systems/.

Zhuoran Tan, Run Hao, Jeremy Singer, Yutian Tang, and Christos Anagnostopoulos. MCP-SandboxScan: WASM-based secure execution and runtime analysis for MCP tools. *arXiv preprint arXiv:2601.01241*, 2026. URL https://arxiv.org/abs/2601.01241v2.

Lillian Tsai and Eugene Bagdasarian. Contextual agent security: A policy for every purpose. In *Proceedings of the Workshop on Hot Topics in Operating Systems*, HOTOS '25, pp. 8–17. ACM, 2025. doi: 10.1145/3713082.3730378. URL https://sigops.org/s/conferences/hotos/2025/papers/hotos25-100.pdf.

Federico Villa, F. Betül Durak, Tadayoshi Kohno, Tapdig Maharramli, and Franziska Roesner. Extending the formalism and theoretical foundations of cryptography to AI. *arXiv preprint arXiv:2603.02590*, 2026. URL https://arxiv.org/abs/2603.02590v1.

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. Agentspec: Customizable runtime enforcement for safe and reliable LLM agents. In *Proceedings of the 48th International Conference on Software Engineering*, pp. 2938–2950. ACM, 2026a. doi: 10.1145/3744916.3764546. URL https://arxiv.org/abs/2503.18666v3.

Haoyu Wang, Christopher M. Poskitt, Jiali Wei, and Jun Sun. ProbGuard: Proactive runtime monitoring for LLM agent safety via probabilistic prediction. In *41st IEEE/ACM International Conference on Automated Software Engineering (ASE), to appear*, 2026b. URL https://arxiv.org/abs/2508.00500v4.

Haoyu Wang, Zibo Xiao, Yedi Zhang, Christopher M. Poskitt, and Jun Sun. SafeClaw-R: Towards safe and secure multi-agent personal assistants. *arXiv preprint arXiv:2603.28807*, 2026c. URL https://arxiv.org/abs/2603.28807v1.

Kun Wang, Guibin Zhang, Zhenhong Zhou, Jiahao Wu, Miao Yu, Shiqian Zhao, Chenlong Yin, Jinhu Fu, Yibo Yan, Hanjun Luo, Liang Lin, Zhihao Xu, Haolang Lu, Xinye Cao, Xinyun Zhou, Weifei Jin, Fanci Meng, Shicheng Xu, Junyuan Mao, Yu Wang, Hao Wu, Minghe Wang, Fan Zhang, Junfeng Fang, Wenjie Qu, Yue Liu, Chengwei Liu, Yifan Zhang, Qiankun Li, Chongye Guo, Yalan Qin, Zhaoxin Fan, Kai Wang, Yi Ding, Donghai Hong, Jiaming Ji, Yingxin Lai, Zitong Yu, Xinfeng Li, Yifan Jiang, Yanhui Li, Xinyu Deng, Junlin Wu, Dongxia Wang, Yihao Huang, Yufei Guo, Jen-tse Huang, Qiufeng Wang, Xiaolong Jin, Wenxuan Wang, Dongrui Liu, Yanwei Yue, Wenke Huang, Guancheng Wan, Heng Chang, Tianlin Li, Yi Yu, Chenghao Li, Jiawei Li, Lei Bai, Jie Zhang, Qing Guo, Jingyi Wang, Tianlong Chen, Joey Tianyi Zhou, Xiaojun Jia, Weisong Sun, Cong Wu, Jing Chen, Xuming Hu, Yiming Li, Xiao Wang, Ningyu Zhang, Anh Tuan Luu, Guowen Xu, Jiaheng Zhang, Tianwei Zhang, Xingjun Ma, Jindong Gu, Liang Pang, Xiang Wang, Bo An, Jun Sun, Mohit Bansal, Shirui Pan, Lingjuan Lyu, Yuval Elovici, Bhavya Kailkhura, Yaodong Yang, Hongwei Li, Wenyuan Xu, Yizhou Sun, Wei Wang, Qing Li, Ke Tang, Yu-Gang Jiang, Felix Juefei-Xu, Hui Xiong, Xiaofeng Wang, Dacheng Tao, Philip S. Yu, Qingsong Wen, and Yang Liu. A comprehensive survey in LLM(-Agent) full stack safety: Data, training and deployment, 2025a. URL https://arxiv.org/abs/2504.15585v4.

Peiran Wang, Yang Liu, Yunfei Lu, Yifeng Cai, Hongbo Chen, Qingyou Yang, Jie Zhang, Jue Hong, and Ye Wu. Securing large language model agents via structured graph abstraction. *arXiv preprint arXiv:2508.01249*, 2025b. URL https://arxiv.org/html/2508.01249v3.

Xilong Wang, John Bloch, Zedian Shao, Yuepeng Hu, Shuyan Zhou, and Neil Zhenqiang Gong. WebInject: Prompt injection attack to web agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2010–2030. Association for Computational Linguistics, 2025c. doi: 10.18653/v1/2025.emnlp-main.104. URL https://aclanthology.org/2025.emnlp-main.104/.

Xinyuan Wang, Bowen Wang, Dunjie Lu, Junlin Yang, Tianbao Xie, Junli Wang, Jiaqi Deng, Xiaole Guo, Yiheng Xu, Chen Henry Wu, Zhennan Shen, Zhuokai Li, Ryan Li, Xiaochuan Li, Junda Chen, Boyuan Zheng, Peihang Li, Fangyu Lei, Ruisheng Cao, Yeqiao Fu, Dongchan Shin, Martin Shin, Jiarui Hu, Yuyan Wang, Jixuan Chen, Yuxiao Ye, Danyang Zhang, Dikang Du, Hao Hu, Huarong Chen, Zaida Zhou, Yipu Wang, Heng Wang, Diyi Yang, Victor Zhong, Flood Sung, Y.Charles, Zhilin Yang, and Tao Yu. OpenCUA: Open Foundations for Computer-Use Agents. arXiv:2508.09123v1, 2025d. URL https://arxiv.org/abs/2508.09123v1.

Xunguang Wang, Zhenlan Ji, Wenxuan Wang, Zongjie Li, Daoyuan Wu, and Shuai Wang. Sok: Evaluating jailbreak guardrails for large language models. *arXiv preprint arXiv:2506.10597*, 2025e.

Yuhui Wang, Xingqi Wu, Junaid Farooq, and Juntao Chen. PROMPTGUARD: Zero Trust Prompting for Securing LLM-Driven O-RAN Control. In *Workshop on Security and Privacy of Next-Generation Networks (FutureG)*, 2026d. URL https://www.ndss-symposium.org/wp-content/uploads/futureg26-22.pdf.

Yuntao Wang, Yanghe Pan, Zhou Su, Yi Deng, Quan Zhao, Linkang Du, Tom H. Luan, Jiawen Kang, and Dusit Niyato. Large Model-Based Agents: State-of-the-Art, Cooperation Paradigms, Security and Privacy, and Future Trends. arXiv:2409.14457v3, 2025f. URL https://arxiv.org/pdf/2409.14457v3.

Fangzhou Wu, Ethan Cecchetti, and Chaowei Xiao. System-level defense against indirect prompt injection attacks: An information flow control perspective. *arXiv preprint arXiv:2409.19091*, 2024. URL https://arxiv.org/abs/2409.19091v1.

Jie Wu and Ming Gong. Beyond single-agent alignment: Preventing context-fragmented violations in multi-agent systems. *arXiv preprint arXiv:2604.22879*, 2026a.

Jie Wu and Ming Gong. Policy-invisible violations in LLM-based agents. *arXiv preprint arXiv:2604.12177*, 2026b.

Yuhao Wu, Franziska Roesner, Tadayoshi Kohno, Ning Zhang, and Umar Iqbal. IsolateGPT: An execution isolation architecture for LLM-based agentic systems. In *Network and Distributed System Security Symposium*, 2025. doi: 10.14722/ndss.2025.241131. URL https://arxiv.org/abs/2403.04960v2.

Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, Dawn Song, and Bo Li. GuardAgent: Safeguard LLM agents via knowledge-enabled reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 68316–68342, 2025. URL https://proceedings.mlr.press/v267/xiang25a.html.

Chenglin Yang. AgentTrust: Runtime safety evaluation and interception for AI agent tool use, 2026. URL https://arxiv.org/abs/2605.04785v1.

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/2406.12045v1.

Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. A survey on evaluation of LLM-based agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 26690–26714, 2026. URL https://aclanthology.org/2026.findings-acl.1330/.

Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, Rui Wang, and Gongshen Liu. R-judge: Benchmarking safety risk awareness for LLM agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1467–1490, 2024. doi: 10.18653/v1/2024.findings-emnlp.79. URL https://aclanthology.org/2024.findings-emnlp.79/.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10471–10506, 2024. doi: 10.18653/v1/2024.findings-acl.624. URL https://aclanthology.org/2024.findings-acl.624/.

Hanrong Zhang, Jingyuan Huang, Kai Mei, Yifei Yao, Zhenting Wang, Chenlu Zhan, Hongwei Wang, and Yongfeng Zhang. Agent security bench (ASB): Formalizing and benchmarking attacks and defenses in LLM-based agents. In *International Conference on Learning Representations*, 2025a. URL https://arxiv.org/abs/2410.02644v4.

Yedi Zhang, Sun Yi Emma, Annabelle Lee Jia En, and Jin Song Dong. RvLLM: LLM runtime verification with domain knowledge. In *NeurIPS*, 2025b.

Zhexin Zhang, Shiyao Cui, Yida Lu, Jingzhuo Zhou, Junxiao Yang, Hongning Wang, and Minlie Huang. Agent-safetybench: Evaluating the safety of LLM agents. *arXiv preprint arXiv:2412.14470*, 2024. URL https://arxiv.org/abs/2412.14470v2.

Wei Zhao, Zhe Li, Peixin Zhang, and Jun Sun. ClawGuard: A runtime security framework for tool-augmented LLM agents against indirect prompt injection, 2026. URL https://arxiv.org/abs/2604.11790v2.

Arman Zharmagambetov, Chuan Guo, Ivan Evtimov, Maya Pavlova, Ruslan Salakhutdinov, and Kamalika Chaudhuri. AgentDAM: Privacy Leakage Evaluation for Autonomous Web Agents. ICML 2025 Workshop on Computer Use Agents manuscript; arXiv:2503.09780; also accepted to NeurIPS 2025, Datasets and Benchmarks Track, 2025. URL https://openreview.net/pdf/06bde18559fc6ddf4c9a1168450a34a055225fd9.pdf.

Peter Yong Zhong, Siyuan Chen, Ruiqi Wang, McKenna McCall, Ben L. Titzer, Heather Miller, and Phillip B. Gibbons. RTBAS: Defending LLM agents against prompt injection and privacy leakage. *arXiv preprint arXiv:2502.08966*, 2025. URL https://arxiv.org/abs/2502.08966v1.

Xueyang Zhou, Weidong Wang, Lin Lu, Jiawen Shi, Guiyao Tie, Yongtian Xu, Lixing Chen, Pan Zhou, Neil Zhenqiang Gong, and Lichao Sun. SafeAgent: Safeguarding LLM agents via an automated risk simulator. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32516–32543, 2026. URL https://aclanthology.org/2026.acl-long.1501/.

Kaijie Zhu, Xianjun Yang, Jindong Wang, Wenbo Guo, and William Yang Wang. MELON: Provable defense against indirect prompt injection attacks in AI agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 80310–80329, 2025. URL https://arxiv.org/abs/2502.05174v4.

Naama Zwerdling, David Boaz, Ella Rabinovich, Guy Uziel, David Amid, and Ateret Anaby-Tavor. Towards enforcing company policy adherence in agentic workflows. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 595–606, 2025. URL https://aclanthology.org/2025.emnlp-industry.41/.

A Source Anchors for the Core Comparison

Table 3 pins the versions and passages used for the core comparison. Publication and inspected versions can differ: DRIFT's March 2026 v3 extends its NeurIPS 2025 record; AgenTEE uses v2. Full comparison records preserve source statements separately from our interpretation.

Table 3: Selected systems, supporting components, and boundary case. An architecture paper's experiments do not verify its assumptions in a different deployment.

Source	Inspected identifier	Reading anchor and role
NeMo Guardrails	EMNLP 2023 demo paper	Sections 3–5, 7, E–G: Colang runtime, custom actions, evaluations; supporting toolkit.
TrustAgent	Findings EMNLP 2024	Sections 3.1–3.4, 4: retrieved regulations, inspection/revision, emulated tools.
AgentSpec	arXiv:2503.18666v3	Sections 3.3–4: formal trajectory inputs, event-dependent state/action inputs, enforcement.
GuardAgent	arXiv:2406.09187v3	Sections 4–5: generated checks and log-based evaluation; ICML 2025 metadata.
Progent	arXiv:2504.11703v3	Sections 4–7, 8.3: policy updates, confinement, approver, unified multi-agent setting.
RvLLM	arXiv:2505.18585v3	Sections 1–4: expert rules, LLM perception, reasoning; output verifier.
CaMeL	arXiv:2503.18813v2	Control/data separation and capability policy; execution architecture.
Fides	arXiv:2505.23643v2	Sections 4–6: labels, selective visibility, and planner expressiveness.
Permissive IFA	arXiv:2410.03055v3	Sections 2–3.4, 4–6: label search and restricted-context regeneration.
AgentArmor	arXiv:2508.01249v3	Sections III, V, VI-E: inferred graph, trusted registry, pre-call gate, delegation choice.
IPIGuard	EMNLP 2025 final	Sections 3–4: planned graph, argument estimation, query expansion, simulated calls.
DRIFT	arXiv:2506.12104v3	Sections 3–4, A.1–A.3: plan, learned revision, masking, approval-as-rejection.
Task Shield	ACL 2025 final paper	Sections 3–4: task contribution and message-specific intervention.

