MADS-CPS: A Machine-Checkable Run-Level Evaluation Contract for Trustworthy Agentic Cyber-Physical Workflows

Mateo Petel
Stanford University
Stanford, California, USA
mpetel@stanford.edu

Abstract

Agentic cyber-physical systems integrate planning, tool use, sensing, coordination, and actuation, so a component fault can propagate across software state, physical state, and downstream authorization. Existing work distributes the relevant functions across capability measurement, telemetry, action mediation, and lifecycle risk management. This division leaves no minimal evidence object from which an independent evaluator can determine whether one recorded run is admissible for audit, control-plane replay, and release. We introduce MADS-CPS, a machine-checkable run-level evaluation contract for agentic cyber-physical workflows. The contract binds a declared assurance envelope to four required artifacts, independently recomputed predicates, monotone conformance tiers, explicit disclosure states, and fail-closed point-of-no-return (PONR) release gating. The resulting verdict depends only on emitted evidence and declared interface relations, preserving a common verification boundary across heterogeneous controllers.

We instantiate MADS-CPS in a robot-centric autonomous-laboratory profile and evaluate the contract through five studies and a replay mutation stress test. A 24-case challenge corpus spanning nine fault families yields exact agreement between predeclared and observed outcomes. Structured public redaction preserves schema, linkage, and integrity checks, with report recomputation, control-plane replay, and PONR semantics reported as unavailable. A process- and implementation-independent verifier attains exact core-report agreement of 1.00 across two scenarios and twenty seeds per scenario, and all 110 replay mutations are correctly localized. The broader stress studies separate evidence validity from operational outcomes. Across 480 controller–scenario–regime runs, conformance and strong replay remain 1.00; a focused coordination shock yields productive success rates of 0.55 and 0.00. A 720-run synthetic version-shift study preserves conformance and replay at 1.00 as productive success declines from 0.7208 to 0.6375 in the regressive condition. MADS-CPS makes structural run admissibility mechanically decidable and externally inspectable under a declared trust boundary. Physical ground truth, deployed-system certification, and continuous runtime safety remain obligations of surrounding controls.

Keywords

agentic AI evaluation, cyber-physical systems, run-level assurance, auditability, replay, release gating, autonomous laboratories

1 Introduction

Agent evaluation has expanded from static task accuracy to interactive, stateful execution. AgentBench, WebArena, and τ-bench measure long-horizon planning and tool-mediated interaction [6, 18, 20]. Benchmark-methodology research shows that coverage, reward construction, and reproducibility determine the validity of such measurements [19, 21]. Production evidence adds a deployment constraint. Agents remain heavily supervised, and reliability dominates operational evaluation [10].

Cyber-physical deployment adds an admission question that task success cannot answer. A run can complete its assigned task and still lack the evidence required for an irreversible or authoritative transition. The admission decision depends on evidence completeness, independent report recomputation, disclosure-sensitive predicate availability, and correct denial at the release boundary. Raw traces support diagnosis, and governance frameworks organize lifecycle controls. A machine-checkable authorization rule requires an additional relational object that connects one recorded execution to its release conditions.

This paper introduces MADS-CPS, a machine-checkable run-level evaluation contract for trustworthy agentic cyber-physical workflows. Each run emits a structured trace, an evaluation report, an evidence bundle, and a release manifest. A declared assurance envelope fixes the scenario, verification mode, PONR obligations, replay-equivalence policy, and evidence trust policy. The conformance checker computes a tiered verdict from those relations, and a fail-closed gate consumes the verdict at an authorization boundary.

The contribution is also to structural admissibility under a declared envelope. Sensor truth, domain safety, and autonomous-product certification require evidence beyond internal artifact consistency. A compromised capture path can emit mutually coherent artifacts that lack an authenticated connection to the physical process. Deployment profiles therefore bind the run contract to the trust anchors required for the application, such as authenticated devices, secure time, signed measurements, remote attestation, or independent observation.

The paper makes four contributions.

(1) A run-level evidence contract that standardizes artifacts, verification modes, replay-equivalence policies, monotone conformance tiers, and fail-closed release semantics.
(2) Controller-agnostic conformance defined at the evidence boundary, with controller identity and hidden optimization state excluded unless they alter a declared predicate.
(3) Explicit trust and redaction boundaries that preserve checkable structural integrity under restricted disclosure and assign physical ground-truth binding to the deployment profile.
(4) An executable evaluation covering adversarial conformance, disclosure-sensitive auditability, independent recomputation, controller stress, and synthetic version shift.

