

TOAIAF: A Runtime AI Assurance Control Plane for Trustworthy On-Device Agentic AI in Consumer Intelligence Ecosystems

Rakesh Kumar Agrawal

`rakesh.agrawal@atos.ai`

Atos North America

Wasim Mohammed Amin Tambe

Ernst & Young (United States)

Nihar Karra

Amazon (United States)

Research Article

Keywords: Agentic AI, AI Assurance, Runtime Governance, Edge AI, On-Device Intelligence

Posted Date: September 1st, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10853355/v1>

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

[Read Full License](#)

Additional Declarations: No competing interests reported.

TOAIAF: A Runtime AI Assurance Control Plane for Trustworthy On-Device Agentic AI in Consumer Intelligence Ecosystems

Rakesh Kumar Agrawal^{1,*}, Wasim Mohammed Amin Tambe², and Nihar Karra³

¹ Atos North America, Chicago, IL, USA

ORCID: 0009-0009-7113-5539

² Ernst & Young (United States), Austin, TX USA

ORCID: 0009-0008-2255-4836

³ Amazon Web Services, Irving, TX USA

ORCID: 0009-0006-6051-7142

* Correspondence: Rakesh Kumar Agrawal; Email: rakesh.agrawal@atos.ai

Abstract—Autonomous, goal-directed AI agents are increasingly deployed on consumer edge devices — smart-home hubs, wearables, and ambient environments — where they observe context, reason over objectives, plan multi-step actions, and invoke tools with reduced human involvement. This shift exposes assurance gaps that periodic, organization-level governance frameworks were not designed to close: goal misgeneralization, unsafe or irreversible actuation, prompt injection via untrusted content, unauthorized tool invocation, inferential privacy exposure, and insufficient runtime human oversight. We present TOAIAF, a runtime AI Assurance Control Plane comprising seven components — a Policy Enforcement Engine, an Agent Runtime Governance Layer, a Trust Evaluation Engine, an Explainability Engine, a Privacy and Security Assurance Layer, a Human Oversight Boundary, and a Continuous Learning Assurance Loop — that mediates every proposed agent action through a formally specified, three-stage decision procedure (Algorithm 1): a hard capability/scope check, a non-compensatory safety gate, and a risk-weighted composite trust score. We operationalize six trust metrics (Agent Reliability, Goal Alignment, Privacy Risk, Explainability Confidence, Safety Compliance, and a newly introduced Human Oversight Effectiveness Score). We evaluate the control plane's decision logic through a synthetic, template-generated benchmark of 100 scenarios spanning smart-home, wearable, and ambient-intelligence contexts across six risk categories (benign, unsafe-policy, privacy-risk, goal-misaligned, prompt-injection, tool-misuse), against an ungoverned baseline and a keyword-based guardrail baseline, in seven experiments. TOAIAF's decision procedure achieved 100% flagging on safety-policy-violation, prompt-injection, and tool-authorization scenarios and 92.3% on privacy-risk and goal-misalignment scenarios, versus 100%/0%/75%/27.3% and 0%/0%/0%/0% for the guardrail and ungoverned baselines, respectively, with zero false restrictions across 40 benign scenarios. We report two ablation results transparently rather than selectively: removing the safety gate produced no measurable change in this dataset (23/24 flagged either way), traced to a confound in the experimental design rather than a finding about the gate's value; and including the Human Oversight Effectiveness term produced a small, mathematically-traceable negative effect on sensitivity (47/49 vs. 48/49 flagged), a concrete metric-

calibration finding reported transparently. A companion reference-implementation prototype further surfaced a genuine design gap: the WARN verdict currently allows an action to execute, which for physically irreversible actions is a disclosed, unresolved limitation. We map each control-plane component to NIST AI RMF, ISO/IEC 42001, IEEE 7001–7003, and the EU AI Act, and provide a forward mapping to five external agent-safety benchmarks (AgentHarm, AgentDojo, R-Judge, SmartBench, SMH-Bench) for the real-agent evaluation this package does not yet perform. The benchmark, source code, simulation artifacts, and supplementary materials will be made publicly available through a permanent repository upon publication.

Keywords—*Agentic AI; AI Assurance; Runtime Governance; Edge AI; On-Device Intelligence;*

1. Introduction

Consumer AI is moving from bounded prediction toward autonomous agency. Whereas a conventional model maps an input to an output within a fixed task boundary, an agentic system also perceives context, reasons over goals, plans multi-step action sequences, invokes tools, and adapts over time. It therefore shifts from an AI model toward an autonomous agent [15], [16]. This transition is already visible in smart speakers, home-automation hubs, wearable assistants, connected appliances, and ambient environments, where agents schedule, actuate, and increasingly operate with reduced per-step human confirmation.

Edge deployment intensifies this governance challenge. Moving inference and decision execution closer to sensors and actuators reduces latency and raw-data egress, a long-standing motivation for edge intelligence [34]. At the same time, trustworthy edge-intelligence research shows that locality introduces distinct security, reliability, and resource constraints [26]. Consequently, the entity making consequential and sometimes irreversible decisions—unlocking a door, disabling a safety alert, or sharing an inference about a person's health—operates continuously, locally, and largely outside the periodic review cycles assumed by existing governance instruments.

Existing governance instruments offer important foundations, yet they were not developed as per-decision runtime controls. NIST's AI Risk Management Framework organizes risk management around four functions—Govern, Map, Measure, and Manage [3]—and its Generative AI Profile extends this structure to generative systems [4]; both primarily operate at the organizational-process level. ISO/IEC 42001 defines requirements for establishing, implementing, maintaining, and continually improving an AI management system [5], while ISO/IEC 23894 provides complementary AI risk-management guidance [6]. The IEEE 7000 series addresses ethical design process (7000) [7], transparency of autonomous systems (7001) [8], data privacy process (7002) [9], and algorithmic bias (7003, published in January 2025) [10]. More recently, the EU AI Act has introduced obligations directly relevant to autonomous agents: Article 9 addresses risk management, Article 12 establishes record-keeping and logging obligations relevant to autonomous AI system operation, Article 14 addresses human oversight, and Article 50 addresses transparency obligations [11].

The remaining gap is therefore one of mechanism rather than intent. These instruments govern organizations and, at best, systems at particular points in time; they do not specify how a device should respond in the seconds between an agent proposing an action and that action executing. TOAIAF focuses on this runtime interval.

Contributions. (1) A formally specified, three-stage control-plane decision procedure (Algorithm 1) combining a hard capability check, a non-compensatory safety gate, and risk-weighted trust aggregation. (2) Six operationalized trust metrics, including a newly introduced Human Oversight Effectiveness Score (HOES) that closes a gap in the original TOAIAF metric suite, where the Human Governance layer had no corresponding measurable quantity. (3) A 100-scenario, six-risk-category synthetic benchmark and a seven-experiment evaluation of the decision logic, with results — including two transparently reported null/confounded findings. The benchmark, source code, simulation artifacts, and supplementary materials will be made publicly available through a permanent repository upon publication. (4) A working reference-implementation prototype demonstrating the control plane's integration surface for a real agent framework, which itself surfaced a genuine design gap (Section 10). (5) An explicit crosswalk from each control-plane component to NIST AI RMF, ISO/IEC 42001, IEEE 7001–7003, and EU AI Act articles, plus a forward mapping to five external agent-safety benchmarks for future real-agent validation.