Continued on next page

Table 3 continued: inspected sources

Source	Inspected identifier	Reading anchor and role
MELON	arXiv:2502.05174v4	Sections 3.2–3.4, 4.2–4.3: masked comparison, ensemble bound, single-prompt variants.
ToolSafe	Findings ACL 2026 final	Sections 3–6: TS-Bench, learned guard, blocking and feedback.
IsolateGPT	arXiv:2403.04960v2	Sections III–V: scope, isolation, hub-mediated permissions, warning metric; NDSS 2025 metadata.
ACE	NDSS 2026 final	Sections II–V: attack surfaces, static flow checks, isolation; distinct injection/Tool Usage metrics.
AgenTEE	arXiv:2604.18231v2	Sections 3–5: attestation, protected channels, prototype latency; supporting substrate.
Authenticated delegation	arXiv:2501.09674v1	Sections 3–5: identity/scope tokens and inter-agent scoping; proposed architecture.
Bounded Agents (APC)	arXiv:2608.15888v1	Sections 3–7: delegation algebra, conditional composition, enforcement evaluation.
DWT broker	arXiv:2609.00267v1	Sections 3–8, 10–11: identity, broker tests, mixed framework evidence, unreleased artifact.
ProbGuard	arXiv:2508.00500v4	Sections 3, 5–7: prediction, intervention, sample bounds; boundary case.
SafeClaw-R	arXiv:2603.28807v1	Sections 3–4: node mediation, verdicts, review, mutation-based evaluation.
Sentinel	arXiv:2604.12177v1	Sections 4, 5.1, 6.3–6.5: graph checks, replay, state coverage.
Distributed Sentinel	arXiv:2604.22879v1	Sections 2.3–2.5, 4–5, 7: composition premises, grounding, scope.

Benchmark anchors are R-Judge's final EMNLP 2024 paper, Sections 3.3–3.5; TraceSafe, arXiv:2604.07223v2 (COLM 2026), Sections 3–4 and Appendices A/E; AgentDojo, arXiv:2406.13352v3, Section 3; AgentHarm, arXiv:2410.09024v3, Section 3.1; EU-Agent-Bench, arXiv:2510.21524v1, Sections 2/4; and MAST, arXiv:2503.13657v3 (NeurIPS 2025 Datasets and Benchmarks Track), Sections 3–4 and Appendix J.1. MELON is published at ICML 2025; method reading remains pinned to v4. Permissive IFA appeared in TMLR in October 2025; we inspect the later arXiv v3. ProbGuard is listed in the ASE 2026 research program (to appear); we inspect v4. TS-Bench and PhantomPolicy/PhantomEcosystem stay within their mechanism records. R-Judge separates human labels from model-scored explanations; MAST separates human taxonomy development from scaled LLM annotation.

Related syntheses include Dantas et al. (2026), arXiv:2608.14590v1; Lotfi et al. (2026), arXiv:2608.01558v1; and Siu et al. (2026), arXiv:2603.19469v1. Siu et al. connect contextual properties and ideal oracles (Section 4), flag action-sequence composition as open (Section 7), and leave multi-agent delegation and guarantee composition unaddressed (Section 8). Our implementation-facing comparison builds on these foundations, not a prior composition solution. Changing versions requires rechecking claims rather than retaining old scores under newer citations.

DWT's framework evidence concerns LangGraph 1.2.10, CrewAI 1.15.13, AutoGen 0.7.5, and MCP authorization dated 2026-07-28, not current versions generally. AgenTEE's preliminary timing study is not a tool-policy attack evaluation. DRIFT's two adaptive-attack types target the Injection Isolator; PAIR selects prior defense-success/base-failure cases (Section 4.6). We do not infer unrestricted adaptive robustness or transfer comparisons with an earlier Progent into a ranking of Progent v3. Progent also supports multiple stakeholders (Section 7) and describes an unevaluated agent-local integration alternative (Section 8.3); its main approval setting automatically approves requests (Section 8.1). These capabilities and settings do not establish cross-organization interoperability.

B Coding and Interpretation Rules

The comparison record $\langle P, O, I, T, E \rangle$ is populated as follows. P names the encoded property, not the paper's broadest motivation. O records the inputs actually supplied to a check: current action, preceding history, represented world state, and provenance. Cross-agent is an additional flag about

where that evidence originates, not a mutually exclusive temporal horizon. I records the controlled event and whether denial suspends execution, requests advice, or merely reports a completed violation. T names trusted labeling, extraction, policy translation, state services, and gate assumptions. E distinguishes mathematical claims, architectural arguments, and empirical results in the inspected setting.

Symbolic denotes explicit predicates or structured reasoning; IFC, propagation and enforcement of dependency-associated restrictions; empirical, learned judgments or predictions evaluated on data. Labels may overlap and do not score guarantee strength. Graphs are not automatically IFC, nor does a legal benchmark policy make its decision mechanism symbolic. Substrate isolation is recorded under I/T, not forced into a decision modality.

Source claims and review inferences are stored separately. Fides supplies trusted runtime labeling; preserving compatible metadata at an additional organizational boundary is our inference, not an observed vulnerability. Likewise, a verifier does not establish complete mediation, and an unavailable artifact does not demonstrate algorithmic failure.

B.1 Independent Human Coding

Two external human coders, A and B, neither an author of this paper, independently annotated all 25 core and 43 bounded-extension records using the frozen `upgrade-1` codebook. Each packet contained source identifiers, pinned versions and reading anchors, but no existing classifications. Coding was manual, with no viewing of the other coder's answers, discussion of labels or reference to the existing classifications, as confirmed by the author. This classification exercise is separate from eligibility screening, source-assertion verification and the nine named-source additions.

Coders first marked applicability of the mechanism dimensions, then assigned sets of H, M, D1, D2 and D3 labels with source anchors and rationales. Evaluation- or synthesis-only records could be marked not applicable with a reason; unknown and not applicable were not negative labels. Agreement on applicability was 65/68: 44 records were jointly applicable, 21 jointly not applicable, and three differed. Dimensional statistics use only the 44 jointly applicable records, without imputing the other 24 as matching label sets.

Table 4: Pre-adjudication independent human coding. Raw agreement and nominal exact-set κ; dimensional results condition on joint applicability.

Dimension	n	Raw agreement	κ	95% bootstrap interval
Applicability	68	95.6%	0.901	[0.767, 1.000]
H	44	81.8%	0.675	[0.460, 0.848]
M	44	79.5%	0.742	[0.589, 0.883]
D1	44	54.5%	0.476	[0.317, 0.626]
D2	44	47.7%	0.426	[0.278, 0.570]
D3	44	88.6%	0.728	[0.514, 0.923]

For each multi-label dimension, nominal exact-set Cohen's κ treats each distinct complete label set as one category, ignoring label order. We also release every per-label binary result. Percentile intervals use 2,000 paired record-bootstrap draws with seed 230923; dimensional intervals condition on the jointly applicable subset. Undefined resamples are counted and omitted from interval estimation, not replaced. All draws for Table 4 were defined; sparse per-label categories sometimes produced constant draws, and one category had undefined κ. A collapsed interval from sparse labels is not precise population reliability. These purposively selected records do not support prevalence estimates for the whole literature.

D1 and D2 show lower exact-set agreement than H, M and D3. Per-label agreement is particularly weak for effect correctness (D1, $\kappa = 0.200$) and semantic grounding (D2, $\kappa = 0.288$). Initially, 35 records differed: three on applicability and 32 within the jointly applicable subset, whose H/M/D1/D2/D3 disagreement counts were 8/9/20/23/5, with overlap. An author subsequently adjudicated all 35 against the frozen codebook and pinned-source anchors, recording the checking unit, final labels and rationale. Runtime Contract and AgentDAM were coded applicable; Auto-mode remained not applicable, yielding 46 applicable and 22 non-applicable records. This is post-coding

author adjudication, not renewed independent agreement or A/B consensus. Table 4 and the original annotations are unchanged. The companion exports the final classifications separately from the source-description synthesis; its nine named-source additions remain outside this exercise.

Core sensitivity and final labels. Thirteen of the 25 core records initially differed on D1/D2: nine on D1 and five on D2, with one overlap. Core H, M and D3 agree record by record and are unchanged by adjudication. The final column records author decisions, not an agreement statistic:

Core label	Coder A	Coder B	Final
D1: Authorization	16	15	14
D1: Effect correctness	1	5	4
D2: Semantic grounding	6	10	7

Counts use all 25 core records; labels can overlap. They diagnose coding sensitivity, not system prevalence. D1/D2 remain source-anchored qualitative annotations; core H/M/D3 agree before adjudication. The companion retains both returns, the disagreement queue and `core_coding_sensitivity.json`; `adjudications.json` stores the 35 decisions and `adjudicated_records.json` the final 68-record mapping. The verifier reconstructs it from original agreements and explicit decisions. Adjudication does not complete eligibility screening or citation-route review.

B.2 Complete Mechanism Matrix and Evaluation Map

Figures 1 and 2 present the mechanism map and conceptual handoff example in the main text. Table 5 retains every core record and its descriptive comparison fields. The evaluation map below has a separate evidence role from independent human classification.

Table 5: Core codes and qualitative annotations for 25 records, including two contextual boundary cases. H: A = current input, T = accumulated context, C = configured scope. M: S = explicit checks, F = flow restrictions, E = learned verdict/score. H/M/D3 agree between coders for this core set. D1/D2 are qualitative target/premise annotations. D3: workflow-local (W), delegated (D), or cross-agent/provider (X) authority.

System	H	M	D1: target	D2: inferred premise	D3
NeMo Guardrails	C	S+E	Dialogue rails	Semantic mapping	W
TrustAgent	T	E	Constitution	Rule compliance	W
AgentSpec	C	S	Safety predicates	Configuration	W
GuardAgent	T	S	Guard requests	Check generation	W
Progent	A	S	Tool privileges	Policy updates	W
RvLLM	A	S	Rule consistency	Fact extraction	W
AgentArmor	T	S+F	Dependency policy	Graph edges	W
CaMeL	T	S+F	Flow / privileges	Control extraction	W
Fides	T	F	Confidentiality / integrity	Planner proposals	W
Permissive IFA	A	F+E	Output flow	Utility estimates	W
IPIGuard	T	S	Plan constraints	Plan / arguments	W
DRIFT	T	S+E	Intent / call plan	Revision / masking	W
Task Shield	T	E	Task contribution	Task alignment	W
MELON	T	E	Injection behavior	Masked runs / similarity	W
ToolSafe	T	E	Call safety	Guard judgment	W
IsolateGPT	T	S	App permissions	Plans / warnings	X
ACE	T	S+F	Plan / flow policy	Plan generation	X
AgenTEE	A	–	Protected assets	Outside substrate	X
Authenticated delegation	A	S	Identity / scope	Outside protocol	D
APC	T	S	Session constraints	Action mapping*	D
DWT broker	A	S	Grant confinement	Outside broker	D
ProbGuard	A	S	Predicted risk	Transition model	W
SafeClaw-R	T	S+E	Workflow rules	Skill execution	W
Sentinel	T	S+F	State invariants	Grounding*	W
Distributed Sentinel	T	S+F	Cross-domain invariants	Entity grounding	X

*Mapping fidelity is required; the code does not assert a learned implementation. AgenTEE protects execution assets rather than evaluating semantic action policy. Full inputs, interventions, evidence and source anchors appear in Appendices A–C.

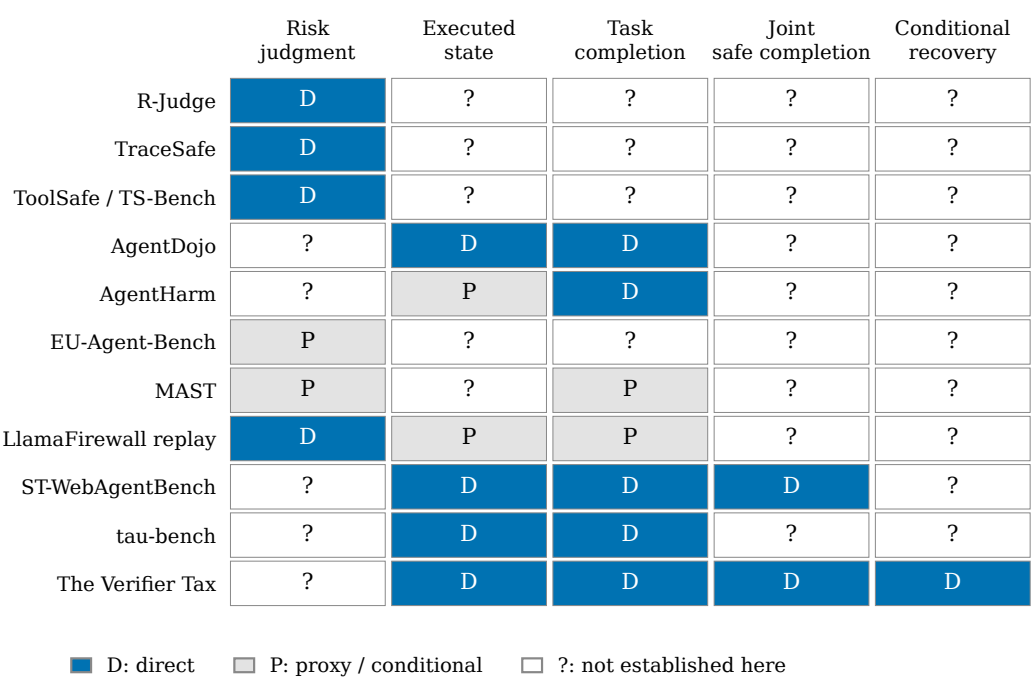

Figure 3: Source-linked coverage for 11 selected evaluation settings. D: direct measurement within the stated oracle and denominator. P: proxy, replay or selected-only evidence. ?: not established in this extraction, not demonstrated absence. Task completion includes harmful-task completion in AgentHarm; the Verifier Tax's conditional recovery also includes environment errors. This AI-assisted descriptive extraction is separate from A/B's H/M/D coding. Cell notes and source anchors are in `figures/data/evaluation_coverage.json` in the companion.

The figure compares measurement interfaces rather than ranking systems. In these 11 settings, five directly measure task completion, two directly measure joint safe completion and one directly measures conditional recovery within its own intervention/error denominator. Four directly inspect executed state. These counts concern the specified settings and coding rules, not every experiment in the cited papers or field-wide coverage.

C Per-System Comparison Matrices

Tables 6 and 7 expand the same 25 comparison records in the same order. They are complementary views of $\langle P, O, I, T, E \rangle$, not capability rankings. A component or proposal is identified as such; a missing deployment study is not coded as an absent capability. Source versions and reading anchors are in Table 3. The final column of Table 7 is our synthesis of transfer requirements, informed by source limitations, not a list of newly observed vulnerabilities.

Table 6: Individual mechanisms: property and decision, observations, and controlled event. Horizon and modality use the main-text HM codes; full inputs disambiguate configuration-dependent scope.

