IEEE-STYLE DRAFT · UNOFFICIAL

IEEE P-XXXX/D1

Draft Standard for Action-Level Assurance of Autonomous Software Agents (LAAS)

Consequence-tiered obligations, verification independence, and decision-trace evidence for actions taken by LLM-based agents

Prepared by the KellerAI Open-Source Working Group

Draft 1, 2026-06-22

Supersedes LAAS v1.0 (standard/LAAS.md, draft)

IEEE Draft Standard for Action-Level Assurance of Autonomous Software Agents (LAAS)

Designation: IEEE P-XXXX/D1, 2026 **Status:** Draft (not yet approved by IEEE)

- **Prepared by:** KellerAI Open-Source Working Group
- **Source of truth:** `conformance/laas/data.json` (machine-readable bundle `laas-fin-1.1.0`)
- **Enforcing policy:** `conformance/laas/laas.rego` (OPA package `kellerai.laas.actions`)
- **Rationale and design record:** `docs/laas/proposal-v1.1.md`
- **Supersedes:** LAAS v1.0 (`standard/LAAS.md`, draft)
- **Date of this draft:** 2026-06-22

Abstract

This standard defines action-level assurance requirements for autonomous software agents powered by large language models (LLMs). It establishes a consequence-tier framework (CT0–CT4), twelve normative obligations, verification independence and qualification criteria, enforcement-plane integrity requirements, decision-trace evidence requirements, and residual escape-rate tolerances. The standard applies to individual agent actions that produce effects outside the agent's own sandbox. It does not certify a model; it gates what an agent may commit. A deployer demonstrates conformance through a signed, sampled decision-trace bundle and an independent auditor attestation that the bundle's verdicts satisfy the obligation set.

Keywords: agent assurance; autonomous agents; blast radius; consequence tier; decision trace; enforcement gate; escape rate; independent verification; LLM; policy-as-code; pre-commit verification; zero trust.

Contents

- 1. Overview
  - 1.1 Scope
  - 1.2 Purpose
  - 1.3 Word usage
- 2. Normative References
- 3. Definitions, Acronyms, and Abbreviations
  - 3.1 Definitions
  - 3.2 Acronyms and Abbreviations
- 4. Conformance
  - 4.1 Conformance predicate
  - 4.2 Conformance attestation
  - 4.3 Obligation lifecycle
- 5. Consequence-Tier Derivation Requirements
  - 5.1 Tier derivation, general (TIER-001)
  - 5.2 Self-report constraint (SELF-001)
  - 5.3 Cumulative blast-radius aggregation (AGG-001)
  - 5.4 Untrusted-input tier elevation (INP-001)
  - 5.5 Regime requirements by tier
  - 5.6 Authorized Operating Envelope
- 6. Verification Requirements
  - 6.1 Independent pre-commit verification (IRR-001)
  - 6.2 Verifier independence (IND-001)
  - 6.3 Verifier qualification (VQ-001)
  - 6.4 Residual escape rate (RES-001)
- 7. Enforcement-Plane and Evidence Requirements
  - 7.1 Enforcement-plane integrity (ENF-001)
  - 7.2 Decision-trace requirements (TRC-001)
  - 7.3 Evidence store
  - 7.4 Bidirectional traceability
- 8. Human Authority and Third-Party Requirements
  - 8.1 Human approval at CT4 (HUM-001)
  - 8.2 Third-party and vendor obligations (VEN-001)
- Annex A (informative): Consequence-Tier Lattice and Obligation Map
- Annex B (informative): Mapping to DO-178C, SR 26-2/SR 11-7, and UL 4600
- Annex C (informative): Bibliography

1. Overview

1.1 Scope

This standard applies to **individual actions** taken by LLM-based autonomous software agents that produce effects outside the agent's own sandbox. Such actions include, but are not limited to, tool invocations, record writes, fund transfers, access-permission changes, and communications to external parties.

This standard:

- defines a consequence-tier framework (CT0–CT4) for gate-derived classification of agent actions by blast radius;
- specifies twelve normative obligations governing tier derivation, verification, enforcement-plane integrity, decision tracing, and human authority;
- establishes verifier independence, qualification, and residual escape-rate requirements;
- requires a tamper-evident decision-trace record for every gated action;
- defines conformance predicate evaluation and attestation.