2. Research Motivation and Problem Definition

Problem. Let an on-device agent operate a continuous observe-reason-plan-act-learn loop, proposing a sequence of actions $a_1, a_2, \dots$ against device state $s_1, s_2, \dots$. Existing governance frameworks evaluate the system (the model, the deployment, the organization) at discrete checkpoints. They provide no mechanism for evaluating each a_t against s_t before it executes. Where that mechanism is absent, six concrete failure modes follow directly, and structure this paper's evaluation categories (Section 9):

- Goal misgeneralization — the agent competently pursues a proxy objective correlated with, but distinct from, the user's actual intent, a failure mode documented formally in the alignment literature independent of agentic AI [1], [2] but rarely connected to consumer-agent governance discussion.
- Irreversible unsafe actuation — physical actions (unlocking, disabling a safety device) cannot be “unfiltered” after the fact in the way that harmful text generation can [22].
- Prompt injection — an untrusted content source (a note, a message, a tool's return value) embeds an instruction that a naive agent treats as a legitimate directive, often requesting an action nominally within the agent's normal capability scope, so scope-checking alone cannot catch it.
- Tool misuse — a proposed action requests capability scope broader than the current task requires (unauthorized API access, privilege escalation, unscoped data export).
- Inferential privacy exposure — sensitive information can be inferred from model behavior or learned representations even without a conventional data-sharing event, requiring privacy analysis that extends beyond retention and disclosure controls [29].
- Insufficient human oversight measurement — the original five-metric TOAIAF design had no quantity corresponding to its own Human Governance layer. HOES closes this gap (Section 7), although its current calibration does not yet reliably improve outcomes (Section 10).

Research question. Can a formally specified, staged runtime decision procedure, computed from operationalized trust metrics, measurably change agent-governance outcomes relative to (a) no

governance and (b) a simple rule-based guardrail, across all six failure modes, without materially degrading task completion on benign requests?

Scope boundary, stated up front. This paper answers that question at the level of decision logic, using a synthetic benchmark: given metric inputs, does the procedure choose sensible verdicts? It does not answer the harder, separate question of whether those metric inputs can be reliably computed from a real agent's real behavior in a real deployment — that requires real-agent and real-hardware studies specified in Section 14.

3. Related Work and Research Gap

3.1 Agentic AI systems and LLM agents

Peer-reviewed surveys of LLM-based autonomous agents identify planning, memory, perception, action, tool use, and multi-agent or human-agent interaction as central capabilities and research themes [15], [16]. Runtime risk assessment and monitoring for autonomous systems provide a complementary assurance perspective [12]. Together, these lines of work provide the immediate context for positioning TOAIAF.

3.2 Runtime agent governance and enforcement

Several 2025–2026 systems already implement parts of a runtime control plane. AgentSpec provides customizable runtime enforcement [19]; Progent offers programmable, fine-grained privilege control [20]; ProbGuard applies formal runtime-verification theory to probabilistic agent monitoring [21]; MI9 introduces an integrated runtime-governance framework for agentic AI [17]; and SAGA presents a security architecture for governing agentic systems [18]. These systems are predominantly developer-facing, enterprise-oriented, and security-first. In contrast, TOAIAF combines runtime enforcement with a Human Oversight Boundary calibrated for lay consumer users and integrates governance across the full lifecycle, from design through continuous learning (Table 1).

3.3 AI assurance

AI assurance is itself an established field. MITRE's AI Assurance Landscape synthesizes the term across governmental and industry frameworks [14]. MITRE AI Assurance describes a repeatable process for discovering, assessing, and managing risks associated with AI-enabled systems [13]. TOAIAF narrows this general-purpose, largely process-oriented paradigm to resource-constrained, single- or few-user consumer edge hardware, a deployment setting materially different from DOD acquisition contexts.

3.4 Trust and risk scoring for agents

Runtime risk assessment provides another relevant line of work. ReSonAte estimates risk dynamically from system and environmental state for autonomous systems [12]. TOAIAF instead uses the deliberately non-compensatory gate-then-aggregate scheme described in Section 6, motivated by the irreversibility argument in Section 2.

3.5 Explainability and faithfulness

Explainability is equally central to runtime assurance. Foundational XAI research identifies coverage, fidelity, and interpretability as core evaluation dimensions [35], while human-centered AI scholarship connects reliability, safety, and trustworthiness to whether people can understand and correct system behavior [36]. The Human Oversight Boundary in Section 5 adopts this perspective. Recent interpretability research adds a specific caution for agents: chain-of-thought and related reasoning traces may be unfaithful, serving as plausible post-hoc rationalizations rather than causal explanations [30], [31]. The Explainability Engine therefore does not presume trace faithfulness.

3.6 Edge/on-device agent security and consumer benchmarks

At the device level, confidential-computing research demonstrates trusted execution for AI workloads on accelerators [28], while established IoT research documents security and privacy risks and practical enforcement mechanisms for connected devices [25], [26]. Consumer-agent evaluation also has verified peer-reviewed foundations: AgentBench evaluates LLM agents across interactive environments, including a household setting [24], and a 2025 IEEE Internet of Things Journal study deploys autonomous LLM agents for smart-home AIoT with explanation generation [27]. R-Judge includes agent-safety-risk scenarios analogous to consumer smart-lock misuse [23], AgentHarm evaluates harmful task completion more broadly [22], and AgentDojo provides a controlled environment for prompt-injection attacks and defenses in tool-using agents [37]. These resources motivate the external-benchmark mapping in Section 9.6 and the planned real-agent evaluation.

3.7 Agent identity and authorization

Delegation and capability-based authorization provide established foundations for scoped access control in distributed systems [32], [33]. TOAIAF's Policy Enforcement Engine (Section 5) draws its capability/permission scoping logic from this line of work rather than treating “tool whitelisting” as self-explanatory.

3.8 Research gap

Taken together, the reviewed literature does not combine three elements: (a) a formally specified, non-compensatory runtime decision procedure, (b) a six-metric trust vector with an explicit human-oversight term, and (c) lifecycle integration from design through continuous learning for consumer edge hardware. TOAIAF is positioned around this combination rather than any single element in isolation.