System	Horizon	Modality	Property / decision	Inputs to the check	Intervention and scope
NeMo Guardrails (Rebedea et al., 2023)	C	S+E	Developer-defined dialogue and safety rails; Colang flows plus custom checks.	Conversation events, canonical forms, action context; rail-specific inputs.	Flow execution and response handling; supporting application runtime, not a universal tool-policy gate.
TrustAgent (Hua et al., 2024)	T	E	Retrieved constitution rules; LLM relevance and compliance judgments.	User instruction, proposed action, trajectory and retrieved regulations.	Before execution: inspector requests revision; repeated mistakes halt the process. Single-agent planning.
AgentSpec (Wang et al., 2026a)	C	S	User-defined safety/reliability predicates and event triggers.	Formal input/trajectory arguments; event-dependent current state and pending action.	Configurable stopping, inspection, reconsideration or correction; these have different control semantics.
GuardAgent (Xiang et al., 2025)	T	S	LLM-generated executable checks for supplied guard requests.	Agent specifications, inputs, output logs and demonstrations.	Guard decision and explanation; deployment must connect the decision to an effect-controlling gate.
Progent (Shi et al., 2026)	A	S	Symbolic privilege policy; SMT checks permission non-expansion for unapproved updates.	Tool names/arguments, current policies and proposed updates.	Tool-call admission and policy-expansion approval; evaluated multi-agent setting shares one user.
RvLLM (Zhang et al., 2025b)	A	S	Domain-knowledge consistency via structured reasoning.	Expert rules and propositions instantiated by LLM perception.	Output verification and follow-up queries; supporting verifier, not itself a tool executor.
AgentArmor (Wang et al., 2025b)	T	S+F	Dependency-security constraints; inferred graph and deterministic label propagation.	Runtime trace, inferred dependencies, registered tools/data.	Blocks violating action nodes before execution; transfer execution depends on policy choice.
CaMeL (Debenedetti et al., 2025)	T	S+F	Control/data separation and capability-based security policies.	Program state, values and capability metadata.	Policy evaluation before tool execution in a trusted interpreter; deployment may request confirmation.
Fides (Costa et al., 2025)	T	F	Confidentiality / integrity IFC, selective visibility and policy checks.	Labeled values, planner state, tool arguments and policies.	Trusted runtime propagates labels and checks tool use before execution.
Permissive IFA (Siddiqui et al., 2025)	A	F+E	Search for useful restricted contexts and more permissive output labels.	Labeled context, prompt, original output and utility estimates.	Regenerates and returns a new output from restricted context; supporting wrapper, not a release gate.

Continued on next page

Table 6 continued: individual mechanisms

System	Horizon	Modality	Property / decision	Inputs to the check	Intervention and scope
IPIGuard (An et al., 2025)	T	S	Trusted-input tool-dependency plan restricts injected redirection.	Planned graph, pending arguments and dependent responses.	Planned execution with query-only expansion and simulated decoy calls; not content masking.
DRIFT (Li et al., 2025)	T	S+E	Query-derived constraints with learned intent/privilege judgments.	Original query, planned/executed calls, arguments and tool output.	Call admission, plan revision and masking before memory insertion; evaluated approval requests count as rejection.
Task Shield (Jia et al., 2025)	T	E	LLM checks contribution to user objectives.	Extracted task set and conversation context.	Blocks noncontributing calls; gives feedback for tool outputs. Task alignment is its decision target.
MELON (Zhu et al., 2025)	T	E	Detects injection through original/masked call comparison.	Proposed calls and a masked-call cache.	Terminates before execution on detected injection; not an authenticated resource-grant check.
ToolSafe (Mou et al., 2026)	T	E	TS-Guard predicts tool-call safety; TS-Flow consumes its verdict.	User request, history, candidate call and available tool specifications.	Blocks calls judged unsafe and supplies feedback; online handling of controversial labels is unclear.
IsolateGPT (Wu et al., 2025)	T	S	Execution/context isolation and hub-mediated permission checks.	Plans, typed messages, app context and permissions.	Controls inter-app communication and sharing; excludes within-app compromise.
ACE (Li et al., 2026a)	T	S+F	Trusted planning, static IFC and isolated worker execution.	Abstract/concrete plans, app clearances, query labels and worker messages.	Plan admission, worker privileges and communication; protects planning and execution stages.
AgenTEE (Abdollahi et al., 2026)	A	–	Attestation and confidential execution protect component assets.	Measurements, identities and channel establishment, not semantic policy facts.	Asset provisioning and protected execution/communication; supporting substrate across providers.
Authenticated delegation (South et al., 2025)	A	S	Proposed identity-linked, scoped authorization tokens.	Identity references, scope and validity metadata.	Mutual authentication and receiving-service validation; architecture proposal, not a tested universal protocol.
APC (Muruaga, 2026)	T	S	Delegation attenuation and specified session-composition constraints.	Principal chain, grant, budgets, history and declared intent.	Infrastructure-mediated tool admission across delegation hops within the session contract.
DWT broker (Dantuluri & Sundi, 2026)	A	S	Scoped grants, workload binding and revocation.	Tokens, caveats, workload identity, revocation state and calls.	Broker token exchange and tool admission across delegated identities.
ProbGuard (Wang et al., 2026b)	A	S	Predictive checking over a learned predicate-abstracted DTMC.	Abstract state and learned transition probabilities.	Stops or re-prompts based on predicted risk; embodied/driving boundary case.

Continued on next page

Table 6 continued: individual mechanisms

System	Horizon	Modality	Property / decision	Inputs to the check	Intervention and scope
SafeClaw-R (Wang et al., 2026c)	T	S+E	Safety nodes apply workflow rules and skills.	Workflow context, rules, skills and code.	Node admission and configured safety actions under an orchestrator; REVIEW needs a disposition.
Sentinel (Wu & Gong, 2026b)	T	S+F	Policy invariants over speculative graph mutations and taint.	World graph, session history and proposed call.	Allow, block or clarification decision at the modeled action boundary.
Distributed Sentinel (Wu & Gong, 2026a)	T	S+F	Local checks with Semantic Taint Tokens and remote predicate queries.	Local graphs, grounded entities, tokens and remote answers.	Sidecar/gateway admission across organizational workflows under trusted enforcement components.

Table 7: Evidence and trust matrix for the same systems. Evidence credits the inspected source's contribution; the final column bounds transfer, not usefulness.

System	Evidence and evaluated setting	Trusted premises	Transfer boundary / required evidence
NeMo Guardrails	Dialogue, moderation, factuality and hallucination evaluations; executable flow examples.	Flow coverage, semantic mapping, checks and runtime integration.	Dialogue/rail scores do not certify arbitrary tool effects or cross-authority policy composition.
TrustAgent	70 cases across five domains; GPT-4-emulated tools, safety/helpfulness judgments and trajectory metrics.	Regulation retrieval, inspector, simulator and revision loop.	Emulated planning results do not validate mediation of all real tool execution paths.
AgentSpec	Runtime architecture and code, embodied and driving evaluations.	Predicate fidelity and event interception.	Evidence must identify which execution routes and configured enforcement actions actually prevent effects.
GuardAgent	Generated checks and label evaluation on EICU-AC and Mind2Web-SC.	Faithful policy translation, toolbox and integration timing.	Runnable checks and label accuracy do not establish complete mediation.
Progent	Confinement analysis; AgentDojo/ASB and shared-user multi-agent integration. Main approval setting auto-approves.	Initial policy, approver and mediated routes.	Confinement between approvals does not certify policy intent or independent-authority translation.
RvLLM	Domain reasoning evaluations, including legal and mathematical tasks.	Rule completeness and faithful perception/encoding, acknowledged by the source.	No detected contradiction is not unrestricted correctness or tool-call enforcement.
AgentArmor	AgentDojo, observation-injection ASB and a transfer-execution policy case.	Inferred graph, registry, tool semantics and interception.	Deterministic propagation does not remove dependency-inference uncertainty; permission tradeoffs remain.
CaMeL	Architectural security argument and AgentDojo protection/utility evaluation.	Control extraction, interpreter, capability metadata and policy.	A receiving runtime must preserve compatible checks; authorized disclosure need not be correct task completion.
Fides	IFC architecture and AgentDojo evaluation of protection and utility.	Trusted labeling, propagation, policies and wrappers.	Additional boundaries must preserve metadata and enforcement; prompt-level tag omission does not refute native IFC.

Continued on next page

Table 7 continued: evidence and trust

System	Evidence and evaluated setting	Trusted premises	Transfer boundary / required evidence
Permissive IFA	Conditional search analysis and label/output-alignment evaluations.	Correct input labels, restricted generation; monotonic utility for search completeness.	The returned artifact is new; its label is not a dependency certificate for the original output.
IPIGuard	AgentDojo experiments and ablations of planned constraints.	Trusted planning inputs, argument inference and Query/Command classification.	Query-only expansion does not itself establish confidentiality or recipient authorization.
DRIFT	AgentDojo, ASB, AgentDyn and ablations; adaptive tests target the Injection Isolator.	Planner, revision/privilege judgments, masking and interception.	Approval-as-rejection does not measure real user approval outcomes; learned revision is not checked permission non-expansion.
Task Shield	AgentDojo attack-resistance and task-utility evaluation.	Instruction extraction, contribution judgments and integration.	Objective contribution does not establish an owner's release permission.
MELON	AgentDojo single-prompt masking variants; separate ensemble-bound analysis.	Masking, similarity test and statistical prerequisites.	Single-prompt tests do not validate an ensemble; bound prerequisites need explicit justification.
ToolSafe	TS-Bench guard judgments; separate online blocking/feedback evaluations.	Guard reliability, call interception and feedback uptake.	Static accuracy, prevented effects and successful recovery are distinct; controversial scoring is not gate semantics.
IsolateGPT	Protection, functionality and overhead; permission-dialog appearance and warning rates.	Host, hub, trusted metadata and user permissions.	Warning quality is not observed user decision quality; within-app compromise is out of scope.
ACE	Planning and execution attack studies; injection call scores and separate Tool Usage final-output/state correctness.	Trusted planning inputs, labels, isolation and orchestrator.	Metadata attacks relax IsolateGPT's assumptions; output-based execution attacks also use its weaker threat model. Neither proves general authorization.
AgenTEE	OpenCCA two-VM prototype and preliminary agent/model latency.	CCA hardware/firmware, cryptography and input delivery.	Asset protection and latency are not evaluations of semantic tool-policy compliance.
Authenticated delegation	Architectural proposal and worked examples.	Issuers, keys, scope translation and receiving services.	Authentic grants need faithful interpretation and enforcement; translation validation remains open.
APC	Conditional proofs, chain/static tests, injected calls, 23 self-designed white-box adaptive attacks and utility runs.	Faithful action classes, complete restrictions, serialized admission and cumulative accounting.	The session contract does not automatically cover parameter semantics, cross-session effects or unrestricted compromise.
DWT broker	Demonstrator tests; scripted LangGraph nodes without live models; CrewAI/AutoGen code and MCP analysis.	Broker/keys, authenticated workload identity, action mapping and mediation.	These are different evidence types, not four live-agent evaluations; artifact was unreleased in the inspected source.
ProbGuard	Conditional PAC analysis; embodied/driving studies; 30 training traces per driving scenario.	Markov abstraction, estimation and sampling assumptions.	Source acknowledges sub-certification driving budgets and abstraction error; not software-agent validation.

Continued on next page

Table 7 continued: evidence and trust

System	Evidence and evaluated setting	Trusted premises	Transfer boundary / required evidence
SafeClaw-R	About 2,000 Workspace cases with negative controls; separate skill/code studies and operational examples.	Orchestrator mediation and faithful safety-skill execution.	Third-party skills overuse REVIEW and lack benign controls; advanced semantic evasion is untested in the code study.
Sentinel	Conditional analysis, reviewed labels and post-hoc/simulated-online agreement.	Current state, faithful mutations and predicates.	Fixed-call agreement does not measure recovery; source frames findings as structural, not incidence estimates.
Distributed Sentinel	Conditional composition analysis and constructed cross-agent enforcement studies.	Trusted sidecars/keys, grounding, propagation and available violation witnesses.	Compromised agents are excluded; monotonicity alone does not supply missing witnesses. Same authors as Sentinel.

APC uses injected attack calls and a self-designed 23-attack white-box suite (Section 6.5), not unrestricted adaptive compromise; residual attacks and utility costs are reported (Muruaga, 2026). DWT executes scripted LangGraph nodes without live models, inspects CrewAI/AutoGen code, and analyzes MCP's specification (Dantuluri & Sundi, 2026). These are not four live-model experiments; broker tests are separate from its production description.

D Coverage Bridges and Additional Evaluation Settings

D.1 Security Foundations and Adjacent Threats

Enforcement power is an explicit modeling choice. The action gate in Section 3 uses the execution-monitoring tradition (Schneider, 2000). Edit automata can also suppress and insert actions (Ligatti et al., 2005); this explains why stopping, filtering, repair and regeneration should not be collapsed into one intervention label. We do not infer that an LLM repair loop satisfies an edit-automaton theorem. Hyperproperties formalize requirements over sets of traces (Clarkson & Schneider, 2010), sharpening the distinction between trace safety and noninterference already used in IFC (Sabelfeld & Myers, 2003). The comparison records therefore identify the actual protected property instead of treating every safety claim as Equation 2.

Delegation requires both credentials and interpretation. Macaroons support decentralized delegation through context-restricting caveats (Birgisson et al., 2014). This is prior authorization machinery, not a new principle created by agent tokens. Agent-specific contributions must identify how actions, policy context and enforcement routes are bound to that machinery. Our handoff questions concern preservation of these meanings, not a claim that authenticated delegation is absent from prior security research.

Content classification, injection and persistent memory. Llama Guard classifies conversation prompts/responses under a content-risk taxonomy (Inan et al., 2023). Greshake et al. demonstrate indirect injection through retrieved content in LLM-integrated applications (Greshake et al., 2023). MINJA adds query-only poisoning of later-retrieved agent memories (Dong et al., 2025); its threat model assumes shared memory or feasible identity disguise for an isolated bank (Section 3). Our inference is that a tool-return boundary is not the only place where untrusted content can enter future decisions. These sources motivate observation and provenance coverage; they are not additional blocking executors or proof of attacks against every system in the matrices.

D.2 Relationship to Existing Syntheses

Table 8: Complementary analytical perspectives. This is a positioning map, not a checklist of topics supposedly absent from earlier surveys.

Source	Organizing perspective	Connection to this paper
Hou et al. (2025)	MCP architecture, server lifecycle, threat surfaces and mitigations; inspected v3, Sections 3–6.	Protocol lifecycle contextualizes where runtime premises can change; not a current-version conformance audit.
Dantas et al. (2026)	Specification, verification and enforcement, including composition challenges.	We compare concrete observation inputs, intervention authority and evidence transfer.
Lotfi et al. (2026)	Trajectory assurance and cross-agent delegation.	We credit the action-sequence perspective and map it to implementation-facing records.
Siu et al. (2026)	Context-dependent security and ideal oracles; composition remains open.	We build on the contextual distinction, then compare concrete mechanisms and their trust premises.