2 Positioning and Related Work

Agent benchmarks and representative evaluation. Interactive agent evaluation has progressed from broad environment suites to long-context and domain-specific tasks. AgentBench, WebArena, and τ-bench establish task-level measurements for planning and tool-mediated interaction [6, 18, 20]. AgencyBench extends the horizon to million-token workflows, and scientific benchmarks ground the task distribution in research practice [3–5]. ADePT further organizes autonomous-laboratory capability at the robotics level [12]. Methodology work shows that representativeness and benchmark construction constrain the meaning of the resulting scores [11, 19, 21]. This literature evaluates capability and distributional validity. MADS-CPS specifies the evidence relation under which one recorded run can support audit, replay, and release.

Production evaluation and observability. Operational evaluation requires trace semantics that remain interpretable across deployed components. Production studies identify human evaluation and reliability as central constraints [10], and AgentTrace structures logs for agent-system observability [1]. OpenTelemetry provides interoperable semantic conventions for telemetry, with generative-AI conventions maintained in a dedicated specification repository [8]. These mechanisms support collection and correlation. MADS-CPS treats the trace as one member of a committed evidence family, then adds normative predicates, independent recomputation, disclosure states, and a release verdict.

Runtime mediation and trace contracts. Runtime assurance controls actions at the execution boundary. AgentTrust evaluates proposed tool calls and returns allow, warn, block, or review decisions [17]. Trace-based orchestration assurance records message-action traces and applies contracts, replay, fault injection, and action mediation [9]. MADS-CPS consumes the complete run package and checks whether recorded decisions, outcomes, reports, and commitments jointly satisfy a declared release policy. Composition occurs when runtime verdicts enter the trace as signed events and the run contract verifies their relation to subsequent outcomes.

Risk, assurance, and provenance. Lifecycle assurance and artifact provenance provide complementary outer layers. NIST AI RMF, its generative-AI profile, and UL 4600 define risk and safety processes, with SACM supplying a structured representation for assurance claims [2, 7, 14, 16]. Software provenance systems bind artifacts to declared production steps through signed metadata [13, 15]. MADS-CPS supplies the run-scoped object that these processes can reference. Its verdict concerns one recorded execution under a declared envelope.

3 Run-Level Contract

3.1 Declared Envelope and Artifacts

A run r is evaluated relative to a declared assurance envelope

$$\mathcal{E} = (s, v, P, \rho, \tau),$$

where s identifies the scenario or profile, v is the verification mode, P is the set of release-relevant point-of-no-return (PONR) obligations, ρ defines replay equivalence, and τ records the trust-boundary requirements for evidence capture. Each run emits

$$A(r) = (T_r, R_r, B_r, M_r),$$

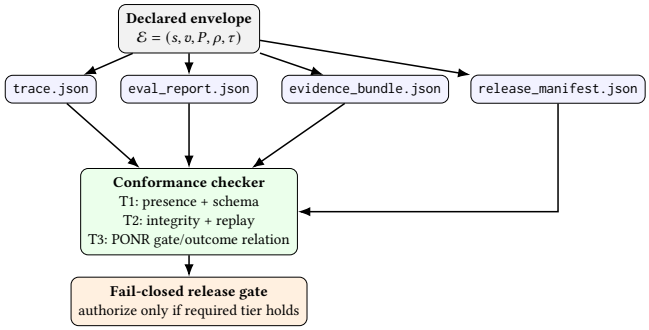

Figure 1: Each run is checked against a declared envelope. The verdict is computed from standardized artifacts and consumed at a release boundary.

where T_r is a structured trace, R_r is an evaluation report, B_r is an evidence bundle, and M_r is a release manifest.

Verification modes. The contract uses public, evaluator, and regulator modes. Each mode determines which predicates must be available. Public mode permits structure-preserving redaction. Evaluator and regulator modes can require replay-capable payloads and stronger trust anchors. The mode changes the required evidence without changing the meaning of evidence that is present.