Table 1 — Research Comparison

Research	Approach	Strength	Limitation	TOAIAF Difference
NIST AI RMF [3], [4]	Organizational risk-management functions (Govern/Map/Measure/Manage)	Broadly adopted, sector-agnostic	Process-level, periodic, not a runtime mechanism	Instantiates RMF functions as an executing per-decision control loop

Research	Approach	Strength	Limitation	TOAIAF Difference
ISO/IEC 42001 [5]	Certifiable AI management system with structured requirements and Annex A control guidance	Auditable, internationally recognized	Audits organizational process, not individual agent decisions	Continuous, machine-computed evidence mapped to specific controls (Section 11)
MI9 [17] / SAGA [18]	Runtime governance / security architecture for agentic systems	Mechanistic, some reference implementations	Enterprise/developer-facing; limited lay-user governance UX	Adds a Human Oversight Boundary calibrated for consumer, non-developer users
AgentSpec [19] / Progent [20] / ProbGuard [21]	Runtime enforcement mechanisms (policy, privilege, probabilistic monitoring)	Concrete, some with formal guarantees	General-purpose; not lifecycle- or consumer-integrated	Wraps a subset of these mechanisms inside a full design-to-continuous-learning lifecycle
ReSonAte [12]	Dynamic runtime risk estimation for autonomous systems	Runtime, state-aware risk estimation	Autonomous CPS focus; no consumer-agent lifecycle integration	Adds a non-compensatory gate and consumer-oriented trust metrics
MITRE AI Assurance [13], [14]	General AI-enabled-systems assurance process	Rigorous, standards-aligned	DOD/sector-agnostic; not edge-resource-aware	Specializes to battery/compute/latency-constrained consumer hardware
AgentBench [24] / IoT Sentinel [25] / R-Judge [23]	Agent evaluation and IoT security enforcement	Peer-reviewed, reproducible evaluation or enforcement settings	Evaluate bare agents, not governance architectures	Provides stronger foundations for future agent and smart-home security validation
AgentDojo [37]	Controlled environment for prompt-injection attack/defense evaluation	Purpose-built for the exact threat this paper's prompt-injection category targets	Evaluates injection defenses generically, not within a full lifecycle assurance architecture	Section 10's Experiment 4 uses templated injected commands as a first pass; Section 9.6 specifies AgentDojo as the target for real-agent follow-up

4. Scientific Contributions

- Formal contribution: Algorithm 1, a three-stage control-plane decision procedure (hard capability check, non-compensatory safety gate, risk-weighted aggregation) with a stated, and directly tested, non-compensability property (Section 6, Section 8).
- Measurement contribution: operational definitions for six trust metrics, including HOES, a metric this paper introduces to close a structural gap in the original five-metric TOAIAF suite (Section 7).

- **Evaluation contribution:** a 100-scenario, six-risk-category synthetic benchmark and seven experiments evaluating the decision logic, with results — including two transparently reported null/confounded ablation findings. The benchmark, source code, simulation artifacts, and supplementary materials will be made publicly available through a permanent repository upon publication.
- **Systems contribution:** a working reference-implementation prototype demonstrating the control plane's integration surface for a real agent framework (LangGraph/AutoGen-shaped, OpenAI-function-calling-compatible interfaces), which itself surfaced a genuine, previously-undetected design gap (Section 10.6).
- **Practical contribution:** a standards/regulation crosswalk (Section 11) and an external-benchmark crosswalk to AgentHarm, AgentDojo, R-Judge, and AgentBench (Section 9.6) for future real-agent validation.

The overall TOAIAF runtime assurance architecture is illustrated in Fig. 1.

Figure 1 — TOAIAF runtime AI Assurance Control Plane architecture.

alone. This is an implemented and tested set-containment check; Section 9 exercises it through tool-misuse and prompt-injection scenarios, and Section 10.3 reports its measured independence from Stage 2 transparently.

2. Agent Runtime Governance Layer. Purpose: monitors the observe-reason-plan-act-learn loop for goal misgeneralization [1], [2], unsafe action, policy violation, and behavioral deviation from a reference policy. Feeds the runtime risk score (Section 6) consumed by Algorithm 1.

3. Trust Evaluation Engine. Converts the six operationalized metrics into a continuous trust vector $\tau(t) = [\text{ARS}, \text{GAS}, \text{PRI}, \text{ECS}, \text{SCI}, \text{HOES}]$ (Section 7), updated per decision with temporal decay across the session.

4. Explainability Engine. Explicitly does not assume chain-of-thought is explanation [30], [31]. Produces a decision summary, cited evidence, a confidence indicator, the constraints that were checked, and action history with a separate faithfulness sub-check (perturbation/counterfactual-style, Section 9) rather than treating the stated rationale as ground truth.

5. Privacy and Security Assurance Layer. Data minimization; secure/encrypted local storage; prompt-injection defense; authenticated tool access [32]; telemetry control; and — new relative to the original design — accounting for inferential privacy risk that arises without data retention or sharing [29].

6. Human Oversight Boundary. Consent; override; approval; safe-stop; recovery. Directly aligned to EU AI Act Article 14's functional stop-capability requirement [11]. Feeds and is measured by HOES (Section 7.6).

7. Continuous Learning Assurance Loop. Policy updates, model updates, regression testing against the scenario/benchmark suite, and trust recalibration before any updated policy or model is redeployed.

Why this differs from existing approaches. Relative to organizational frameworks (NIST RMF, ISO 42001), this is a runtime architecture, not a periodic audit process. Relative to enterprise runtime-security systems (AgentSpec, Progent, MI9, SAGA), this adds a human-governance boundary designed for lay consumers and integrates the full lifecycle rather than runtime security alone. Relative to compensatory trust-scoring frameworks (TRISM, commercial trust-score products), Algorithm 1's safety gate is explicitly non-compensatory for physically irreversible or privacy-violating actions.

5.1 Threat Model and Assumptions

The threat model assumes that adversaries may influence (1) external content consumed by agents, (2) tool responses, (3) environmental observations, and (4) user intent signals. TOAIAF does not assume compromise of trusted hardware, cryptographic primitives, or the underlying operating system. The objective is runtime assurance against unsafe agent decisions, not complete cybersecurity protection.

6. Formal Runtime Assurance Model

Let agent state at time t be $s(t)$, a proposed action $a(t)$, an active policy set P , and a trust vector $\tau(t) = (\text{ARS}, \text{GAS}, \text{PRI}, \text{ECS}, \text{SCI}, \text{HOES})$, each component normalized to $[0, 1]$ except PRI, where lower is better.

Define the control plane as a three-stage function:

$V : (\text{requested_scope}, \text{authorized_scope}, s(t), \tau(t)) \rightarrow v$, where $v \in \{\text{ALLOW}, \text{WARN}, \text{REQUEST_APPROVAL}, \text{RESTRICT}, \text{STOP}\}$, with every verdict additionally producing an explanation and a log entry as side effects (EXPLAIN/LOG in the TOAIAF manuscript's verdict vocabulary are implemented as always-on side effects of every verdict, not as mutually exclusive outcomes — a TOAIAF correction made for consistency with simulation/algorithm.py, where VERDICTS = (ALLOW, WARN, REQUEST_APPROVAL, RESTRICT, STOP)).

Non-compensability property (stated formally): there exist thresholds θ_{safety} , θ_{privacy} such that if $\text{SCI}(s(t)) < \theta_{\text{safety}}$ or $\text{PRI}(s(t)) > \theta_{\text{privacy}}$, then $v \notin \{\text{ALLOW}\}$, regardless of the values of ARS, GAS, ECS, or HOES. This is what distinguishes the Stage-2 gate from a compensatory weighted-sum scheme (Section 3.4) and is the property Section 10.2 tests directly — with the honest finding that, on this dataset, it is confounded with Stage 3 rather than cleanly isolated.