D.3 Additional Evaluation Evidence

The following six settings extend Table 2. Their roles differ even when they use similar tool APIs. ASB means *Agent Security Bench*, not *Agent-SafetyBench*; their constructions and scores must not be interchanged. The inspected versions are ToolEmu 2309.15817v2 (Sections 3–5, 7), InjecAgent's ACL 2024 final paper (Sections 2.3, 8), ASB 2410.02644v4 (Table 4, Appendix C.2.3), Agent-SafetyBench 2412.14470v2 (Sections 3–4), OS-Harm 2506.14866v1 (Sections 3–4), and ConVerse's EACL 2026 final paper (Sections 3–6, 8–9). Source-reported human annotation or judge validation is not independent replication by this paper.

Table 9: Additional evaluations: execution substrate, measured outcome and interpretation boundary. None of the columns is a universal quality score.

Evaluation	Interaction / substrate	What is measured	What requires additional evidence
ToolEmu (Ruan et al., 2024)	LM-emulated tool responses and state; 36 toolkits and 144 curated cases.	Trajectory safety/helpfulness judged by LMs; human validation of emulation and evaluation.	Emulated outcomes require fidelity checks before claims about live effects; human validation is not universal equivalence.
InjecAgent (Zhan et al., 2024)	Starts after an assumed successful user-tool call; 1,054 paired cases per prompting setting.	Harmful next-tool selection; data stealing additionally simulates extraction and tests subsequent transmission.	Conditional continuations are not complete task episodes or unrestricted adaptive attacks.
Agent Security Bench (Zhang et al., 2025a)	Scenario-specific agents/tools with prompt, tool, memory and backdoor attack surfaces.	Attack-tool invocation and required-tool task success; seven metrics in inspected v4.	Tool invocation criteria do not by themselves establish arbitrary argument semantics or correct external effects.
Agent-SafetyBench (Zhang et al., 2024)	349 interaction environments and 2,000 cases; eight risk categories and ten failure modes.	Behavior/content safety from interaction records, using a fine-tuned scorer and source-side human labeling.	A learned safety score is not a deterministic authorization oracle or proof of complete mediation.

Continued on next page

Table 9 continued: additional evaluations

Evaluation	Interaction / substrate	What is measured	What requires additional evidence
OS-Harm (Kuntz et al., 2025)	Computer-use tasks in OSWorld-based environments; misuse, injection and model misbehavior.	LLM semantic judgments of task success and safety over executed computer-use traces.	Judge validity and tested attack coverage bound conclusions; no general cross-agent enforcement guarantee follows.
ConVerse (Gomaa et al., 2026)	Multi-turn assistant/external-agent conversations with simulated data/tools in three domains.	GPT-5 matches privacy/security outcomes to annotations; utility includes plan coverage and option ratings.	Contextual cross-agent risk evaluation, not a proof of enforcement; simulated environments and judge validity bound transfer.

The expanded evidence map changes what can fairly be claimed: cross-agent evaluation is not wholly absent, as ConVerse demonstrates. The remaining question is which evaluated interactions preserve the contracts needed by a particular enforcement composition. Broader benchmarks, conditional tool-call tests and risk classifiers inform that question at different levels; their scores cannot be pooled into a ranking of runtime architectures.

E Suggested Evidence Record and Diagnostic Order

This reporting checklist operationalizes this paper's research questions, not a collected benchmark. Studies should identify supported fields and comparisons rather than imply complete coverage.

Sample independently constructed scenario families, each defining initial state, policy oracle, legitimate objective, proposed actions, and permitted/prohibited variants. Retain family identifiers for splitting and uncertainty estimates; near-duplicates are not independent evidence. Record model identifiers, prompts, tool and policy versions, and supported seeds.

Each decision record contains: scenario and pair identifiers; agent identity; policy version; pre-action state or an immutable reference to it; proposed action and arguments; the exact monitor observation; provenance and its validation status; decision and clarification outcome; whether the gate intercepted the operation; executed effect; resulting state; oracle label; completion status; latency and usage. Store unavailable fields explicitly. An explanation without an observed execution record is not evidence that an action was blocked.

Table 10: Diagnostic order for an unsafe executed action. Multiple failures can coexist; retain secondary causes rather than forcing every case into a single label.

Question	Negative outcome	Positive outcome
Did the required gate intercept this execution?	Execution-control failure	Inspect decision and its inputs
Was authorization decidable from those inputs?	Missing-information limitation	Inspect policy representation
Did the encoded check capture the intended rule?	Specification mismatch	Inspect checker correctness
Did the checker deny the prohibited action?	Decision failure	Inspect post-check state changes and gate implementation

For replayed proposed actions, compute unsafe execution rate over prohibited opportunities and false-denial rate over permitted controls. For full interactive episodes, additionally report any-violation rate and safe completion among permitted tasks. A terminal clarification or timeout is not successful completion. Report how errors affect each denominator, and show the trade-off as a threshold or intervention budget changes. No numerical outcome is presupposed by this protocol.

F Supporting Details for Policy Translation and Handoff Tests

F.1 Artifacts and Scope

The example imports released code without modifying tracked upstream files. The inspected repository pins, not asserted to be the original papers' evaluation commits, are:

CaMeL	`f083b6b396399d3b3c7f2ddaf613a5945eaf32d8`
Fides	`669c046c4adbc56ee49c9672fdd30c54937ea062`

The recorded environment uses CPython 3.12.14, AgentDojo 0.1.34, and Pydantic 2.11.5. Fides declares Python $\geq$3.13 for its full project; we execute selected unmodified tutorial definitions on 3.12, not its complete pipeline or declared environment. No condition makes a model request or contacts an external tool service. The proposed actions are scripted and effects remain in an isolated mailbox.

F.2 Contract and Complete Finite Coverage

The supported action is `send_email(recipients, subject, body)`, without CC, BCC, attachments, or other tools. Three fictional identities define the reader universe. The subject and body are fixed public fixture strings assigned the same reader label. Labels are supplied, not inferred from content or asserted by a model. Admission requires nonempty recipients, each permitted to read both payload fields. Trust alone does not declassify.

The matrix enumerates eight reader subsets plus the native public representation, eight recipient subsets including empty, and two trust labels: $9 \times 8 \times 2 = 144$ conditions per configuration. The all-readers subset and public label agree on this universe but exercise different constructions. They are not independent tasks. An independently expressed set-inclusion oracle supplies expected outcomes only for scoring; neither runtime policy calls it.

CaMeL uses native capability membership, its interpreter, and AgentDojo wrappers with our configured send policy. NORMAL and STRICT are separate configurations. Fides uses native lattice operations and its tutorial policy-before-tool loop. Our bridge replaces the tutorial's printed messaging fixture with the same in-memory AgentDojo send operation. The shared payload label makes its action-level label exact for this restricted contract. These policies and the bridge are ours, not upstream defaults.

All 432 checks are retained. Each configuration produces 52 single-mailbox effects with exactly matching arguments and 92 denials with unchanged complete state. No condition ends in a runtime error. Conditions, decisions, effect payloads, and before/after state hashes are recorded. The offline verifier recomputes the Cartesian coverage, expected labels, decisions, effects, and summary counts. It checks recorded evidence rather than independently attesting that logs are authentic.

Table 11: Complete finite supplied-label matrix: 144 conditions per configuration, 432 checks total. Counts describe metadata/action fixtures, not independent tasks.

Configuration	Permitted effects	Denied, no effect	Conforming checks
CaMeL NORMAL	52	92	144/144
CaMeL STRICT	52	92	144/144
Fides tutorial, configured bridge	52	92	144/144

F.3 Composed Handoff and Revocation

The same pinned components execute 80 further fixtures. The source policy allows a CaMeL send to Alice and records the payload's initial reader set and policy version. Its subject/body are then sent through the Fides tutorial policy-before-tool loop to a requested receiver. Four initial reader sets (all containing Alice), two receivers (Bob or Charlie), and a binary revocation flag give 16 fixtures. On revocation, the requested receiver is removed from the current reader set. The current-policy oracle is separate from the native lattice check.

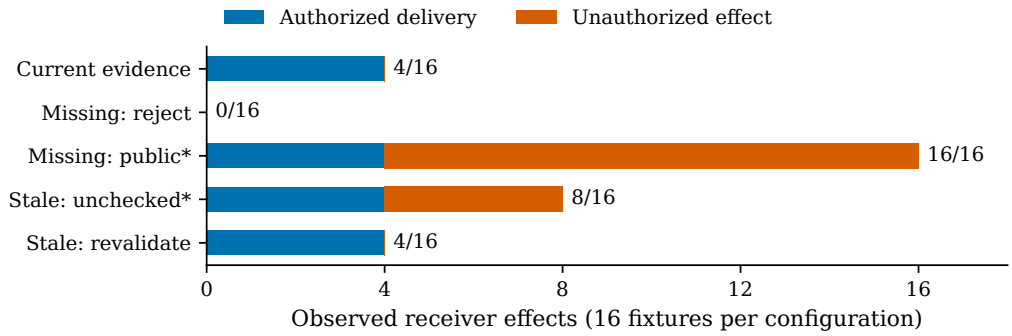

Figure 4: Configured-native handoff illustration, 80 fixtures total. Blue segments are authorized deliveries; hatched segments are unauthorized effects. Each arm has four currently authorized requests among 16 fixtures. Asterisks mark our permissive negative-control bridges. The planner is scripted and makes no recovery attempt.

The five configurations are: (i) preserve refreshed evidence, (ii) reject missing evidence at the bridge, (iii) convert missing evidence to the native public label, (iv) reuse the original label without refreshing after revocation, and (v) revalidate against current readers. Configurations (iii) and (iv) are our negative controls, not upstream defaults. Configuration (v) assumes the configured bridge can obtain authoritative current reader information; the experiment does not implement a distributed authority service. All arms bind the actual source payload to the receiver arguments, execute the same native send operation, and record complete before/after state hashes.

Table 12: Native handoff fixture counts. "Permitted denied" is a false-denial count among four currently authorized receiver calls per arm, not an on-policy autonomous false-denial rate.

Bridge	n	Unsafe	Delivered	Authorized	Permitted denied
Current evidence	16	0	4	4	0
Missing, reject	16	0	0	0	4
Missing, public (control)	16	12	16	4	0
Stale, unchecked (control)	16	4	8	4	0
Stale, revalidated	16	0	4	4	0

All allocated fixtures are retained and none ends in a runtime error. The five arms respectively block 12, 16, 0, 8 and 12 receiver calls. No scripted recovery step follows a denial, so post-block completion is zero by construction where defined, and undefined for the no-block arm; this is not a recovery comparison. Denominators are finite fixtures, not sampled tasks, so no population confidence interval is attached. The companion includes the native runner, frozen 80-row archive, source pins, metric implementation and integrity/metric checks.

Executable regression oracle. The companion's `handoff/expected_outcomes.json` pairs every fixture with expected arguments, a source-body hash, a decision, delivery, current authorization and unsafe effects. These expectations were exported retrospectively from the five declared bridge policies, without reading recorded outcomes; they are not a preregistered test set. The dependency-free `handoff_oracle.py` recomputes them without importing either runtime. It checks all 80 outcomes and binds receiver arguments to fixed fixture payloads and the recorded source-body hash, including denied calls. Delivered payloads must match; denied calls must leave state unchanged. Tests reject joint argument/effect mutations, changed source hashes or expectations, and missing/duplicate fixtures. A replacement bridge can run the same fixtures against the declared contract; these policy-specific expectations do not rank different security policies.

F.4 Interpretation and Reproduction

The measured claim is supplied-label policy-to-effect consistency over the complete declared finite space. Labels and payloads are fixed inputs; source-to-label ingestion, autonomous planning and mutable authorization are separate components of the deployment contract. Section 7 states the corresponding generalization boundary.

The companion contains the complete condition file, recorded package versions, runner and loader snapshots, contract, development protocol, and offline analyzer. Condition bytes are unchanged. The public export limits repository-pin metadata to the two runtimes needed here; its manifest records the original hashes and the changed metadata hashes separately. No outcome or fixture is removed. Native reruns require the pinned external checkouts; upstream source is not redistributed.

G Distinguishing Evidence and Composition Failures

For local trace policies P_i, including projections, specification non-composition means

$$\left(\bigwedge_i P_i(\tau)\right) \not\Rightarrow \Phi(\tau). \tag{3}$$

Here local policies hold but do not entail the global property, unlike checker or executor errors. Read and summarize permissions need not imply release permission. An existing release rule with unobservable grants instead presents an evidence deficit.

Observation 1 (Indistinguishable authorization states)**.** *Suppose two reachable decision contexts x and x' have the same monitor observation, $O(x) = O(x')$, but opposite correct binary authorization decisions. No deterministic monitor using only O can decide both correctly.*

Proof. Identical inputs produce identical decisions, so one context is misclassified. □

This elementary observation is not new (Siu et al., 2026; Wu & Gong, 2026a). Randomization cannot eliminate both errors; always blocking loses utility. Clarification needs distinguishing evidence. This is not an empirical failure of any compared mechanism.

H SEARCH PROTOCOL AND EXTENDED EVIDENCE TABLES

This paper uses purposive evidence synthesis and targeted gap checks, rather than an exhaustive systematic-review design. This extension compares 43 selected, source-anchored records: 19 mechanisms or components, nine evaluations and 15 syntheses. It is a curated analytical subset, not the complete eligible pool or 43 independently validated executors. The core comparison retains its 25-record denominator. Together with the nine separately logged named-source records, the companion dataset contains 77 citation-keyed comparison records; this is not the total bibliography or search yield. Source claims, review interpretations and process provenance are separate fields. The AI Use Statement describes assistance, Appendix B.1 reports independent classification coding, and Section 7 consolidates the remaining review scope.

H.1 IDENTIFICATION AND ELIGIBILITY

The initial protocol fixed eligibility, extraction fields and queries before the bounded topic-search wave on September 21, 2026. The primary window is 2022 through September 21, 2026; older foundations are contextual. Public research preprints and peer-reviewed papers are eligible. Publication status is recorded, not used as a quality threshold. No language exclusion was imposed. Later acquisition dates do not move the literature cutoff. Inspected versions, rather than presumed latest versions, support the claims below.