Replay equivalence. The policy ρ defines the comparison relation $\approx_{\mathcal{E}}$ used by replay. L0 replay recomputes decision and enforcement outcomes from recorded state transitions. L1 replay also consumes recorded observations. Continuous signals need not match bit-for-bit. A profile can declare canonicalization rules, numeric tolerances, temporal windows, and invariant checks. Replay succeeds when the recomputed report is equivalent under ρ, which limits false failures caused by benign floating-point or timing variation without silently accepting undeclared drift.

3.2 Conformance Tiers

Let present$(A(r))$ require all four artifacts, and let schemaOK$(A(r))$ validate their declared schemas. Let linked$(A(r), \mathcal{E})$ require consistent run, scenario, schema-version, and artifact references. Let commitOK$(B_r, M_r, A(r))$ recompute every declared digest, reject duplicate or traversal-bearing paths, and verify final-state commitments. Let $\widehat{R}_r$ and $\widehat{S}_r$ be the independently recomputed evaluation report and control-plane terminal state. For each PONR obligation p, ponrOK(p, T_r) requires exactly one pre-transition gate, exactly one terminal outcome, gate/outcome agreement, and the absence of completion after denial. Define structOK = present $\wedge$ schemaOK $\wedge$ linked and $\text{reportOK}_\rho = (\widehat{R}_r \approx_\rho R_r)$. The tiers are

$$\begin{aligned}
\text{Tier1}(r) &\iff \text{structOK}(A(r), \mathcal{E}),\\
\text{Tier2}(r) &\iff \text{Tier1}(r) \wedge \text{commitOK}(r)\\
&\qquad \wedge \text{reportOK}_\rho(r) \wedge \text{stateOK}(r),\\
\text{Tier3}(r) &\iff \text{Tier2}(r)\\
&\qquad \wedge \forall p \in P,\ \text{ponrOK}(p, T_r).
\end{aligned}$$

Table 1: Adjacent evaluation layers and the specific object contributed by MADS-CPS. Entries report typical primary emphasis; specific implementations can support additional functions.

Approach class	Interactive tasks	Runtime telemetry	Standard run evidence	Independent recomputation	Release verdict	Primary emphasis
Agent benchmarks	yes	partial	uncommon	uncommon	uncommon	Capability, policy following, task success
Observability systems	optional	yes	partial	optional	uncommon	Tracing, debugging, monitoring
Runtime mediators	optional	yes	decision log	optional	action-level	Pre-execution tool or action control
Risk / assurance frameworks	no	partial	profile-specific	profile-specific	policy-level	Lifecycle governance and structured argument
MADS-CPS	profile-specific	yes	yes	yes	yes	Run admissibility under a declared contract

A profile with no PONR obligation can declare Tier 2 as its required release tier. Unknown scenarios fail envelope resolution instead of inheriting an empty obligation set.

Release semantics. A normal release path authorizes a transition only when the tier required by $\mathcal{E}$ holds. Missing, unverifiable, or contradictory evidence therefore denies release. An emergency override, when a deployment permits one, is a distinct authorization path with separate authority, logging, and accountability. It never changes a failed conformance verdict into a passing one.

3.3 Trust Boundary and Ground-Truth Binding

The checker establishes consistency between declared artifacts and predicates. It cannot establish that the artifacts faithfully represent the physical world unless the capture path supplies that connection. The trust policy τ therefore declares which producers, sensors, clocks, keys, and measurement channels are trusted or independently corroborated. A deployment can require device-bound signatures, secure timestamps, remote attestation, append-only storage, cross-sensor checks, or independent witnesses. If those requirements fail, the checker can issue a distinct ground-truth-binding failure even when schemas and internal hashes remain valid.

This distinction blocks an otherwise serious overclaim. A compromised controller could fabricate a trace, report, bundle, and manifest that are mutually consistent. MADS-CPS would detect the fabrication only when it violates a declared trust anchor or independent observation. Structural admissibility and physical truth are separate claims, and the envelope must state which one is being established.

3.4 Key Properties

Tier monotonicity. Each higher tier conjoins every obligation of the lower tier with additional predicates. Tier 3 therefore implies Tier 2, and Tier 2 implies Tier 1.

Controller-agnostic conformance. Fix $\mathcal{E}$. If two runs produced by different controllers agree on every predicate read by the checker, they receive the same verdict. This equality concerns interface-level invariance and leaves competence, productivity, and physical safety outside the verdict.