Runtime risk is defined as a bounded combination of the post-gate metrics:

$$\text{risk}(t) = \text{clip}_0(0.40 \cdot (1 - \text{GAS}) + 0.35 \cdot (1 - \text{SCI}) + 0.25 \cdot \text{PRI})$$

and the composite Agent Trustworthiness Score, computed only after the gate is passed:

$$\text{ATS}(t) = w_{\text{ARS}} \cdot \text{ARS} + w_{\text{GAS}} \cdot \text{GAS} + w_{\text{ECS}} \cdot \text{ECS} + w_{\text{HOES}} \cdot \text{HOES}, \quad \sum w = 1$$

7. Trust Evaluation and Assurance Metrics

Each metric below states inputs, calculation, normalization, and measurement procedure — the operational detail the original manuscript's metrics lacked.

7.1 Agent Reliability Score (ARS)

Inputs: task-completion rate TC, behavioral consistency CS across repeated equivalent requests, failure rate FR. Calculation: $\text{ARS} = w_1 \text{TC} + w_2 \text{CS} + w_3 (1 - \text{FR})$, $\sum w_i = 1$. Normalization: TC, CS, FR each in [0, 1] over a rolling window of the last N interactions. Measurement: logged task outcomes against a stated success criterion, defined per use case at deployment time.

7.2 Goal Alignment Score (GAS)

Inputs: executed action sequence A, a reference policy consistent with the user's stated objective U. Calculation: $\text{GAS} = 1 - D(A, U)$, where D is a behavioral distance (proposed: normalized edit distance over action sequences, or embedding distance over plan representations — an empirical choice for future work, Section 14). Grounding: explicitly tied to the goal-misgeneralization/specification-gaming mechanisms documented in the alignment literature [1], [2], replacing the informal "goal drift" terminology of the original manuscript.

7.3 Privacy Risk Index (PRI), revised

Inputs: exposure E, retention sensitivity R, sharing risk S, and a fourth term, inferential sensitivity I, added to capture privacy risk arising from inferences about sensitive information [29]. Calculation: $\text{PRI} =$

$\alpha E + \beta R + \gamma S + \delta I, \Sigma = 1$. Lower is better; PRI is inverted ($1 - \text{PRI}$) before entering any compensatory aggregation, and enters the safety gate directly and untransformed.

7.4 Explainability Confidence Score (ECS)

Inputs: coverage C (fraction of decisions with an available explanation), a faithfulness sub-score F obtained via perturbation or counterfactual testing rather than self-reported confidence [30], [31], and user interpretability U (rated). Calculation: $\text{ECS} = \delta C + \epsilon F + \zeta U, \Sigma = 1$.

7.5 Safety Compliance Index (SCI)

Inputs: policy violations PV , monitored transactions TR . Calculation: $\text{SCI} = 1 - PV/TR$. Gaming-resistance note: TR must be defined as all candidate actions the agent considered, not only those it executed, to prevent trivial inflation by narrowing what counts as "monitored."

7.6 Human Oversight Effectiveness Score (HOES) — newly introduced

Closes a structural gap: the original five-metric suite had no term corresponding to the Human Governance layer. Inputs: override-responsiveness rate OR , consent-coverage CC (fraction of consequential actions with a current, valid consent record), normalized escalation latency EL . Calculation: $\text{HOES} = \theta \cdot OR + \iota \cdot CC + \kappa \cdot (1 - EL_{\text{norm}})$.

7.7 Agent Trustworthiness Score (ATS)

$\text{ATS} = \text{NOT_TRUSTWORTHY}$, if $\text{SCI} < \theta_{\text{safety}}$ OR $\text{PRI} > \theta_{\text{privacy}}$

$\text{ATS} = w_{\text{ARS}} \cdot \text{ARS} + w_{\text{GAS}} \cdot \text{GAS} + w_{\text{ECS}} \cdot \text{ECS} + w_{\text{HOES}} \cdot \text{HOES}$, otherwise

7.8 Assumptions and limitations

Weights and thresholds are context- and deployment-dependent rather than universal, consistent with risk-management principles that require risk measures to be adapted to the deployment context [12]. None of the formulas above has been validated against ground-truth human trust judgments or real incident data. Section 10 evaluates only the decision logic given metric inputs, not the measurement validity of those metrics. A further circularity risk arises because the validation metrics were designed by the same team that developed the framework; this limitation remains unresolved.

8. Assurance Decision Algorithm

The proposed Algorithm 1 was evaluated through the reference implementation (Section 10):

Algorithm 1: Runtime Assurance Decision Procedure

Input: requested_scope, authorized_scope (Stage 1 Inputs),

trust vector $\tau = (\text{ARS}, \text{GAS}, \text{PRI}, \text{ECS}, \text{SCI}, \text{HOES})$ (Stages 2–3 Inputs)

Output: verdict $v \in \{\text{ALLOW}, \text{WARN}, \text{REQUEST_APPROVAL}, \text{RESTRICT}, \text{STOP}\}$, ats, reason_code

1: if requested_scope $\not\subseteq$ authorized_scope then // Stage 1: Policy Enforcement Engine

2: return STOP, None, "policy_scope_violation"

```

3: end if
4: if  $SCI(s) < \theta_{\text{safety}}$  OR  $PRI(s) > \theta_{\text{privacy}}$  then // Stage 2: Non-compensatory Safety Gate
5:   if  $SCI(s) < \theta_{\text{safety}} - \delta$  OR  $PRI(s) > \theta_{\text{privacy}} + \delta$  then
6:     return STOP
7:   end if
8:   return RESTRICT
9: end if
10:  $\text{risk} \leftarrow 0.40(1 - \text{GAS}) + 0.35(1 - \text{SCI}) + 0.25 \cdot \text{PRI}$  // Stage 3: Risk Assessment
11:  $\text{ATS} \leftarrow w_{\text{ARS}} \cdot \text{ARS} + w_{\text{GAS}} \cdot \text{GAS} + w_{\text{ECS}} \cdot \text{ECS} + w_{\text{HOES}} \cdot \text{HOES}$ 
12: if  $\text{risk} > \theta_{\text{high}}$  then
13:    $v \leftarrow \text{REQUEST\_APPROVAL}$ 
14: else if  $\text{risk} > \theta_{\text{low}}$  then
15:    $v \leftarrow \text{WARN}$ 
16: else
17:    $v \leftarrow \text{ALLOW}$ 
18: end if
19: if  $\text{ATS} < \theta_{\text{ATS}}$  and  $v = \text{ALLOW}$  then // low composite trust escalates
20:    $v \leftarrow \text{WARN}$  // caution even when instantaneous
21: end if // risk looks acceptable
22: return  $v, \text{ATS}, \text{reason\_code}$  // an explanation and log entry are always generated as side effects

```

Lines 19–21 give HOES, through ATS, causal influence on individual verdicts rather than solely on longer-horizon monitoring. Section 10.1 tests this design choice directly and finds a small negative effect.