Eligibility and extraction. M denotes a software-agent runtime mechanism or component with an extractable observation/intervention interface and trust model. Proposals are permitted but distinguished from experimentally evaluated implementations. E denotes action/trajectory evaluation of safety, security, authorization, privacy or relevant execution failures, including benign failures. S denotes an explicit synthesis of runtime safety, action enforcement, assurance or authority. General architecture, content-only moderation, attack background and embodied boundary cases may be retained as context, outside the primary M/E/S yield.

Title and available primary-abstract screening precede targeted reading of methods, threat models, evaluation and limitations. Eligibility requires inspectable primary text with version and section anchors; abstract-only access does not establish it. Extraction records protected properties, observations, interventions, trusted assumptions, evaluation unit/judge, benign utility, adaptivity and limitations. Weak empirical evidence is coded as a limitation, not suppressed by a quality-score threshold. Reused source audits are counted separately from new targeted readings, neither of which is a replication. DOIs, arXiv identifiers and normalized titles support initial grouping; cross-version merges require primary-source identity evidence. An acronym match alone is insufficient. Counts remain provisional study groups rather than independently verified unique studies.

Search design and diagnostic. Five initial query strings were each issued to three domain groups (15 calls), followed by eight prospectively recorded amended queries. These were domain-filtered web-index searches, not native database searches. Complete tool-returned windows were retained, but backend hit totals, stable ranking and pagination were unavailable. The final two-wave search recovered seven of eight predefined known-paper probes. Because probe feedback informed the amendment, this is a development diagnostic, not an independent recall estimate. The companion preserves both query versions, their per-wave probe outcomes and the unmatched identity. Named-paper recovery and version lookups are separate routes and do not inflate the topic-search yield.

Table 13: Exact bounded-search strings. Each v1 row was executed once per domain group; each v2 row once over the pooled group U.

Query / filter	Search string
v1-1 (X/C/P)	LLM agents runtime safety enforcement tool calls
v1-2 (X/C/P)	LLM agents information flow isolation prompt injection defense
v1-3 (X/C/P)	LLM agents authorization delegation access control policy
v1-4 (X/C/P)	LLM agents safety security benchmark evaluation

Query / filter	Search string
v1-5 (X/C/P)	LLM agent safety security survey runtime
V2-1 (U)	"LLM" "runtime enforcement"
V2-2 (U)	"agents" "information-flow control"
V2-3 (U)	"agents" "privilege control"
V2-4 (U)	"LLM" "execution isolation"
V2-5 (U)	"LLM agents" safety regulations planning inspection
V2-6 (U)	"LM agents" "emulated sandbox"
V2-7 (U)	"prompt injections" "design" capabilities agents
V2-8 (U)	"agents" "tool" safety benchmark guardrails

Domain filters: X: arxiv.org. C: aclanthology.org, openreview.net, proceedings.mlr.press, proceedings.neurips.cc. P: usenix.org, ndss-symposium.org, dl.acm.org, ieeexplore.ieee.org. U: arxiv.org, aclanthology.org, openreview.net, proceedings.mlr.press, usenix.org, ndss-symposium.org, dl.acm.org.

H.2 Direct Database Supplement

The supplementary database route queried the Semantic Scholar bulk-search and OpenAlex works APIs on September 24, 2026 (UTC), retaining the 2022–September 21, 2026 publication window. Semantic Scholar searched titles and abstracts; OpenAlex also searches indexed full text. Boolean syntax was adapted to each endpoint. This is a separate route from the web-index search. The companion's `database_supplement/` directory preserves the protocol, exact requests, response hashes, pagination limits and stop reasons.

Table 14: Direct Semantic Scholar queries and returned occurrences. All six completed at the endpoint. Symbols + and | denote AND and OR. Counts precede identity grouping.

ID	Query	Occurrences
D01	(LLM | "language model") +agent +runtime +(safety | security)	325
D02	"prompt injection" +agent +(defense | defence | guard)	338
D03	"information flow" +(LLM | "language model")	474
D04	(LLM | "language model") +agent +(privilege | authorization | delegation)	484
D05	("LLM agent" | "language model agent") +(monitor | guardrail | enforcement | shielding)	618
D06	("LLM agent" | "language model agent") +(benchmark | evaluation) +(security | safety | privacy)	809

Semantic Scholar returned 3,048 occurrences; OpenAlex added 100 from D05. Rate limits prevented further OpenAlex retrieval, so that route is partial. Across both APIs, 3148 occurrences form 2546 provisional identity groups using exact provider IDs, normalized DOI/arXiv IDs and normalized titles. Of these, 83 link provisionally to the original candidate register and 2463 form a separate discovery pool, outside the assessed evidence. The database route provisionally matches 17 of 25 core and 20 of 43 extension records. It recovers three of the eight original probes, including AgentSpec; together with the earlier broad-query route, all eight probes are recovered. This is a known-set diagnostic after query development, not an independent recall estimate.

Targeted assessment and discovery reserve. The author completed selection for six targeted sources: five inclusions and one scope exclusion (Appendix H.3). AIOracles uses its dated arXiv v1. Three decisions concern database candidates; four concern the citation route, with one source shared between routes. The five inclusions are reported separately from the historical 136-record base yield and frozen 77-record comparison, whose denominators remain unchanged.

For the other 2460 database records, the author confirmed human verification of AI-assisted title/abstract dispositions and 42 later identity/abstract rechecks affecting 49 records. The effective

initial dispositions are 1322 advance, 665 exclude, 460 context and 13 information-pending. These are initial dispositions. The 1335 active records form 1330 reading units after confirmed version links: 1317 advance and 13 information-pending units.

On September 25, after exploratory screening, we retained the supplementary database and citation pools as discovery reserves for targeted gap checks, including potential counterexamples. Comparison analysis uses source-assessed records; reserve candidates without full-text assessment remain in the retrieval log rather than the included corpus or the scope-exclusion count. This retrospective scope amendment preserves the literature cutoff and M/E/S criteria. The companion's `scoped_search/` directory preserves the current initial dispositions, identity links and amendment; its `selection_followup/`, `selection_recovery/` and `selection_residual/` directories retain source-linked decisions, quality corrections and acquisition history. Table 15 reports the assessed base route separately from these discovery reserves.

H.3 Author-Reviewed Follow-up Sources

An author reviewed six targeted candidates against the primary sources after consulting AI-prepared reading notes: three were included as M, two as E, and one excluded as X1. These are source-informed author decisions, not independent blind screening. The five inclusions supply supporting evidence below; they are outside the frozen 77-record comparison and the 68-record coding exercise. The companion's `author_selection/decisions.json` preserves all six decisions, source versions, anchors and author rationales. Historical AI recommendations remain separate. The original 136-record base-route yield is unchanged; it is not a count of these follow-up decisions.

Runtime architecture and audit components (M). Domkundwar et al. (2024), arXiv v3, February 28, 2025 (Sections IV–VII, Figure 5, Table I; PDF pp. 2–6), describe input/output filtering, a safety agent and hierarchical consultation. Their 21-prompt response-safety evaluation is distinct from an external-effect or benign-completion denominator. Tan et al. (2026), arXiv v2, June 22, 2026 (Sections 2–5, Tables 1–8; PDF pp. 2–10), describe SandScope under the MCP-SandboxScan title. Its runtime audit separates source-to-sink witnesses from declared capabilities and egress evidence; WASI and native stdio backends have different containment assumptions. These records broaden the observation and orchestration interfaces without being treated as complete action gates.

Evaluation evidence (E). ToolAlign (Chen et al., 2024) (Sections 2.3 and 3.1–3.2, Table 2; PDF pp. 4–6 and 9) contributes tool-use safety evaluation despite its training-based intervention. Response refusal is kept separate from prevented unsafe effects. WebInject (Wang et al., 2025c) (Sections 3–5.1, Equation 5, Table 1; PDF pp. 3–7) measures exact target-action output. This extends the action-selection evidence, not an executed-harm or legitimate-completion rate. Neither inclusion changes the frozen 11-setting coverage figure.

Positive composition evidence (M). The dated AIOracles predecessor (Villa et al., 2026), arXiv v1, March 3, 2026 (Section 6.5, Theorem 1, Figure 8 and Appendix E.2), supplies a conditional generator/filter composition result. Its premises concern component completeness and security; the appendix gives the corresponding reduction. We cite v1 rather than the undated Microsoft-hosted revision. The latter's theorem numbering, seven-subpredicate decomposition and multi-step reduction are not attributed to v1. These are versioned HTML section anchors, not PDF page locators or a new independent proof verification.

Reasoned scope exclusion (X1). *LLMs Information Flow Diagnostic: Memory-based Evidence from Random Matrix Theory*, ICNLSP 2025 (Abstract and Section 1; PDF pp. 1–2), studies neural layer/weight dynamics rather than software-agent runtime information-flow control. The author excluded it for scope, not quality. Its primary identifier is `2025.icnlsp-1.12`; the companion retains the source link and decision.

H.4 Reconciled Selection Flow

The base route comprises 100 legacy bibliography occurrences and 315 web occurrences. Importing an existing citation does not verify it. After 119 identity links, 296 provisional groups remain. Table 15 separates dispositions from acquisition failures and context records.

Stage / disposition	Count
Legacy bibliography occurrences	100
Web-index occurrences (23 calls)	315
All base-route occurrences	415
Identity links (not exclusions)	119
Provisional candidate groups	296
Eligible: reused section-level audits	39
Eligible: new targeted source readings	97
Context only	76
Content/scope exclusions	77
Full text not retrieved; unassessed	5
Identity unresolved; unassessed	2

Table 15: Base-route accounting. The last six rows partition 296 groups; eligible roles are M=61, E=48, S=27 (136 total). Unassessed records are not content exclusions.

The 77 content/scope exclusions comprise eight records without a software-agent connection (X1), five training/content-only records without a runtime/action connection (X2), ten capability-only records without relevant safety/failure evaluation (X3), and 54 non-research pages (X4). No outside-window (X5) or post-full-text insufficient-detail (X6) exclusion was assigned. Exclusions use the recorded source depth and are not all full-text exclusions. Context records are a separate background pool, not failed studies or promised citations.

Five full texts could not be obtained and two fragmentary identities could not be resolved. They are outside assessed evidence, with retrieval history retained; acquisition follow-up was closed without a content decision. The 87-record residual queue therefore ends with 37 eligible, 21 context, 11 content/scope exclusions, 11 verified duplicates and seven unassessed closures. This is 80 AI dispositions and seven unassessed records, not 87 full-text reviews. Unavailability can bias coverage; it does not establish irrelevance.

A separate backward/forward citation export from three predefined synthesis seeds yielded 771 occurrences, grouped provisionally into 756 candidates. At export, there were 36 tentative matches to the base route and 720 awaiting primary-abstract screening. These exports did not supply additional eligible records and are not merged into Table 15. The citation export is retained as a discovery reserve for question-driven follow-up; its historical counts do not establish completed snowballing or citation saturation. The later six cross-route decisions are reported separately in Appendix H.3.

Analytical subset and limitations. The 43 records below extend comparison of runtime interfaces, authority and evaluation denominators. Selection is purposive within the eligible pool; there is no statistical sampling claim, performance-based threshold or claim that every eligible source is tabulated. The subset excludes all seven unassessed records and a course manuscript retained in the register. OpenCUA is used only for benign execution-failure analysis, not security validation. The Harness Engineering source uses a pinned author-repository revision dated July 21, 2026; repository metadata is not a peer-reviewed publication date. Source-specific review flags and earlier decisions remain in the companion register. Classification agreement for the 68 core/extension records is reported in Appendix B.1; eligibility decisions retain their own provenance in the selection register. Section 7 states the coverage and review boundaries.

Editorial allocation, recorded retrospectively. The eligibility protocol determines the 136-record pool, not membership in each comparison table. The companion now records a rationale for every candidate's current editorial allocation, dated September 23, 2026. These are retrospective explanations, not prospectively fixed selection criteria. The 25 core records retain interfaces around which the comparison was developed; 23 belong to the primary eligible pool, while RvLLM and ProbGuard are contextual boundary cases. The 43 extension records broaden interfaces, evaluation settings, and comparison with neighboring syntheses. Twelve original evaluations and four related syntheses retain a separate supporting-comparison role. Four further eligible records challenge broad interpretations below; the remaining 50 eligible records have no comparison-table allocation: two are cited as attack background and 48 remain register-only. They are not deemed irrelevant, redundant, or lower quality. Thus $23 + 43 + 16 + 4 + 50 = 136$; the nine named-source additions remain a separate

route. This makes discretionary selection visible without establishing that it is representative. The anonymous export includes all 296 current candidates, 415 occurrence identities, 119 identity links, query protocols, and historical decision/recovery files. Source texts and raw search snippets are not redistributed.

H.5 Retrospective Scope Challenge

Four already-eligible records outside the 77-record comparison set were selected on September 23 to challenge broad readings of the synthesis. They add no search hits and constitute neither a random holdout nor independent validation. Their source anchors and current allocation are in the companion register. These examples narrow the claims as follows.

Table 16: Scope challenges from the eligible pool. The last column is our interpretation, not a source-authored criticism.

Source	Source contribution	Consequence for this paper
SAGA (Syros et al., 2026)	Contact policies and receiving-agent tokens govern inter-agent communication, with protocol and empirical evidence.	Cross-agent authorization is already concrete prior work. Our remaining question is which policy and effect semantics survive a handoff, not whether cross-agent enforcement exists.
ToolGate (Liu et al., 2026)	Preconditions gate tool invocation; postconditions regulate returned-result admission to symbolic state.	A postcondition check controls state admission; rollback of external effects requires separate evidence. Distinguish these intervention units rather than calling both pre-execution prevention.
Privacy-R1 (Hui et al., 2026)	An RL-trained policy routes query chunks between local and remote models to trade privacy against response quality.	Learning can implement a runtime disclosure decision, not merely training-time alignment. Its stated tradeoff permits some task-critical PII disclosure; this is not zero-disclosure enforcement.
The Verifier Tax (Sah et al., 2026)	Measures joint safe success and recovery after intervention on tool-using tasks.	Joint success accounting and recovery are prior work. Our incremental contribution is locating different observation, selection and executor obligations in a shared reporting record, not inventing a safety-utility metric.

H.6 Consequences for the Synthesis

Different checks protect different relations. ToolGuards connects policy interpretation to executable checks, but deterministic execution does not establish that generated policies or checkers are correct (Zwerdling et al., 2025). AttriGuard tests whether calls survive attenuated-observation replay; this criterion does not authenticate delegated content (He et al., 2026). Flow-centric systems regulate represented dependencies: AgentFlow constrains modeled flows, FlowSeal combines recipient authorization with semantic declassification, and SPA preserves labels across query boundaries (Ammanaghatta Shivakumar et al., 2026; Shim et al., 2026; Girrens & Wang, 2026). CommandSans instead sanitizes observations before they enter agent context (Das et al., 2025). These interfaces address different relations, not interchangeable implementations of one safety predicate.