Redaction boundary. Structure-preserving redaction can retain event order, types, timestamps, commitments, and integrity metadata. Schema and integrity checks can remain available even when semantic replay and payload-dependent PONR checks are unavailable. The checker records that availability state explicitly instead of treating omitted semantics as successful.

4 Reference Instantiation and Protocol

We instantiate the contract in a robot-centric autonomous-laboratory thin slice. This reference organism combines planning, tool use, coordination, evidence emission, and release-relevant decisions. The evaluation runs entirely in synthetic software.

The implementation emits the four artifacts shown in Figure 1. The Lab scenario uses an empty PONR obligation set and a Tier-2 release requirement. The laboratory profile contains a disposition-commit obligation, which exercises Tier 3 and release gating.

We evaluate five questions and reuse the portfolio's independent replay-mutation study.

- **E1.** Does the checker assign the intended tier to injected structural, semantic, replay, report, and PONR faults?
- **E2.** Which predicates remain computable under restricted disclosure?
- **E3.** Does an implementation-independent process reproduce the reported evaluation result?
- **E4.** Does evidence-layer admissibility remain distinct from controller productivity under coordination stress?
- **E5.** Does a fixed contract expose a synthetic version-induced regression under unchanged interface semantics?

The protocol first validates the admission rule itself. E1 applies 24 predeclared cases, and E2 changes disclosure without changing the underlying run. E3 then removes producer-code trust through a standard-library verifier and adds a 110-variant mutation sweep. The final studies hold artifact schemas fixed as operational behavior changes. E4 spans 480 runs over two controller adapters, three scenarios, four regimes, and twenty seeds. E5 spans 720 runs across stable, benign, and regressive synthetic version conditions. The E5 conditions test contract behavior under a controlled shift without emulating a provider API or estimating production prevalence.

5 Results

5.1 E1: Adversarial Conformance Corpus

The checker agrees with all 24 predeclared outcomes in Table 3. The most diagnostic negative case is `unsafe_ponr_completion`. Its mutator refreshes every digest after an unsafe completion, so stale-hash detection provides no signal; rejection depends on relational

Table 2: Evaluation map.

Study	Primary unit	Supported claim
E1	run directory	Tier failures are mechanically decidable for the injected cases.
E2	predicate × mode	Restricted disclosure can preserve a subset of audit predicates.
E3	run × seed	Reports are recomputable by an independent verifier.
E4	controller × regime	Admissibility and productivity can separate under stress.
E5	version × cell	A fixed contract can expose bounded version-sensitive regressions.

Table 3: E1 oracle agreement by fault family. The three positive controls include a fail-closed PONR denial that remains Tier-3 admissible.

Fault family	Cases	Agreement
Positive controls	3	1.00
Presence / schema	3	1.00
Integrity	4	1.00
Cross-artifact linkage	2	1.00
Envelope resolution	1	1.00
Trace semantics	3	1.00
Replay	3	1.00
Report recomputation	1	1.00
PONR relation	4	1.00
Total	**24**	**1.00**

Table 4: E2 predicates computed from executable full and redacted packages. "U" denotes unavailable; failure is a separate state.

Package	Schema	Link	Integrity	Report	Replay	PONR
Evaluator full	pass	pass	pass	pass	pass	pass
Regulator full	pass	pass	pass	pass	pass	pass
Public full	pass	pass	pass	pass	pass	pass
Public redacted	pass	pass	pass	U	U	U

Table 5: E3 implementation-independent report recomputation. Brackets are 95% intervals over seed-level task-latency summaries.

Scenario	Seeds	Exact match	Mean $p95$ ms [95% CI]
`lab_profile`	20	1.00	49.19 [38.94, 59.43]
`lab`	20	1.00	35.37 [27.11, 43.62]

PONR verification. The remaining cases isolate structural integrity, cross-artifact linkage, trace semantics, report recomputation, and replay attestation. The result establishes exact behavior for this designed corpus; population-level attack prevalence remains outside the study.

5.2 E2: Executable Disclosure Modes

Structured redaction preserves schema validity, cross-artifact linkage, and committed-content integrity. It withholds the payload needed for report recomputation, control-plane replay, and PONR semantics. The checker represents these predicates as unavailable, preventing restricted disclosure from being misreported as either semantic success or artifact corruption.