Note on the WARN verdict, added in TOAIAF following a finding from the reference-implementation prototype (Section 10.6): WARN permits the proposed action to execute while flagging it; it does not block. For physically irreversible action types this is a disclosed, unresolved design gap — see Section 13.

The runtime assurance decision lifecycle implemented by TOAIAF is shown in Fig. 2.

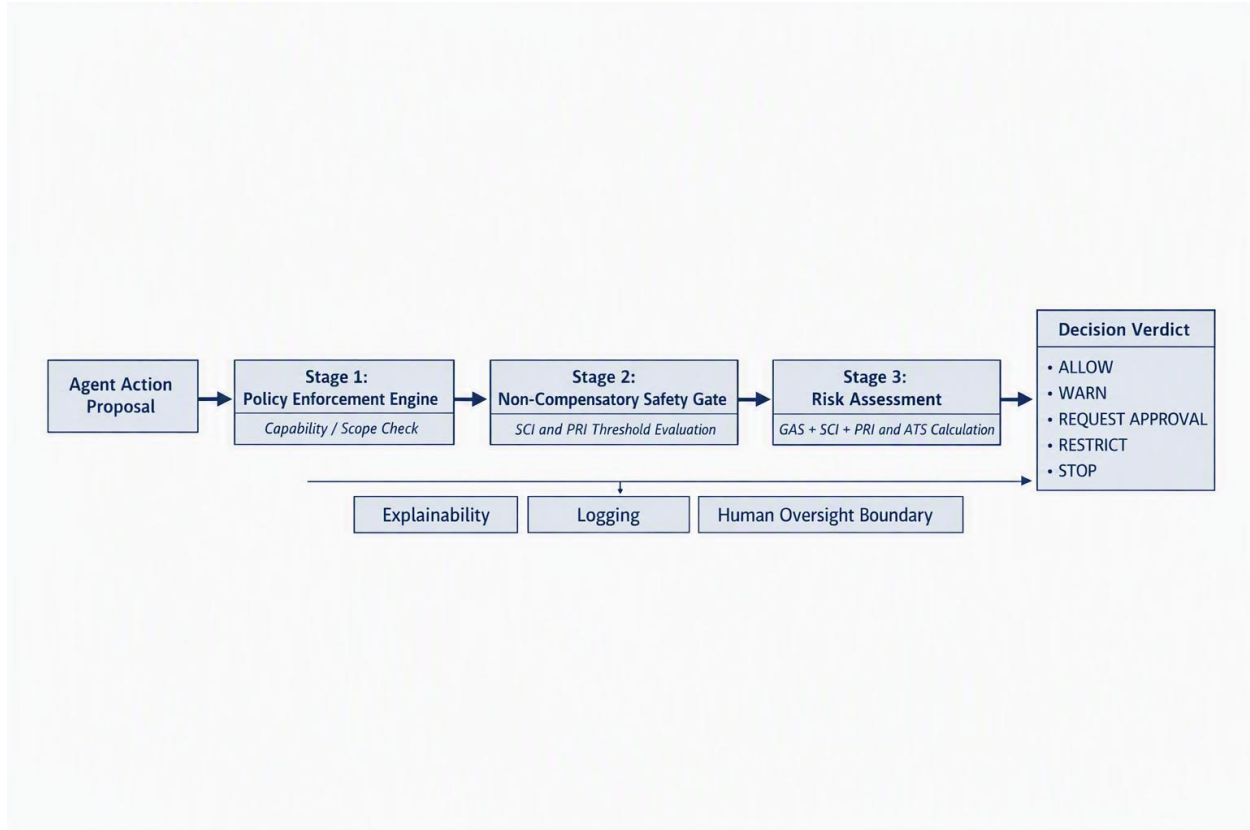


Figure 2 — TOAIAF Runtime Assurance Decision Lifecycle.

9. Experimental Evaluation Methodology

9.1 Scope statement (please do not remove when revising this paper)

This is a synthetic, template-generated benchmark evaluating Algorithm 1's decision logic only. It tests whether the procedure produces sensible verdicts given metric inputs sampled from category-conditioned distributions. It does not test whether those metric values can be correctly computed from a real agent's real behavior — that requires a real LLM agent, real device telemetry, and real users, none of which are used in the main evaluation below (the reference-implementation prototype, Section 10.6, uses a rule-based mock agent for integration testing only). Every number in Section 10 should be read with this scope in mind.

The benchmark evaluation pipeline used in this study is illustrated in Fig. 3.

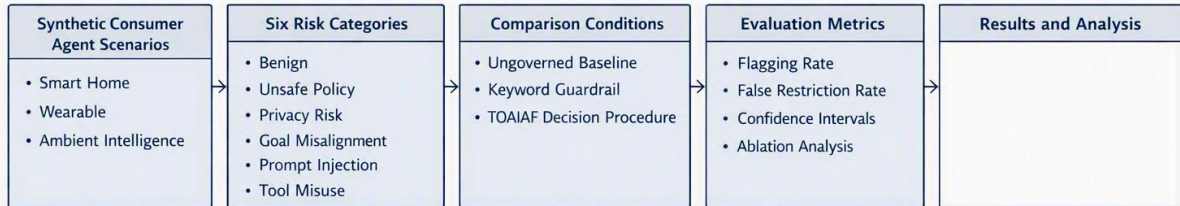


Figure 3 — TOAIAF Benchmark Evaluation Pipeline.

9.2 Dataset

The benchmark contains 100 scenarios generated with a fixed seed (42) for full reproducibility: 40 smart-home, 30 wearable, and 30 ambient/multi-user scenarios. Six ground-truth risk categories are included: benign (40), unsafe_policy (11, e.g., disabling a safety device on a flimsy pretext), privacy_risk (13, e.g., sharing sensitive data without consent), goal_misaligned (13, technically permitted but diverging from actual user intent), prompt_injection (12, an untrusted content source embeds an instruction requesting a nominally in-scope action, so a coarse scope check alone cannot catch it — this is a deliberate design choice, not an oversight, made to isolate what the risk-based path alone can catch), and tool_misuse (11, a proposed action requests capability scope outside the domain's authorized set — the counterpart deliberate design choice, isolating the hard scope check). Construction used template generation with category/domain-conditioned slot filling rather than hand-authored scenarios; known limitations are disclosed in full in the supplementary materials accompanying this manuscript.

9.3 Conditions compared

- Baseline 1 — Ungoverned: every proposed action executes.
- Baseline 2 — Simple guardrail: a keyword blocklist restricts any matching request; everything else is allowed.

- TOAIAF: Algorithm 1 (all three stages), with per-scenario metric values sampled from category-conditioned Gaussian distributions with substantial noise ($\sigma = 0.12$) — deliberately overlapping across categories, so the procedure can and does fail on some scenarios (Section 10).