Integration remains part of the claim. AgenTRIM combines tool exposure with call assessment, while skill-oriented VIGIL and PRISM describe hooks with different evaluated environments (Betser et al., 2026; Li et al., 2026c; Li, 2026). Runtime Governance and the relational authorization overlay make organizational state and delegation explicit; the latter's schema-level soundness assumes typed roots, compatible scopes and trusted relation updates (Kaptein et al., 2026; Ibrahim & Li, 2026). Our inference is that specifying an interceptor does not establish faithful mediation of every delegated or external execution path.

Useful protection needs a stated denominator. AgentDAM separates final-state completion from action-level privacy judgments; CI-Work measures sensitive and essential entry conveyance in simulated workflows (Zharmagambetov et al., 2025; Fu et al., 2026). RealSafe rewards identification and refusal on malicious tasks, whereas LPS-Bench permits confirmation pauses in its benign-context safety criterion (Ma, 2025; Chen et al., 2026b). These outcomes cannot be substituted for one another

in a common utility curve. The comparison below also retains configuration changes, evaluation-set reuse, and retrospective versus preventive checks.

Overlap is evidence, not a novelty score. Earlier threat, harness and cybersecurity-agent reviews cover execution controls, cooperating agents and containment (Deng et al., 2024; He et al., 2025; Wang et al., 2025f; Meng et al., 2026; Addis et al., 2026). The proposed contribution is the auditable comparison of concrete interfaces and evidence boundaries. Additional labels or more citations alone would not establish a new analytical contribution.

Newly integrated evidence sharpens the same distinctions. ControlValve combines graph/parser constraints with learned contextual rules; its acknowledged graph/rule errors prevent reading the graph check alone as an end-to-end guarantee (Jha et al., 2025). The Minimizer/Sanitizer firewall acts at different sides of the tool boundary, and the paper itself demonstrates a bypass despite strong benchmark results (Bhagwatkar et al., 2025). These are concrete examples of why the observed interface and evaluation configuration belong in an enforcement claim.

IntentGuard's robust execution rate credits attack detection as well as completion; Prudentia's benign evaluation assumes approval of endorsement requests (Cong et al., 2026; Kolluri et al., 2026). Neither is an observed human-approval study. The browser same-origin analysis concerns historical January–February 2026 product versions, while OpenCUA's error analysis concerns benign execution and recovery (Roesner & Kohlbrenner, 2026; Wang et al., 2025d). Neither supplies a general protection rate for current deployments.

The methodology survey already discusses tool-execution and inter-agent defenses, and Harness Engineering directly covers governance, delegation and policy composition (Luo et al., 2025; Li et al., 2026b). These overlaps narrow our contribution to source-level comparison of observations, controlled events, assumptions and supported conclusions, not a claim that earlier surveys overlook runtime safety.

H.7 Expanded Comparisons

Entries describe inspected interfaces and evidence. Limits are review interpretations unless attributed to the source. Absence of deployment evidence is not proof of incapability. U-prefixed identifiers link to the version notes below; repeated table rows are not distinct studies.

Table 17: Related syntheses and the focus of this comparison

Source	Substantive overlap	Comparison focus or evidence boundary
Authorization Architectures [U01] (Surapani et al., 2026)	Principal chains, invocation-time enforcement, delegation attenuation, temporal validity, audit and separate coverage/evidence ratings.	Relate principal chains to the observations, controlled event and validity retained at a handoff.
Toward Secure LLM Agents [U02] (Ling et al., 2026)	Runtime defense authority, trust assumptions, recovery and security-utility-cost evaluation.	Attach claimed protection to particular interfaces, trusted inputs and evaluation conditions.
SoK: Agent attack and defense [U03] (Kim et al., 2026)	Information flow, privilege separation, IAM/delegation and complete mediation.	Compare concrete mediation paths and assumptions within the broader attack/defense landscape.
Design Patterns for Securing LLM Agents [U04] (Beurer-Kellner et al., 2025)	Six architectural patterns and case studies, including fixed control flow versus parameter/output manipulation.	Use architectural distinctions as prior foundations, then inspect system-level enforcement evidence.
Runtime Contract [U05] (Ng et al., 2026)	Preventive checks, evidence-gated completion and conditional composition of contracts.	Distinguish conditional contract composition from evidence of integrated enforcement.

Source	Substantive overlap	Comparison focus or evidence boundary
JARVIS or Ultron? [U06] (Chen et al., 2026a)	Computer-use-agent defenses, monitoring, environment constraints, auditing and evaluation metrics.	Connect computer-use defenses to the observations and effects each setting actually exposes.
Evaluation of LLM-based Agents [U07] (Yehudai et al., 2026)	Policy-violating task success, trajectory diagnosis, harness/model separation and cost.	Trace evaluation-unit distinctions to specific denominators and intervention interfaces.
Full-stack safety survey [U08] (Wang et al., 2025a)	Tool, memory, multi-agent and protocol defenses plus safety-utility evaluation.	Focus within the broader lifecycle on runtime authority and the limits of evidence transfer.
Agents Under Threat [U25] (Deng et al., 2024)	Threat sources, models and target effects organized by four knowledge gaps.	Broader than runtime enforcement. Gives prior discussion of auditing, permissions and multi-agent dependencies; do not claim those topics are absent from earlier surveys.
Agent Harness [U26] (Meng et al., 2026)	Six-component taxonomy and tool-registry design patterns.	Already discusses permission composition. Causal-sounding harness scaling claims are the survey's interpretation, not evidence we independently validate.
Emerged Security and Privacy [U27] (He et al., 2025)	Inherited versus agent-specific threats; human/environment/agent impacts.	Already discusses context isolation, oversight and tool evaluation. Its characterization of individual methods is not imported as verified mechanism evidence.
Large Model-Based Agents [U28] (Wang et al., 2025f)	Authentication, denial-of-service, adversarial inputs and privacy taxonomies.	Includes non-agent/model-level and embodied material. Authentication and cross-agent risks are already discussed; our contribution cannot be their first identification.
Cybersecurity Agents [U35] (Addis et al., 2026)	Tool returns, shared memory, credentials and execution boundaries.	Section V-C4 notes offensive/hybrid scope, evolving preprints and heterogeneous evaluation. It does not establish transferable protection rates.
Methodology / Applications [U38] (Luo et al., 2025)	Surveyed models, tools, external data and inter-agent interactions.	Broad architecture survey. Use primary papers for mechanism and empirical claims.
Harness Engineering [U43] (Li et al., 2026b)	ETCLOVG layers and permission/evaluation configuration.	Related-work overlap includes delegation, policy composition and generation/enforcement separation. Different project-page and PDF snapshots are not interchangeable.

Table 18: Additional runtime interfaces: property, observation and intervention

Source / property	Observation	Controlled event or intervention
ShieldAgent [U09] (Chen et al., 2025) Action compliance with extracted government/platform policies.	Pending action, current observation, prior history and policy circuits.	Safety/rule flags and explanations; online post-verification feedback guides the task agent step by step.

Source / property	Observation	Controlled event or intervention
ToolGuards [U10] (Zwerdling et al., 2025) Company policy adherence for invoked tools, including eligibility and required user confirmation.	Tool arguments, conversation history and read-only data APIs queried by generated guards.	Execute a proposed tool only if all mapped guard functions pass; otherwise return a violation explanation for revision.
AttriGuard [U11] (He et al., 2026) User-intent support of tool calls against indirect injection, not general authorization or content authenticity.	User task, proposed calls, full original action history and incrementally attenuated observation history. Fuzzy judge sees task and compared calls, not full history/tool documentation.	Execute original proposed calls only if they survive shadow replay; otherwise return an empty result with an injection warning.
AgenTRIM [U12] (Betser et al., 2026) Per-step least-privilege exposure and tool-call admission.	Inventory/control metadata, candidate calls, history-based triggers and LLM-maintained task status.	Filter exposed tools; reject calls failing approval; continue restricted loops or terminate on assessed completion.
ClawGuard [U13] (Zhao et al., 2026) Rule-constrained tool/path/network access and secret-pattern sanitization.	Arguments, command/path/destination attributes, rules and versioned skill contents.	Sanitize inputs/outputs, block denied calls/skills, pause ambiguous calls for approval, deny on timeout.
VIGIL (skills) [U14] (Li et al., 2026c) Skill prohibitions, preconditions and finite-trace temporal obligations.	Trusted tool-call trace, pending invocation, typed facts, artifact bindings and specifications.	Deny violating pending calls and return witnesses; deployment selects halt/approval/rollback. Future obligations are judged at run end.
Runtime Governance [U15] (Kaptein et al., 2026) Organization-level policy compliance over execution paths, including cross-agent information barriers.	Registered agent identity/metadata, partial path or compressed state, proposed action and shared governance state.	Prospective pass, steer/pause for approval, or block/terminate; retrospective deployment only flags completed events.
PRISM [U16] (Li, 2026) Gateway-visible tool/path/network restrictions, secret-pattern handling and tamper-evident records.	Lifecycle messages/calls/results, normalized path/destination attributes, conversation/session risk state and policy configuration.	Pre-tool blocking, persistence/outbound sanitization, escalation to tool and sub-agent restrictions; audit also operates after events.

Source / property	Observation	Controlled event or intervention
AgentFlow [U17] (Ammanaghatta Shivakumar et al., 2026) Configured confidentiality, integrity, path, task-capability and release constraints over policy-visible mediated flows.	Source/destination edges, selected fields, labels, lineage/history, task scope, capabilities and modeled response sinks.	Allow, deny or pause before tool invocation; allowed calls update labels/lineage/trace. Optional scanners only upgrade labels.
FlowSeal [U18] (Shim et al., 2026) Authorized release of protected records and derived content to modeled recipients.	Protected records/ownership, recipient grants, session taint and proposed outbound content; isolated checker receives records and output.	Mandatory pre-dispatch write/share checks; scoped authorization requests and declassification; blocks return recovery guidance.
SPA [U19] (Girrens & Wang, 2026) Planning/execution integrity, confidentiality and cross-query preservation of information-flow labels.	Trusted queries and artifact metadata for planning; explicit DSL dependencies, tool policies and stored labels for flow checking.	Validate/flow-check complete plans before deterministic execution; fail closed on invalid plans, mappings, schemas or tool failures. Quarantined LLM does not remove taint.
Overlaying Governance [U20] (Ibrahim & Li, 2026) Human-rooted authorization with current delegation, scope/session compatibility and guard validity.	Typed principal/resource graph, authenticated relational facts, active scopes/sessions and contextual predicates.	Controller clients consult a centralized Agent Controller Engine before execution; relation writer maintains authorization state.
AgentTrust [U22] (Yang, 2026) Risk-relative checks on destructive actions, credential access, obfuscation and known multi-step chains.	Typed proposed actions, payload/parameters, session history and optional LLM context.	Allow/warn are executable; block/review stop or escalate. SafeFix supplies suggestions without overriding the verdict.
CommandSans [U29] (Das et al., 2025) Resistance to indirect prompt injections.	Tokens in tool outputs, tagged for AI-directed instructions.	Removes predicted instruction spans before context ingestion; nonblocking sanitizer.
PROMPTGUARD (O-RAN) [U30] (Wang et al., 2026d) Integrity of network-slice allocation against injected control intent.	Telemetry, historical state and contextual summaries; operator-defined admissible intents.	Pre-model semantic classification and thresholding; described sanitization, source isolation and deterministic allocation fallback.

Source / property	Observation	Controlled event or intervention
Minimizer / Sanitizer [U37] (Bhagwatkar et al., 2025) Injection resistance and confidential-argument minimization.	User task, tool metadata, arguments and returned observations.	Rewrites arguments before tool execution and observations before agent ingestion.
IntentGuard (AAAI workshop) [U39] (Cong et al., 2026) Task-intent consistency and tool-chain safety.	Raw user prompt, plan, current call and recent call history.	Pre-execution learned checks; rewrite or abort.
Prudentia [U40] (Kolluri et al., 2026) Integrity and authorized information release under specified IFC policies.	Context/data labels, tool policies and planned expansions.	Existing IFC tool checks; strategic expansion and user endorsement requests.
ControlValve [U41] (Jha et al., 2025) Permitted invocation order and contextual use.	Invocation trace, conversation and edge rules.	Grammar/parser checks plus LLM rule judgments; permit, replan or reject.

Table 19: Trusted assumptions and supporting evidence

Source	Trusted assumptions	Evidence and limits
ShieldAgent [U09] (Chen et al., 2025)	Policy construction, LLM grounding and generated checkers; trusted integration.	Offline checks and online Agent Workflow Memory experiments. Table 4 conditions compliance on task success; it is not joint safe completion.
ToolGuards [U10] (Zwerdling et al., 2025)	Correct policy extraction and generated guard code; trusted read-only APIs and interception.	Human-authored compliance tests and tau-Bench subsets. Cannot catch a required call that is never proposed; deterministic execution does not prove correct translation.
AttriGuard [U11] (He et al., 2026)	Trusted user task, LLM attenuation/fuzzy judge and mandatory interception.	AgentDojo subset and sampled ASB; defense-aware adaptive search. Static/adaptive attenuation levels differ. Explicit delegation and text-only goals are excluded.
AgenTRIM [U12] (Betser et al., 2026)	Offline tool-role inventory, action judge, status manager and exposure filtering.	AgentDojo experiments and targeted attacks in Section 4.5. Distinguish the 500 offline inventory configurations from task-evaluation runs.
ClawGuard [U13] (Zhao et al., 2026)	Configured permissions, skill judge, sanitizer and enforced approval decisions.	Security evaluation denies all approvals. Utility uses separate approval budgets; attack continuation is not legitimate task success.