Table 6: Replay mutation stress test reused from the portfolio's P3 study.

Mutation family	Variants	Correctly localized
Payload corruption	30	30
Omission	20	20
Duplication	20	20
Reordering	20	20
Schema drift	10	10
Hash mismatch	10	10
Total	**110**	**110**

Table 7: Focused E4 scheduling scenario under coordination shock. Each row aggregates twenty seeds.

Controller	Conf.	Replay	Prod.	Safe NP	Tasks / $p95$ ms
`centralized`	1.00	1.00	0.55	0.00	3.30 / 147.25
`rep_cps`	1.00	1.00	0.00	1.00	0.00 / 0.00

Table 8: E5 summary under fixed artifact-interface semantics. Each version condition contains 240 runs.

Condition	Conf.	Replay	Prod.	Safe NP	Tasks / $p95$ ms
V0 stable	1.00	1.00	0.7208	0.0833	3.6917 / 84.47
V1 benign	1.00	1.00	0.7208	0.0833	3.6917 / 62.70
V2 regressive	1.00	1.00	0.6375	0.1667	3.3583 / 81.52

5.3 E3: Independent Recomputation and Mutation Stress

The verifier executes in a separate process and implements report recomputation without importing the producer metric implementation. Its source digest is `20ea8d3...b66b9c25`. Exact core-report agreement is 1.00 in both scenarios. The result covers control-plane reconstruction and report recomputation. Physical and bit-identical hardware replay remain outside its scope.

The mutation sweep complements E1 by perturbing replay corpora at finer granularity. Every one of the 110 variants is localized to its expected mutation family. This result tests deterministic corruption localization under the supplied mutation operators; adaptive semantic forgery remains outside its scope.

5.4 E4: Admissibility Versus Productivity

Across all 480 E4 runs, structural conformance and strong replay are 1.00. The focused cell in Table 7 shows why those verdicts must remain separate from productivity. The centralized adapter completes productive work on 55% of seeds. The representative-CPS adapter denies every run at the scheduling gate, producing safe nonproductivity on all seeds. Both controllers emit internally admissible evidence. The result establishes evidence-layer invariance for the tested interface variants and leaves controller competence outside the verdict.

5.5 E5: Synthetic Version Shift

The regressive condition lowers productive success by 0.0833, raises safe nonproductivity by 0.0833, and reduces mean completed tasks by 0.3333, with zero change in conformance or strong replay. The strongest local effect occurs for the representative-CPS adapter on

the baseline scheduling scenario. Productive success falls from 1.00 to 0.00, safe nonproductivity rises from 0.00 to 1.00, and completed tasks fall from 4.00 to 0.00. The contract exposes the behavioral regression and correctly leaves artifact validity unchanged.

5.6 Reproducibility Binding

The hardened implementation is fixed at commit `1e84f8e...8cccc`; the generated results are fixed at `ab4ad06...fa38f`. Eighty-one focused P0/P3 tests pass with zero failures, errors, or skips. The release manifest records the exact experiment commands, independent-verifier digest, GitHub Actions run, complete evidence-artifact digest, and SHA-256 digest of each table-generating artifact. Every camera-ready value is generated from those executable outputs and bound to the manifest.

6 Discussion

The results separate structural admissibility from operational performance. E1 and E2 establish that integrity failure and predicate unavailability are distinct verifier states. E3 shows that report recomputation can be independent of producer code under declared control-plane semantics. E4 and E5 then hold the evidence contract fixed as productivity changes. A stable conformance verdict therefore indicates interface-level validity, without implying stable capability or safe physical behavior.

Composition with adjacent infrastructure occurs through the evidence boundary. OpenTelemetry-compatible spans can populate normalized trace events, after which signed runtime verdicts provide action-level evidence and provenance attestations bind the package to its execution environment. SACM or a sector safety case can cite the resulting run verdict. MADS-CPS specifies the internal relations and release condition for this composed object.

The trust boundary determines the strongest defensible claim. Hashes and replay protect consistency after capture, whereas sensor truth depends on the evidence-to-world binding declared in τ. A high-consequence deployment should require the relevant device identities, authenticated measurements, secure time, attested execution, append-only storage, or independent observations. Missing trust anchors produce a ground-truth-binding failure even when the package remains internally valid.