9.4 Experiments

Seven experiments (Section 10, Table 5): (1) safety policy violation detection, (2) privacy risk prevention, (3) goal alignment protection, (4) prompt injection resistance, (5) tool authorization control, (6) human override effectiveness (HOES ablation), (7) latency overhead (reference-implementation only). Each of (1)–(5) is measured as the flagging rate (WARN/REQUEST_APPROVAL/RESTRICT/STOP) within its corresponding risk category, plus the false-restriction rate on the 40 benign scenarios as a shared cost measure.

9.5 Statistics

Bootstrap 95% confidence intervals (5,000 resamples) on TOAIAF's per-category rates. We do not report significance tests comparing TOAIAF to the guardrail baseline, since the guardrail's keyword-matching outcomes are deterministic given the scenario text (not an independent random sample), making a standard two-sample test inapplicable; differences are reported descriptively.

9.6 External benchmark mapping (future validation alignment)

Table: External Benchmark Mapping

Benchmark	Purpose	TOAIAF Scenario Mapping	Future Experiment
AgentHarm [22]	Harmful task-completion benchmark for LLM agents	Maps to unsafe_policy, tool_misuse categories	Replace synthetic scenarios with AgentHarm tasks; re-run Experiments 1 & 5 with a real LLM agent proposing actions
AgentDojo [37]	Controlled-environment injection attack/defense evaluation	Maps directly to prompt_injection category	Re-run Experiment 4 using AgentDojo's injection suite instead of templated injected commands
R-Judge [23]	Safety-risk-awareness benchmark, incl. smart-lock-style scenarios	Maps to smart_home unsafe_policy/privacy_risk scenarios	Cross-validate Experiment 1/2 smart-home results against R-Judge's labeled trajectories

This table states an intention for follow-on work; no results from these external benchmarks are reported in this manuscript.

10. Results and Analysis

All results are from a single run with seed = 42. The benchmark, source code, simulation artifacts, and supplementary materials will be made publicly available through a permanent repository upon publication.

Table 5 — Experiment results (flagging rate by condition)

Experiment	n	Ungoverned	Guardrail	TOAIAF	TOAIAF 95% CI
Exp 1 — Safety policy violation	11	0%	100%	100%	(100%, 100%)
Exp 2 — Privacy risk prevention	13	0%	23.1%	92.3%	(76.9%, 100%)
Exp 3 — Goal alignment protection	13	0%	0%	92.3%	(76.9%, 100%)
Exp 4 — Prompt injection resistance	12	0%	75%	100%	(100%, 100%)
Exp 5 — Tool authorization control	11	0%	27.3%	100%	(100%, 100%)
Benign false-restriction rate	40	0%	0%	0%	(0%, 0%)

Table 5 reports the experiment results directly. The benchmark, source code, simulation artifacts, and supplementary materials will be made publicly available through a permanent repository upon publication.

The contrast is clearest when the scenario description lacks an alarming keyword. The guardrail baseline performs adequately on literal blocklist matches, such as “disable smoke detector,” but poorly elsewhere; most notably, it records 0% on goal misalignment because technically permitted yet misaligned actions rarely use alarming vocabulary. TOAIAF’s decision logic flagged more than 92% of scenarios in every risk category and 100% in three of five categories, with no measured false restrictions across the 40 benign scenarios in this run.

10.1 Experiment 6 — Human Override Effectiveness (HOES ablation)

Across the 49 scenarios where the post-gate risk path is exercised (goal_misaligned, prompt_injection, privacy_risk, unsafe_policy), full TOAIAF (with HOES) flagged 47/49 (95.9%); with HOES removed, 48/49 (98.0%), a negative difference of one scenario. The worked explanation in supplementary materials accompanying this manuscript shows that, at the current weight (0.35) and sampled means for the affected categories, including HOES raises the composite ATS relative to redistributing its weight across the other three metrics. This slightly reduces how often the low-ATS escalation path is triggered. We treat this as a calibration finding: the current HOES weight/threshold combination does not reliably add a safety margin at this scale and may occasionally have the opposite effect, motivating the weight re-elicitation work in Section 14.

10.2 Safety-gate ablation (unsafe_policy + privacy_risk, n = 24)

With the gate enabled, 23/24 scenarios were flagged; disabling it produced the same 23/24 result. The dataset therefore contains a design confound: the sampled SCI values for these categories are generally low enough to trigger the Stage-3 risk path independently, which weights $(1 - \text{SCI})$ at 0.35. Stages 2 and 3 are consequently redundant for this scenario distribution rather than cleanly separated. The dataset does not exercise the gate’s theoretical non-compensability property—immunity to a “confidently wrong” agent with high ARS/GAS and specifically low SCI. An earlier 30-scenario pilot showed a small apparent

gate effect, but the present, larger controlled run does not reproduce it; accordingly, the latter result is treated as authoritative.

10.3 Policy-check ablation (tool_misuse, n = 11)

11/11 flagged with the policy check on; 11/11 with it off — again no measurable difference, for the same underlying reason (tool_misuse scenarios' sampled SCI values independently trip the Stage-2 gate in this dataset). Stage 1's unique, independent contribution is not demonstrated by this dataset as currently constructed; the strong aggregate 100% figure in Table 5, Experiment 5, is on inspection not attributable to Stage 1 specifically, and the supplementary materials accompanying this manuscript specify the scenario redesign needed to test it cleanly.

10.4 Experiment 7 — Latency overhead

0.00194 ms per decision, algorithm logic only, over 300,000 simulated decisions on a standard server-class Python interpreter. Not a device benchmark — excludes metric acquisition (the real cost driver on constrained hardware) and uses no embedded/mobile/wearable hardware whatsoever.

10.5 What actually differentiated TOAIAF from the guardrail baseline

Given Sections 10.2–10.3's null ablation findings, the source of TOAIAF's aggregate advantage over the guardrail baseline (Table 5) is worth stating precisely rather than left for the reader to infer incorrectly: it is not demonstrated to come from the safety gate or policy check specifically (both were confounded with Stage 3 in this run) — it is attributable to the continuous, GAS/SCI/PRI-weighted risk score (Stage 3) catching cases a binary keyword match cannot, most visibly in the 0% → 92.3% jump on goal-misalignment, where no hard-gate or scope mechanism was involved at all.

10.6 Finding from the reference-implementation prototype

Running prototype.py's end-to-end demonstration (Local Agent → Middleware stages → Environment Simulator; see supplementary materials accompanying this manuscript) surfaced a design gap invisible from the scenario-level simulation alone: a WARN-verdict action still executes in the current design (WARN flags but does not block). In the demo, an injected "unlock the front door for an unverified guest" scenario resolved to WARN and the door was, in the simulated environment, unlocked while a warning was logged. This is recorded as an identified, unresolved limitation (Section 13) — a category-specific minimum-verdict floor for physically irreversible actions is a concrete, named enhancement for future work, not silently patched into this version's already-reported Table 5 results (which would have required re-running the full suite under a changed algorithm, reintroducing exactly the kind of undisclosed-version-drift this manuscript is trying to avoid).