Source	Trusted assumptions	Evidence and limits
VIGIL (skills) [U14] (Li et al., 2026c)	Correct typed instrumentation, artifact binding, deterministic grounding and enforcement integration.	Typed traces and skill-bundle replay use external CLI logging stubs. This is not validation of live external effects or every integration path.
Runtime Governance [U15] (Kaptein et al., 2026)	Correct organizational state, identity and policy integration; eventual consistency remains a risk.	Described implementation and policy examples. Risk scores are not calibrated violation probabilities; full mediation and delegation provenance remain open.
PRISM [U16] (Li, 2026)	Gateway/plugin hooks, rule coverage, trusted session state and proxy controls.	Includes an 80-case live-model harness, distinct from mock tests. Not end-to-end gateway validation; string paths do not resolve symlink semantics.
AgentFlow [U17] (Ammanaghatta Shivakumar et al., 2026)	Faithful policies, labels, tool mappings and complete mediation; semantic errors can lie outside the model.	Live AgentDojo/AgentDyn, separate breadth replays, bounded SMT and ASB planner tests. Score postprocessing, coverage and task utility remain setting-specific.
FlowSeal [U18] (Shim et al., 2026)	Correct ownership/provenance, authentic consent, mandatory dispatch and LLM declassification.	Simulations, adapted privacy tasks and a small live MCP study. The theorem excepts incorrect SAFE declassification; authorized release is evaluated separately.
SPA [U19] (Girrens & Wang, 2026)	Deployment lattices, tool policies/bindings, deterministic interpreter and label-preserving persistence.	AgentDojo and a 97-task multi-query derivative. Existing injection weakly exercises delayed poisoning; strict IFC reduces utility, and abstract mapping often prevents execution.
Overlaying Governance [U20] (Ibrahim & Li, 2026)	Central trusted authorization service, authentic relation updates, controller clients and schema conditions A1-A6.	Conditional schema proofs and synthetic OpenFGA workloads. No live adversarial-agent task validation; allowed-envelope misuse and semantic scope ordering remain separate.
AgentTrust [U22] (Yang, 2026)	Trusted in-process runtime calls the interceptor, rules cover relevant patterns, labels are meaningful and escalation is honored.	300 curated and 630 independently constructed labeled action scenarios. The latter were retained after rules were patched using them and are explicitly not a zero-shot held-out set. Default no-match verdict is allow, distinct from exception/LLM-failure review. Safety contracts are regression tests, not formal soundness proofs. Distinguish production-only 95.0% verdict accuracy from 97.0% with benchmark-specific rules. Human agreement values are proposed targets, not measured agreement.
CommandSans [U29] (Das et al., 2025)	Learned token classifier, training labels and instruction/data distinction.	Five injection benchmarks; AgentDojo supplies task utility, other benchmarks may use proxies. Not a recipient-authorization or mandatory tool-effect reference monitor. CommandSans* additionally trains on synthetic malicious samples; do not generalize the clean-only training claim to both variants.

Source	Trusted assumptions	Evidence and limits
PROMPTGUARD (O-RAN) [U30] (Wang et al., 2026d)	Uncompromised RAN infrastructure, model parameters and independent verification xApp; trustworthy policy and semantic classifier.	OAIC testbed allocation/data-rate behavior and two-slice utility/reliability against statistical baselines. Do not convert semantic thresholding into a formal guarantee, infer that every supported mitigation mode was separately validated, or adopt the speculative explanation for the GPT-4o/o3 performance difference.
Minimizer / Sanitizer [U37] (Bhagwatkar et al., 2025)	Benign user and effective limited-privilege firewall models.	Four agent benchmarks; source-specific success checks and benchmark revisions. No universal guarantee. Important-instruction main experiments and tool-knowledge revised tables must stay distinct.
IntentGuard (AAAI workshop) [U39] (Cong et al., 2026)	Correct safety judgments and preserved original intent.	ASB attack families; attack success and detection metrics. RER credits attack detection, not only safe task completion. Semantic checks are not a formal authorization proof.
Prudentia [U40] (Kolluri et al., 2026)	Trusted user, planner and tools; correct labels and policies.	Instrumented AgentDojo and WASP; TCR at intervention budgets. Endorsement changes trust labels. Metrics are not a human approval study or guaranteed endorsement correctness.
ControlValve [U41] (Jha et al., 2025)	Trusted planning inputs; adequate generated policies and semantic judge.	CFH-Hard attacks on Magentic-One; adversarial-objective success. Section 4 acknowledges lax or erroneous graphs/rules. No end-to-end formal safety guarantee or control of agent internals.

Table 20: Evaluation units, conditions and unsupported transfers

Source	What was measured	Not established / interpretive limit
ShieldAgent [U09] (Chen et al., 2025)	Table 4 compliance among successful tasks.	Joint safe-and-successful completion across all attempted tasks.
ClawGuard [U13] (Zhao et al., 2026)	Security runs deny every approval request; utility has separate approval budgets.	Outcomes under actual user approval/denial behavior.
VIGIL (skills) [U14] (Li et al., 2026c)	Typed trace/artifact checks and skill-bundle replay with external CLI logging stubs.	Prevention of external effects in live integrated skills.
Constitutional monitoring [U21] (Storf et al., 2026)	Suspicion on complete recorded trajectories; unsuccessful side-task attempts are excluded from positives.	Online prevention, all-attempt detection or benign task completion.
AgentFlow [U17] (Ammanaghatta Shivakumar et al., 2026)	Live AgentDojo/AgentDyn, broader replays and an ASB planner test are distinct; ASB success is 2/1,200 despite zero attack success.	Uniform live evaluation across all benchmarks or useful protection from zero ASR alone.
AttriGuard [U11] (He et al., 2026)	Static level-2 and adaptive level-3 attenuation on a threat-model-filtered task set.	An unconditioned comparison on all AgentDojo tasks at one fixed configuration.

Source	What was measured	Not established / interpretive limit
SPA [U19] (Girrens & Wang, 2026)	Soft-hit reuse excludes turns whose producer failed to store an artifact; strict reuse retains those failures.	High end-to-end persistence utility from soft-hit reuse alone.
FlowSeal [U18] (Shim et al., 2026)	The confidentiality theorem excepts wrong SAFE declassification by the semantic checker.	Unconditional confidentiality from a mandatory dispatcher.
AgentTrust [U22] (Yang, 2026)	Action-verdict accuracy after rules were patched using the retained evaluation set.	Zero-shot held-out generalization or a formal soundness proof.
Auto Mode red-teaming [U23] (Remedios et al., 2026)	Replay, LLM simulation and live sandbox rollouts; benign action FPR and token-based latency estimates.	Full-product/sandbox security or measured end-to-end benign completion and deployment latency.
Overlaying Governance [U20] (Ibrahim & Li, 2026)	Schema-level conditional soundness and synthetic authorization-check/write latency.	Safe use of already-authorized resources by live agents, or cross-organization interoperability.
SafeAgent [U24] (Zhou et al., 2026)	Simulated action-risk evaluation with human agreement and terminal-subset judge checks; manual task utility also reported.	A mandatory runtime reference monitor or confirmed harmful effects for every unsafe proposed action.
Minimizer / Sanitizer [U37] (Bhagwatkar et al., 2025)	Four agent benchmarks; source-specific success checks and benchmark revisions. Separately reports clean utility and utility under attack where supported.	No universal guarantee. Important-instruction main experiments and tool-knowledge revised tables must stay distinct. Section 7 demonstrates a crafted bypass.
IntentGuard (AAAI workshop) [U39] (Cong et al., 2026)	ASB attack families; attack success and detection metrics. PNA and task success are separate from robust execution rate.	RER credits attack detection, not only safe task completion. Semantic checks are not a formal authorization proof. Named attack families; unrestricted adaptive evaluation not established.
Prudentia [U40] (Kolluri et al., 2026)	Instrumented AgentDojo and WASP; TCR at intervention budgets. Benign traces assume human approval; HITL load sums interventions on successful tasks.	Endorsement changes trust labels. Metrics are not a human approval study or guaranteed endorsement correctness. Benchmark attacks, not unrestricted adaptive compromise.
ControlValve [U41] (Jha et al., 2025)	CFH-Hard attacks on Magentic-One; adversarial-objective success. Answer accuracy and an LLM judge for open-ended tasks.	Section 4 acknowledges lax or erroneous graphs/rules. No end-to-end formal safety guarantee or control of agent internals. Stronger control-flow attacks are evaluated; completeness is not established.
AgentDAM [U31] (Zharmagambetov et al., 2025)	246 tasks across three self-hosted applications; leakage fraction uses all instances. Environment-state task completion measured separately; prompting improves privacy while lowering completion.	Action-text privacy judgments are not universal information-flow proofs. High probing scores cannot substitute for executed-agent measurements.
LPS-Bench [U32] (Chen et al., 2026b)	570 cases, 65 scenarios, seven domains and nine risk types in this version. Benign safety accepts risk-mitigated completion OR pausing for confirmation; cannot be equated with task completion.	Customized mock tools are deliberately decoupled from live systems. Safety-awareness scores do not certify irreversible real-world effects.

Source	What was measured	Not established / interpretive limit
RealSafe [U33] (Ma, 2025)	Fourteen scenarios; AgentBench-derived and generated API tasks. Standard/ambiguous helpfulness is completion; malicious helpfulness is identification/refusal, not benign completion.	Do not pool the differently defined helpfulness measures or interpret title wording as universal live-environment verification.
CI-Work [U34] (Fu et al., 2026)	125 seeds, each normally four sensitive and four essential entries; entry leakage and any-leak violation have different denominators. Conveyance is essential-entry transmission, not independent end-to-end task completion.	Reported human agreement is imperfect. Simulated traces and final-action judgments are not audited production information-flow enforcement.
Agentic Browsers / SOP [U36] (Roesner & Kohlbrenner, 2026)	Seven products, including Atlas modes; behavioral probes and an end-to-end Atlas demonstration. Capability probes, not a common legitimate-task completion benchmark.	Non-observation does not prove architectural impossibility. Results describe January-February 2026 versions, not current deployments.
OpenCUA (failure analysis) [U42] (Wang et al., 2025d)	Online OSWorld/Windows tasks; offline action matching is a separate unit. Task success under benign conditions.	Appendix E identifies repetition, premature termination and recovery failures, without calibrated category prevalence or authorization guarantees.

H.8 Version and Reading Anchors

These are targeted source readings, not full experimental replications. Publication metadata is distinct from the version actually inspected.

U01 Surapani et al. (2026). arXiv:2609.15906v1, submitted September 14, 2026 Sections 2.5-2.6, 8.1-8.3, 8.5 and 9.8; PDF pages 9-10, 37-41, 47, 50 and 57

https://arxiv.org/pdf/2609.15906v1

U02 Ling et al. (2026). arXiv:2606.10749v2, August 23, 2026; template ACM header is not verified publication metadata Sections 3.2-3.4, 7.2, 8.1-8.3, 10 and Table 3; PDF pages 7, 15, 17-18 and 23

https://arxiv.org/pdf/2606.10749v2

U03 Kim et al. (2026). 35th USENIX Security Symposium, August 12-14, 2026, proceedings PDF Sections 5.1-5.2.5, 5.4 and 5.6; Table 2; PDF pages 8, 10-11, 13 and 15

https://www.usenix.org/system/files/usenixsecurity26-kim-juhee-agentic.pdf

U04 Beurer-Kellner et al. (2025). arXiv:2506.08837v3, June 27, 2025 Section 3.1, introduction to Section 4, Section 4.3.3 and Appendix A; PDF pages 3-9, 16 and 32

https://arxiv.org/pdf/2506.08837v3

U05 Ng et al. (2026). arXiv:2608.11274v1, August 11, 2026 Sections 3.3-4.3, beginning of Section 5 and Appendix A; PDF pages 4-6 and 15

https://arxiv.org/pdf/2608.11274v1

U06 Chen et al. (2026a). ACL 2026 long paper 2106, July 2-7, 2026 Sections 4.1-4.2, 5.2-5.3, Table 3 and Limitations; PDF pages 5-6, 8 and 10

https://aclanthology.org/2026.acl-long.2106.pdf

U07 Yehudai et al. (2026). Findings of ACL 2026 paper 1330, July 2-7, 2026 Sections 5 and 7.2 and Limitations; PDF pages 5-6 and 9-10

https://aclanthology.org/2026.findings-acl.1330.pdf

U08 Wang et al. (2025a). arXiv:2504.15585v4, June 9, 2025; generic Journal of LaTeX Class Files header is not publication evidence Sections 6.2.1-6.2.2, 6.2.5-6.2.6, 6.3.2, 6.4 and 6.5; PDF pages 26, 28 and 30-32

https://arxiv.org/pdf/2504.15585v4

U09 Chen et al. (2025). ICML 2025, PMLR 267, chen25ae proceedings PDF Sections 3.1-3.3, Algorithm 1, Sections 4-5.2, Tables 2-4; PDF pages 3 and 5-8

https://raw.githubusercontent.com/mlresearch/v267/main/assets/chen25ae/chen25ae.pdf

U10 Zwerdling et al. (2025). EMNLP 2025 Industry Track, pages 595-606, November 4-9, 2025 Sections 3.1-4, Limitations and Appendix 7.2; PDF pages 3-7 and 10

https://aclanthology.org/2025.emnlp-industry.41.pdf

U11 He et al. (2026). USENIX Security 2026 proceedings, August 12-14, 2026 Sections 2.3, 3.2-4.4, 5.1, 5.6 and 7; Algorithm 1 and Tables 5-6; PDF pages 5-10 and 14-16

https://www.usenix.org/system/files/usenixsecurity26-he-yu.pdf

U12 Betser et al. (2026). arXiv:2601.12449v2, August 30, 2026 Sections 3.1-3.2, 4.1-4.2, 4.5 and Limitations; PDF pages 3-6 and 9

https://arxiv.org/pdf/2601.12449v2

U13 Zhao et al. (2026). arXiv:2604.11790v2, May 11, 2026 Sections II.B, III.A-B and IV.A-D; Algorithm 1, Table IV and Figure 4; PDF pages 3-5, 7-8 and 10-11

https://arxiv.org/pdf/2604.11790v2

U14 Li et al. (2026c). arXiv:2606.26524v1, June 25, 2026 Sections 3.2-4.2, 5.3, 6, 7.1 and 7.5; PDF pages 4-5, 8-9 and 12

https://arxiv.org/pdf/2606.26524v1

U15 Kaptein et al. (2026). arXiv:2603.16586v1, March 17, 2026; manuscript dated March 18 Sections 3.2-3.3, 4.1-4.6 and 6.1; PDF pages 8-9, 12-14 and 17

https://arxiv.org/pdf/2603.16586v1

U16 Li (2026). arXiv:2603.11853v1, March 12, 2026 Sections 2.3-2.4, 3.1-3.5, 5.5, 7 and Table 4; PDF pages 4-8, 15-16 and 20

https://arxiv.org/pdf/2603.11853v1

U17 Ammanaghatta Shivakumar et al. (2026). arXiv:2608.22868v1, August 24, 2026; cover dated August 25 Sections 3, 5.2-6.3, 7.2 and 8.1-8.9; PDF pages 5, 13-16 and 18-23