7 Limitations

The evaluation uses synthetic workflows in a software-only environment. Real sensors introduce noise, calibration error, packet loss, latency, and environmental nondeterminism. The implemented replay is deterministic control-plane reconstruction; the more general tolerance policy in ρ requires domain-specific validation for continuous physical signals.

PONR gating operates at pre-transition and release boundaries. Continuous plant stabilization, trajectory interruption, and emergency stopping remain responsibilities of runtime safety controllers and physical interlocks.

The 24-case corpus and 110 mutation variants are designed tests whose distribution differs from an operational attack distribution. A fully compromised evidence producer that controls trusted signing material remains outside the demonstrated threat model. Stronger roots of trust, protected capture, independent sensing, or cross-domain attestation are required for that adversary.

The E4 adapters share one simulator and differ at a bounded controller interface, so their results are limited to those variants and scenarios. E5 uses controlled synthetic version conditions instead of observed changes from a deployed model provider. Broader validation should include hardware-in-the-loop runs, independently implemented producers, compromised capture paths, sensor spoofing, delayed evidence, and profile-specific replay tolerances.

Finally, MADS-CPS is an evidence substrate. Regulatory compliance, safety-case completion, and claims of beneficial operation require domain hazards, human factors, lifecycle controls, and physical validation outside the contract.

8 Conclusion

MADS-CPS defines a machine-checkable run-level evaluation contract for agentic cyber-physical workflows. A declared envelope binds standardized artifacts to schema and linkage checks, independently recomputed commitments and reports, control-plane terminal-state reconstruction, relational PONR evidence, disclosure modes, and fail-closed release.

The hardened autonomous-laboratory thin slice achieves exact oracle agreement across 24 conformance cases, executable redaction semantics, implementation-independent report recomputation, complete localization of 110 replay mutations, and reproducible controller and synthetic version-shift studies over 1,200 runs. These results establish an inspectable substrate for structural run admissibility. Physical truth, continuous runtime control, and certification remain explicit obligations of the surrounding assurance system.

References

[1] Adam AlSayyad, Kelvin Yuxiang Huang, and Richik Pal. 2026. AgentTrace: A Structured Logging Framework for Agent System Observability. arXiv:2602.10133 [cs.SE] doi:10.48550/arXiv.2602.10133

[2] Chloe Autio, Reva Schwartz, Jesse Dunietz, Shomik Jain, Martin Stanley, Elham Tabassi, Patrick Hall, and Kamie Roberts. 2024. *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile.* Technical Report NIST AI 600-1. National Institute of Standards and Technology. doi:10.6028/NIST.AI.600-1

[3] Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. 2025. ScienceAgentBench: Toward Rigorous Assessment of Language Agents for Data-Driven Scientific Discovery. In *The Thirteenth International Conference on Learning Representations.* arXiv:2410.05080 doi:10.48550/arXiv.2410.05080

[4] Jon M. Laurent, Albert Bou, Michael Pieler, Conor Igoe, Alex Andonian, Siddharth Narayanan, James Braza, Alexandros Sanchez Vassopoulos, Jacob L. Steenwyk, Blake Lash, Andrew D. White, and Samuel G. Rodriques. 2026. LABBench2: An Improved Benchmark for AI Systems Performing Biology Research. arXiv:2604.09554 [cs.AI] doi:10.48550/arXiv.2604.09554

[5] Keyu Li, Junhao Shi, Yang Xiao, Mohan Jiang, Jie Sun, Yunze Wu, Shijie Xia, Xiaojie Cai, Tianze Xu, Weiye Si, Wenjie Li, Dequan Wang, and Pengfei Liu. 2026. AgencyBench: Benchmarking the Frontiers of Autonomous Agents in 1M-Token Real-World Contexts. arXiv:2601.11044 [cs.AI] doi:10.48550/arXiv.2601.11044

[6] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2024. AgentBench: Evaluating LLMs as Agents. In *The Twelfth International Conference on Learning Representations.* arXiv:2308.03688 doi:10.48550/arXiv.2308.03688

[7] Object Management Group. 2023. *Structured Assurance Case Metamodel (SACM), Version 2.3.* Technical Report formal/23-05-08. Object Management Group. https://www.omg.org/spec/SACM/2.3

[8] OpenTelemetry Authors. 2026. OpenTelemetry Semantic Conventions, Version 1.43.0. https://opentelemetry.io/docs/specs/semconv/ Generative AI conventions

maintained in the dedicated OpenTelemetry GenAI repository; accessed July 23, 2026.