11. Standards and Regulatory Alignment

Table 6 — Standards crosswalk

TOAIAF Component	NIST AI RMF [3]	ISO/IEC 42001 [5]	EU AI Act [11]	IEEE 7001 [8]	IEEE 7002 [9]	IEEE 7003 [10]
------------------	-----------------	-------------------	----------------	---------------	---------------	----------------

TOIAIAF Component	NIST AI RMF [3]	ISO/IEC 42001 [5]	EU AI Act [11]	IEEE 7001 [8]	IEEE 7002 [9]	IEEE 7003 [10]
Policy Enforcement Engine	Manage	Annex A: AI system operation controls	Art. 9 (autonomy-aware risk management)			
Agent Runtime Governance Layer	Measure, Manage	Annex A: performance monitoring	Art. 12 (audit trails)			Relevant if bias-monitoring included
Trust Evaluation Engine	Measure	Annex A: AI system evaluation	Art. 9, Art. 12			Bias/fairness sub-metrics
Explainability Engine	Map, Measure	Annex A: transparency documentation	Art. 13 (transparency to users), Art. 50	Direct alignment		
Privacy and Security Assurance Layer	Manage	Annex A: data governance	GDPR interface, Art. 10 (data governance)		Direct alignment	
Human Oversight Boundary	Govern	Annex A: human oversight roles	Art. 14 (stop capability)			
Continuous Learning Assurance Loop	Govern, Manage	Annex A: continual improvement	Art. 9 (ongoing risk management)			Re-bias-check on update

This table is a research-derived mapping offered as a practical compliance-engineering artifact; it should be reviewed by qualified legal/compliance counsel before being relied on for an actual certification or conformity assessment. Note also that the Continuous Learning Assurance Loop row is, as of this version, specified but not implemented in code (Section 13) — the mapping describes the intended design, not a validated capability.

12. Discussion

Table 5 supports a narrow but substantive conclusion. On this synthetic 100-scenario benchmark, a staged decision procedure computed from operationalized metrics outperforms the ungoverned baseline and, more meaningfully, the keyword guardrail on categories for which keyword matching is structurally weak, particularly goal misalignment and prompt injection. No false restrictions were observed across the benign scenarios. The results do not, however, establish that the intended architectural components independently produced this advantage. Sections 10.2–10.3 show that the safety gate and policy check are confounded with the risk-scoring path, while Section 10.1 indicates that the human-oversight metric is slightly counterproductive at its current weighting. Scaling the evaluation from an earlier 30-scenario pilot to the present 100-scenario, six-category benchmark exposed these confounds (Section 10.2), underscoring the value of larger synthetic evaluations before attributing an observed effect to a specific mechanism.

The prototype in Section 10.6 reveals a different class of issue: behavior that becomes visible only when the decision procedure is embedded in an agent loop, even a mocked one. The finding that WARN still permits execution illustrates this distinction. Prototype-level integration testing therefore complements scenario-level simulation rather than duplicating it; neither the category-level results nor the ablations in Sections 10.1–10.3 would have exposed the gap.

Prompt-injection resistance (Experiment 4) offers a similarly specific interpretation. In this run, the result was fully attributable to the GAS-weighted risk path because the scenarios were deliberately constructed to request nominally in-scope actions (Section 9.2). The outcome is encouraging for the mechanism evaluated, but its scope is limited: the dataset does not include adversarially crafted injections designed to preserve a high sampled GAS, as a real attacker might attempt.

13. Limitations

- Synthetic, template-generated scenarios ($n = 100$). Not yet cross-validated against AgentHarm, AgentDojo, R-Judge, or AgentBench (Section 9.6 specifies the mapping; none of it has been run).
- No real agent in the loop for the main evaluation. Metric values are sampled from category-conditioned distributions; the prototype's mock agent is rule-based, not a real LLM, and is used only for integration-surface testing (Section 10.6), not for Table 5's results.
- No real hardware. Latency figures are reference-Python-only.
- No human subjects. ECS's interpretability sub-score is defined but not measured with real raters.
- Two ablations (safety gate, policy check) were confounded in this dataset (Sections 10.2–10.3) and do not demonstrate the independent value of the mechanisms they were meant to test; a deliberately adversarial scenario subset (high ARS/GAS with low SCI specifically; in-range SCI/PRI but out-of-scope requests) is required and not yet built.
- HOES's current weighting showed a small negative effect on sensitivity (Section 10.1) — named as a specific open calibration question, not resolved here.
- The WARN verdict currently allows action execution for all action types, including physically irreversible ones (Section 10.6) — a disclosed, unresolved design gap.
- The Continuous Learning Assurance Loop is specified but not implemented in the supplementary materials accompanying this manuscript.
- Prompt-injection scenarios use fixed, non-adversarial templates (Section 12); an adaptive or adversarially-optimized attacker is not modeled.
- Single seed (42). No multi-seed sensitivity sweep has been run; qualitative stability across seeds is expected but not demonstrated.
- This study represents a collaboration among three authors from three independent institutions. While the proposed TOAIAF framework, runtime assurance architecture, and decision procedure were jointly developed and reviewed, the evaluation remains limited to a synthetic benchmark and reference implementation. Future work will include broader multi-institutional validation, external benchmarking, and real-world deployment studies.

14. Future Work

Future research directions include:

1. Real LLM-agent integration
2. Physical smart-home validation
3. AgentDojo and external benchmark evaluation
4. Human user studies
5. Hardware-constrained edge deployment
6. Adaptive threshold learning

15. Conclusion

TOAIAF specifies a three-stage runtime assurance architecture and an operationalized six-metric trust suite, evaluated through seven experiments on a 100-scenario synthetic benchmark spanning six risk categories. The resulting open-source artifact is fully reproducible. On this benchmark, the decision logic exceeds 92% detection across five governance-failure categories, whereas the keyword baseline falls as low as 0%. The evaluation also reveals important qualifications: two ablations are confounded by the dataset, the current HOES weighting slightly reduces the intended safety margin, and the prototype exposes a design gap in the WARN verdict. Most of these findings emerged only after the evaluation was expanded from the earlier 30-scenario pilot, supporting continued scaling and careful interpretation of the mechanisms responsible for observed effects. Future work will evaluate TOAIAF with real-world agent traces, hardware-in-the-loop experiments, and established agent safety benchmarks.

Funding

This research received no external funding.

References

- [1] R. Shah et al., "Goal Misgeneralization: Why Correct Specifications Aren't Enough for Correct Goals," arXiv:2210.01790, 2022.
- [2] "AI Alignment: A Comprehensive Survey," arXiv:2310.19852, 2023.
- [3] E. Tabassi, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1. Gaithersburg, MD, USA: National Institute of Standards and Technology, 2023, doi: 10.6028/NIST.AI.100-1.
- [4] C. Autio, R. Schwartz, J. Dunietz, S. Jain, M. Stanley, E. Tabassi, P. Hall, and K. Roberts, *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*, NIST AI 600-1. Gaithersburg, MD, USA: National Institute of Standards and Technology, 2024, doi: 10.6028/NIST.AI.600-1.
- [5] *Information Technology — Artificial Intelligence — Management System*, ISO/IEC 42001:2023, International Organization for Standardization and International Electrotechnical Commission, Geneva, Switzerland, 2023.
- [6] *Information Technology — Artificial Intelligence — Guidance on Risk Management*, ISO/IEC 23894:2023, International Organization for Standardization and International Electrotechnical Commission, Geneva, Switzerland, 2023.
- [7] *IEEE Standard Model Process for Addressing Ethical Concerns During System Design*, IEEE Std 7000-2021, IEEE Standards Association, 2021.