https://arxiv.org/pdf/2608.22868v1

U18 Shim et al. (2026). arXiv:2609.14003v1, September 12, 2026 Sections V-VI, VII-IX and Theorem 1; PDF pages 5-13

https://arxiv.org/pdf/2609.14003v1

U19 Girrens & Wang (2026). arXiv:2608.27234v2, September 4, 2026 Sections 3, 4.1-4.4, 5.4-6.2, 7.1-7.3 and 8; PDF pages 4, 6, 9, 11 and 13

https://arxiv.org/pdf/2608.27234v2

U20 Ibrahim & Li (2026). arXiv:2606.03518v1, June 2, 2026 Sections 3.2-3.3, 4.1-4.3 and 5.1-5.2; applicability conditions A1-A6 and Theorem 5.3; PDF pages 4-11

https://arxiv.org/pdf/2606.03518v1

U21 Storf et al. (2026). arXiv:2603.00829v2, May 30, 2026; cover identifies ICML 2026, PMLR 306 Sections 3.3-3.4, 4.1-4.2, 6 and Appendix D.1; PDF pages 5-6, 9 and 23

https://arxiv.org/pdf/2603.00829v2

U22 Yang (2026). arXiv:2605.04785v1, May 6, 2026 Sections 3.1-3.3, 4.5-4.7, 4.10, 5.1-5.3, 6.1-6.3 and 7.1; PDF pages 5-6, 10, 12-16 and 24

https://arxiv.org/pdf/2605.04785v1

U23 Remedios et al. (2026). arXiv:2609.19587v1, September 17, 2026 Sections 3.1-3.4, 4.1-4.2, 4.7-4.8 and 5; PDF pages 3-5 and 7-9

https://arxiv.org/pdf/2609.19587v1

U24 Zhou et al. (2026). ACL 2026 long paper, pages 32516-32543, July 2-7, 2026 Sections 3.1-3.3, 4.1-4.3, 4.6 and evaluator definition; PDF pages 3-7 and 14

https://aclanthology.org/2026.acl-long.1501.pdf

U25 Deng et al. (2024). arXiv:2406.02630v2, September 6, 2024; not the final journal text Section 2.2, Section 3 and future directions; PDF pages 5, 13, 25

https://arxiv.org/pdf/2406.02630v2

U26 Meng et al. (2026). Preprints.org 202604.0428 v3, April 28, 2026 Definition/intro and Section 6.6, especially 6.6.4-6.6.7

https://www.preprints.org/manuscript/202604.0428

U27 He et al. (2025). arXiv:2407.19354v2, November 2, 2025; placeholder journal fields not treated as publication metadata Sections 1, 5.3.2 and 5.3.3

https://arxiv.org/html/2407.19354v2

U28 Wang et al. (2025f). arXiv:2409.14457v3, June 2, 2025; IEEE DOI 10.1109/COMST.2025.3576176 metadata separately checked Security threat taxonomy and countermeasure Table VIII; PDF pages 24-25, 29-30

https://arxiv.org/pdf/2409.14457v3

U29 Das et al. (2025). arXiv:2510.08829v1, October 9, 2025 Sections 4-5.1, PDF pages 4-6

https://arxiv.org/pdf/2510.08829v1

U30 Wang et al. (2026d). FutureG workshop 2026 proceedings PDF Sections II-IV, equations 7-13; PDF pages 2-4

https://www.ndss-symposium.org/wp-content/uploads/futureg26-22.pdf

U31 Zharmagambetov et al. (2025). ICML 2025 Computer Use workshop manuscript Sections 3.1-3.4 and 4.1-4.4, PDF pages 3-8

https://openreview.net/pdf/06bde18559fc6ddf4c9a1168450a34a055225fd9.pdf

U32 Chen et al. (2026b). arXiv:2602.03255v1, February 3, 2026 Sections 3.1-3.3 and 4.1; PDF pages 4-6

https://arxiv.org/pdf/2602.03255v1

U33 Ma (2025). COLING 2025, pages 9586-9617 Sections 3-5.1, PDF pages 3-6

https://aclanthology.org/2025.coling-main.642.pdf

U34 Fu et al. (2026). ACL 2026 Industry Track, pages 1483-1508 Sections 4.1-4.5; PDF pages 3-5

https://aclanthology.org/2026.acl-industry.103.pdf

U35 Addis et al. (2026). IEEE Access 14, 100884-100917; DOI 10.1109/ACCESS.2026.3707573. Full-text header: published June 26, 2026; current version July 9, 2026. Full-text header; Sections I-C/D, V-B1/2/4 and V-C4; Table 9. Public full-text transcription inspected, not a locally acquired PDF or experiment replication.

https://doi.org/10.1109/ACCESS.2026.3707573

U36 Roesner & Kohlbrenner (2026). Author-hosted ICLR 2026 workshop paper; project page dated April 15, 2026 Sections 2.2-5; Tables 1-3; project-page date.

https://homes.cs.washington.edu/~franzi/pdf/roesner_kohlbrenner_2026_agentic_sop.pdf

U37 Bhagwatkar et al. (2025). arXiv:2510.05244v1 (2025) Section 3, Algorithm 1; Sections 4.1-4.2, 6-7; Appendix C.

https://arxiv.org/html/2510.05244v1

U38 Luo et al. (2025). arXiv:2503.21460v1 (2025) Section IV-A, Table III; Sections IV-B1-IV-B2.

https://arxiv.org/html/2503.21460v1

U39 Cong et al. (2026). AAAI 2026 Trustworthy Agentic AI workshop; official listing dated January 27, 2026 Sections 3.2-3.3, Algorithm 1; Sections 4.1-4.5.

https://trustagenticai.github.io/AAAI2026/AAAI-Workshop/47.pdf

U40 Kolluri et al. (2026). arXiv:2602.11416v1, February 11, 2026 Sections 2-5; Section 3 metric definitions and Section 4 endorsement.

https://arxiv.org/html/2602.11416v1

U41 Jha et al. (2025). arXiv:2510.17276v1, October 20, 2025 Sections 4-6, particularly Section 4 limitations and Section 5.2.

https://arxiv.org/html/2510.17276v1

U42 Wang et al. (2025d). arXiv:2508.09123v1 (2025) Section 4.1 and Appendix E (benign execution and recovery failures).

https://arxiv.org/html/2508.09123v1

U43 Li et al. (2026b). Author-hosted 71-page revision, repository-dated July 21, 2026 (commit 4884d874); this is not a venue publication date. Sections 8.3, 8.6-8.7, 9.1, 9.6-9.7 and 12.1; printed pages 35, 38-40, 45-46 and 49. The URL pins the inspected PDF revision.

https://raw.githubusercontent.com/Picrew/LLM-Harness/4884d8748b92ee375b8f460c232ab3ca96d186d7/docs/main.pdf

I Supplementary Named-Source Route

Nine named-paper leads supplied during pre-submission review were checked separately on September 23, 2026. Targeted primary-text readings retained five mechanisms, three evaluations, and AI Control as adjacent threat-model context. All nine have inspectable text predating the September 21 cutoff. They are neither extra hits from the frozen queries nor completed screening of the citation-export queue. The source statements, reading anchors, and review inferences are recorded below and in the companion dataset. This route shares the AI-assisted review status described in Appendix H.

Table 21: Named-source coverage: mechanisms, evidence, and interpretation. H/M use the core coding rules; evaluation and context records have no mechanism code.

Source / inspected version	Source contribution / evidence	Review interpretation
LlamaFirewall (Chennabasappa et al., 2025) arXiv:2505.03574v1 H=T; M=S+E	Layered input, alignment and code scanners; alignment uses user goal, bounded trace and selected action. AgentDojo is an offline incremental rescan of fixed traces; flagged successful attacks are credited as defended.	Replay credit does not establish intervention before the harmful effect or agent recovery after blocking. Anchors: Sections 3, 4.2-4.4; Appendix C.1, C.4
f-secure LLM system (Wu et al., 2024) arXiv:2409.19091v1 H=T; M=F	An integrity-labeled planning pipeline filters untrusted inputs, structures executable steps and propagates labels. Conditional execution-trace non-compromise argument, injection cases and functionality benchmarks.	The formal claim depends on trusted labels and monitor/executor behavior; it is distinct from arbitrary resource authorization. Anchors: Sections 4-7; Definition 6.1, Theorem 6.2

Source / inspected version	Source contribution / evidence	Review interpretation
Conseca (Tsai & Bagdasarian, 2025) HOTOS 2025 proceedings, pp. 8-17 H=A; M=S	Trusted task context feeds policy generation; deterministic API and regex-argument checks admit or reject proposed calls. A Linux computer-use prototype and 20 locally crowdsourced tasks selected with Gemini; Figure 3 reports averages over five trials.	Generated policy fidelity and the choice of trusted context remain premises; synthetic tasks bound empirical transfer. Anchors: Sections 3.2-3.4, 4-4.1 and 5; Figure 3; printed pp. 10-12 (PDF pp. 3-5). Policy Limitations and Agent Integration are paragraphs within Section 4.1.
Adaptive out-of-band evaluation (Narisetty et al., 2026) arXiv:2606.26479v1	A defense-aware attack targets Progent's policy-update component with a Qwen2.5-7B agent and policy model. Three repeated runs of a fixed AgentDojo subsample; one defense and one hand-crafted adaptive attack family.	The paper explicitly limits its inference; its narrow evidence should not become a general ranking of in-band versus out-of-band defenses. Anchors: Sections 10-12; Tables 3-4
RTBAS (Zhong et al., 2025) arXiv:2502.08966v1 H=T; M=F+E	A learned dependency screener chooses a label, actual redaction restricts model input, and tool policies require confirmation for disallowed flows. AgentDojo injection evaluation and a separate privacy-leakage study.	Redaction before generation is essential to the mechanism; dependency prediction alone is not its enforcement argument. Anchors: Section 7.3, Algorithms 2-4; Sections 8.1-8.2
AirGapAgent (Bagdasarian et al., 2024) arXiv:2405.05175v2; CCS 2024 H=A; M=E	A minimizer uses trusted task context and user data; a separate conversational agent sees only the selected data and third-party messages. Privacy/utility experiments under context hijacking; Section 4.2 acknowledges the model-based minimizer's limitations.	Isolation bounds available data, while the correctness of the selected disclosure subset remains a learned premise. Anchors: Sections 4.1-4.2 and 5-6
ST-WebAgentBench (Levy et al., 2024) arXiv:2410.06703v1	Completion under Policy multiplies task completion by an indicator of zero measured policy violations. Web tasks with organizational/user/task policy hierarchy and trajectory-aware evaluators.	It is direct prior work on jointly measuring legitimate completion and policy adherence, within the specified evaluator coverage. Anchors: Sections 3.1-3.3, 4; Appendix A.2.1
tau-bench (Yao et al., 2024) arXiv:2406.12045v1	Simulated users interact with tools and a policy-guided agent; final database states and repeated-trial reliability determine outcomes. Retail and airline domains; source discusses user-simulator limits and additional rule-following metrics.	Final-state matching supports task outcomes, not observation of every intermediate policy-sensitive action. Anchors: Sections 3-6
AI Control (Greenblatt et al., 2024) ICML 2024, PMLR 235:16295-16336	Protocols combine trusted and untrusted models and limited trusted auditing to resist intentionally subverted programming outputs. Red-teamed coding protocols including monitoring and trusted editing; trusted human labor is simulated with GPT-4.	Intentional model subversion is an important adjacent threat model, not an equivalent injection benchmark. Anchors: Sections 1-2; Table 1

J HM Decision Units

The table fixes the unit of modality coding. Upstream generation/grounding remains in D2 even when its output is consumed by an explicit checker. In contrast, a direct learned verdict or semantic-score threshold is E. These unit descriptions remain the review synthesis; the separate independent coding and subsequent author adjudication in Appendix B.1 leave the core H/M codes unchanged. Source versions and anchors are shared with Appendix A.

Table 22: Named checking stages behind the core HM matrix.

System / M	Coded decision unit
NeMo Guardrails / S+E	Configured Colang flow checks (S) and semantic event/intent mapping used for flow decisions (E). The codes describe these components, not every possible custom rail.
TrustAgent / E	Regulation-compliance inspection of proposed actions against the supplied trajectory (E).
AgentSpec / S	Configured trigger-predicate evaluation (S); model-assisted reconsideration is an intervention option, not the coded predicate.
GuardAgent / S	Execution of generated guard-check code (S), excluding the upstream model that writes it.
Progent / S	Tool-call policy evaluation and SMT validation of proposed policy updates (S), excluding policy generation.
RvLLM / S	Rule consistency and inference over supplied extracted propositions (S), excluding fact extraction.
AgentArmor / S+F	Explicit graph-policy checks (S) with runtime label propagation (F), excluding dependency inference.
CaMeL / S+F	Tool security-policy predicates over capabilities (S) and interpreter flow propagation (F), excluding generated control code.
Fides / F	Flow-sensitive visibility and tool-execution restrictions using accumulated labels (F).
Permissive IFA / F+E	Utility-based restricted-context selection (E in the inspected model-perplexity instantiation) and restricted-context regeneration/labeling (F).
IPIGuard / S	Dependency-graph traversal and query-only expansion restrictions (S), excluding learned graph and argument construction.
DRIFT / S+E	Planned-call constraints (S), learned privilege/intent-sensitive revision and injection isolation (E).
Task Shield / E	Task-contribution judgments for candidate calls and instructions (E).
MELON / E	Learned embedding-similarity detection over original and cached masked calls (E), including deterministic thresholding of that score.
ToolSafe / E	TS-Guard call-safety verdict from request, history, call and tool specifications (E).
IsolateGPT / S	Permission enforcement at hub-mediated exchanges (S), excluding learned plans/warnings and treating user consent as an external input.
ACE / S+F	Generated-plan verification (S) and worker communication flow checks (F), excluding plan generation.
AgenTEE / –	Attested execution and channel substrate; no semantic action-policy check is coded (–). Cryptographic operations are not omitted from the source record.
Authenticated delegation / S	Receiving-service validation of identity, scope and token validity (S).
APC / S	Serialized session-policy admission and cumulative accounting (S).
DWT broker / S	Broker validation of workload-bound grants and current revocation metadata (S).
ProbGuard / S	Risk/model checking against an estimated abstract transition model (S), excluding offline parameter estimation.
SafeClaw-R / S+E	Orchestrator safety-node admission (S) and model-executed safety-skill verdicts (E).
Sentinel / S+F	Graph-invariant checks (S) with accumulated taint propagation (F), excluding entity grounding.
Distributed Sentinel / S+F	Local/remote graph predicates (S) with Semantic Taint Token propagation (F), excluding entity grounding.