[9] Ciprian Paduraru, Petru-Liviu Bouruc, and Alin Stefanescu. 2026. A Trace-Based Assurance Framework for Agentic AI Orchestration: Contracts, Testing, and Governance. arXiv:2603.18096 [cs.MA] doi:10.48550/arXiv.2603.18096

[10] Melissa Z. Pan, Negar Arabzadeh, Riccardo Cogo, Yuxuan Zhu, Alexander Xiong, Lakshya A. Agrawal, Huanzhi Mao, Emma Shen, Sid Pallerla, Liana Patel, Shu Liu, Tianneng Shi, Xiaoyuan Liu, Jared Quincy Davis, Emmanuele Lacavalla, Alessandro Basile, Shuyi Yang, Paul Castro, Daniel Kang, Joseph E. Gonzalez, Koushik Sen, Dawn Song, Ion Stoica, Matei Zaharia, and Marquita Ellis. 2026. Measuring Agents in Production. In *Proceedings of the 43rd International Conference on Machine Learning*. arXiv:2512.04123 [cs.CY] doi:10.48550/arXiv.2512.04123 Spotlight presentation.

[11] Jinhu Qi, Yifan Li, Minghao Zhao, Wentao Zhang, Zijian Zhang, Yaoman Li, and Irwin King. 2026. Beyond Benchmark Islands: Toward Representative Trustworthiness Evaluation for Agentic AI. arXiv:2603.14987 [cs.CL] doi:10.48550/arXiv.2603.14987

[12] Pablo Salazar-Villacis and Brahim Benyahia. 2026. The ADePT Framework for Assessing Autonomous Laboratory Robotics. *Communications Chemistry* 9, 1 (2026), 99. doi:10.1038/s42004-026-01932-9

[13] SLSA Community. 2026. SLSA Specification, Version 1.2: Provenance. https://slsa.dev/spec/v1.2/provenance Approved specification, accessed July 23, 2026.

[14] Elham Tabassi. 2023. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. Technical Report NIST AI 100-1. National Institute of Standards and Technology. doi:10.6028/NIST.AI.100-1

[15] Santiago Torres-Arias, Hammad Afzali, Trishank Karthik Kuppusamy, Reza Curtmola, and Justin Cappos. 2019. in-toto: Providing Farm-to-Table Guarantees for Bits and Bytes. In *28th USENIX Security Symposium (USENIX Security 19)*. USENIX Association, Santa Clara, CA, 1393–1410. https://www.usenix.org/conference/usenixsecurity19/presentation/torres-arias

[16] UL Standards & Engagement. 2023. UL 4600: Standard for Evaluation of Autonomous Products. https://ulse.org/ul-4600/ Third edition, March 17, 2023.

[17] Chenglin Yang. 2026. AgentTrust: Runtime Safety Evaluation and Interception for AI Agent Tool Use. arXiv:2605.04785 [cs.CR] doi:10.48550/arXiv.2605.04785

[18] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2025. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. In *The Thirteenth International Conference on Learning Representations*. doi:10.48550/arXiv.2406.12045

[19] Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. 2025. Survey on Evaluation of LLM-based Agents. arXiv:2503.16416 [cs.AI] doi:10.48550/arXiv.2503.16416

[20] Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *The Twelfth International Conference on Learning Representations*. doi:10.48550/arXiv.2307.13854

[21] Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun, Andy Zhang, Shu Liu, Sasha Cui, Sayash Kapoor, Shayne Longpre, Kevin Meng, Rebecca Weiss, Fazl Barez, Rahul Gupta, Jwala Dhamala, Jacob Merizian, Mario Giulianelli, Harry Coppock, Cozmin Ududec, Jasjeet Sekhon, Jacob Steinhardt, Antony Kellermann, Sarah Schwettmann, Matei Zaharia, Ion Stoica, Percy Liang, and Daniel Kang. 2025. Establishing Best Practices for Building Rigorous Agentic Benchmarks. arXiv:2507.02825 [cs.AI] doi:10.48550/arXiv.2507.02825