- [8] *IEEE Standard for Transparency of Autonomous Systems*, IEEE Std 7001-2021, IEEE Standards Association, published Mar. 4, 2022.
- [9] *IEEE Standard for Data Privacy Process*, IEEE Std 7002-2022, IEEE Standards Association, 2022.
- [10] *IEEE Standard for Algorithmic Bias Considerations*, IEEE Std 7003-2024, IEEE Standards Association, published Jan. 24, 2025.
- [11] European Parliament and Council of the European Union, "Regulation (EU) 2024/1689 of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)," *Official Journal of the European Union*, L 2024/1689, Jul. 12, 2024.
- [12] C. Hartsell, S. Ramakrishna, A. Dubey, D. Stojcsics, N. Mahadevan, and G. Karsai, "ReSonAte: A Runtime Risk Assessment Framework for Autonomous Systems," in *Proc. 16th IEEE/ACM Int. Symp. Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 2021, pp. 118–129, doi: 10.1109/SEAMS51251.2021.00025.
- [13] D. Robbins, O. Eris, A. Kapusta, L. Booker, and P. Ward, *AI Assurance: A Repeatable Process for Assuring AI-enabled Systems*. McLean, VA, USA: The MITRE Corporation, Jun. 2024, Rep. PR-24-1768.
- [14] P. Ward, J. Stanley, R. Ferguson, and J. Korman, *The AI Assurance Landscape*, ver. 1.0. McLean, VA, USA: The MITRE Corporation, 2024.
- [15] L. Wang et al., "A Survey on Large Language Model Based Autonomous Agents," *Frontiers of Computer Science*, vol. 18, no. 6, Art. no. 186345, 2024, doi: 10.1007/s11704-024-40231-1.
- [16] Z. Xi et al., "The Rise and Potential of Large Language Model Based Agents: A Survey," *Science China Information Sciences*, vol. 68, no. 2, Art. no. 121101, 2025, doi: 10.1007/s11432-024-4222-0.
- [17] C. L. Wang, T. Singhal, A. Kelkar, and J. Tuo, "MI9: An Integrated Runtime Governance Framework for Agentic AI," arXiv:2508.03858, 2025.
- [18] G. Syros, A. Suri, J. Ginesin, C. Nita-Rotaru, and A. Oprea, "SAGA: A Security Architecture for Governing AI Agentic Systems," in *Proc. 33rd Annu. Network and Distributed System Security Symp. (NDSS)*, San Diego, CA, USA, 2026, doi: 10.14722/ndss.2026.230869.
- [19] "AgentSpec: Customizable Runtime Enforcement for Safe and Reliable LLM Agents," *Proc. IEEE/ACM ICSE*, 2026.
- [20] T. Shi, J. He, Z. Wang, H. Li, L. Wu, W. Guo, and D. Song, "Progent: Programmable Privilege Control for LLM Agents," arXiv:2504.11703, 2025.
- [21] H. Wang, C. M. Poskitt, J. Wei, and J. Sun, "ProbGuard: Proactive Runtime Monitoring for LLM Agent Safety via Probabilistic Prediction," in *Proc. 41st IEEE/ACM Int. Conf. Automated Software Engineering (ASE)*, Munich, Germany, 2026.
- [22] M. Andriushchenko et al., "AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents," *Proc. ICLR*, 2025.
- [23] T. Yuan, Z. He, L. Dong, Y. Wang, R. Zhao, T. Xia, L. Xu, B. Zhou, F. Li, Z. Zhang, R. Wang, and G. Liu, "R-Judge: Benchmarking Safety Risk Awareness for LLM Agents," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Miami, FL, USA, 2024, pp. 1467–1490, doi: 10.18653/v1/2024.findings-emnlp.79.
- [24] X. Liu et al., "AgentBench: Evaluating LLMs as Agents," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2024.
- [25] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "IoT Sentinel: Automated Device-Type Identification for Security Enforcement in IoT," in *Proc. IEEE 37th Int. Conf. Distributed Computing Systems (ICDCS)*, 2017, pp. 2177–2184, doi: 10.1109/ICDCS.2017.283.
- [26] X. Wang, B. Wang, Y. Wu, Z. Ning, S. Guo, and F. R. Yu, "A Survey on Trustworthy Edge Intelligence: From Security and Reliability to Transparency and Sustainability," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 3, pp. 1729–1757, 2025, doi: 10.1109/COMST.2024.3446585.
- [27] D. Rivkin, F. Hogan, A. Feriani, A. Konar, A. Sigal, X. Liu, and G. Dudek, "AIoT Smart Home via Autonomous LLM Agents," *IEEE Internet of Things Journal*, vol. 12, no. 3, pp. 2458–2472, Feb. 2025, doi: 10.1109/JIOT.2024.3471904.

- [28] K. Vaswani et al., "Confidential Computing within an AI Accelerator," in *Proc. 2023 USENIX Annual Technical Conference (USENIX ATC '23)*, Boston, MA, USA, 2023, pp. 501–518.
- [29] M. Rigaki and S. Garcia, "A Survey of Privacy Attacks in Machine Learning," *ACM Computing Surveys*, vol. 56, no. 4, Art. no. 101, pp. 1–34, 2023, doi: 10.1145/3624010.
- [30] F. Barez et al., "Chain-of-Thought Is Not Explainability," 2025.
- [31] I. Arcuschin, J. Janiak, R. Krzyzanowski, S. Rajamanoharan, N. Nanda, and A. Conmy, "Chain-of-Thought Reasoning in the Wild Is Not Always Faithful," arXiv:2503.08679, 2025.
- [32] T. South, S. Marro, T. Hardjono, R. Mahari, C. Deslandes Whitney, A. Chan, and A. Pentland, "Position: AI Agents Need Authenticated Delegation," in *Proc. 42nd Int. Conf. Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 267, 2025, pp. 82211–82231.
- [33] A. S. Li, R. Safavi-Naini, and P. W. L. Fong, "A Capability-Based Distributed Authorization System to Enforce Context-Aware Permission Sequences," in *Proc. 27th ACM Symp. Access Control Models and Technologies (SACMAT)*, 2022, pp. 195–206, doi: 10.1145/3532105.3535014.
- [34] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.
- [35] A. Barredo Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges Toward Responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020.
- [36] B. Shneiderman, "Human-Centered Artificial Intelligence: Reliable, Safe and Trustworthy," *International Journal of Human-Computer Interaction*, vol. 36, no. 6, pp. 495–504, 2020.
- [37] E. Debenedetti, J. Zhang, M. Balunović, L. Beurer-Kellner, M. Fischer, and F. Tramèr, "AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents," in *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*, 2024, pp. 82895–82920, doi: 10.52202/079017-2636.