This standard does **not**:

- certify a model, model provider, or model weight;
- specify the internal behavior, training data, or architecture of an LLM;
- replace vertical-sector regulations; and
- constrain actions that remain fully within a read-only or sandboxed scope (CT0).

1.2 Purpose

This standard provides a machine-actionable, two-layer assurance framework that bounds the rate at which an undetected error in an LLM agent's output reaches and is committed as a consequential external action.

This standard applies Zero-Trust to model outputs and to the apparatus surrounding them. Specifically, a conforming system shall not trust:

1. the agent's self-classification of its own action;
2. a verifier's soundness or independence without evidence; and
3. the integrity of the enforcement plane.

A control that allows the constrained party to tier, grade, or gate its own actions is non-conforming.

This standard is a **standard of care**, not a correctness guarantee. Conformance asserts that the right checks ran, by the right party, with evidence; it does not assert that no error can occur.

1.3 Word usage

The word **shall** indicates a mandatory requirement. The word **should** indicates a recommendation. The word **may** indicates a permissible action. The word **can** indicates a possibility or capability.

These definitions conform to IEEE Standards Style Manual conventions.

2. Normative References

The following documents are referenced in the body of this standard and are indispensable for its application.

- `standard/LAAS.md`: LAAS prose normative reference, v1.1 (canonical; `data.json` is derived from it).
- `conformance/laas/data.json`: machine-readable obligation bundle `laas-fin-1.1.0`; the single source of truth for tier-lattice values, escape-rate tolerances, and obligation metadata.
- `conformance/laas/laas.rego`: OPA policy implementing the conformance predicate, package `kellerai.laas.actions`.

Where the prose in this document and `conformance/laas/data.json` disagree, that disagreement is a defect; open a `policy-bug` issue against the repository. Prose cannot block a non-conforming action; the Rego policy does.

---

3. Definitions, Acronyms, and Abbreviations

3.1 Definitions

For the purposes of this standard, the following definitions apply.

action A single atomic operation issued by an agent to a tool, API, or system that may produce an effect outside the agent's own sandbox.

actor The LLM-based agent that proposes and initiates an action.

actor model lineage The base-model family, fine-tune chain, or model-family identifier associated with the agent issuing an action; used to determine whether a verifier shares the actor's substrate.

aggregate window CT The consequence tier derived from the cumulative effect of a windowed set of actions by the same principal, session, or effect class; computed by the gate and used to enforce the anti-structuring rule (see `conformance/laas/laas.rego:47–49`).

append-only store A storage system to which records can be added but not modified or deleted by any party, including the actor; integrity is maintained by content-addressing and, where applicable, periodic Merkle anchoring.

Authorized Operating Envelope (AOE) The operator-declared set of permissions, tool classes, data domains, action classes, and CT ceilings within which the agent may operate; analogous to the Operational Design Domain in autonomous-vehicle standards.

backtest An empirical evaluation of a verifier or system against a held-out, representative, and adversarially-stressed set of inputs, yielding an estimated escape rate with a stated confidence interval.

blast radius The maximum adverse impact of an action along the axes of reversibility, scope, and consequence; the quantity from which the consequence tier is derived.

blocked action An action that the gate has prevented from committing because a required check did not pass or returned `fail`, `abstain`, or `indeterminate`; recorded in the decision trace with `action_blocked: true`.

Bucket A The class of claims for which a deterministic exact verifier exists (e.g., schema validation, ledger reconciliation, hash comparison); the escape rate for Bucket A approaches zero, bounded by verifier soundness.

Bucket B The class of claims for which no exact oracle exists; the operator shall declare a maximum acceptable escape rate, estimate it by backtesting, and control it by independent or human review above tolerance.

bundle The signed, version-pinned policy artifact comprising the obligation set, tier-lattice values, and escape-rate tolerances; identified by `bundle_id` in `conformance/laas/data.json`.

consequence The worst-case impact of an action, classified as `none`, `low`, `material`, or `high` by financial, safety, legal, or reputational effect; unknown consequence defaults to `high` (`conformance/laas/data.json:9`, `docs/laas/proposal-v1.1.md:§6.2`).

consequence tier (CT) An integer in the range [0, 4] that classifies an action's blast radius; higher values indicate greater consequence; derived by the gate from the observed effect surface (`conformance/laas/laas.rego:29–44`).

decision trace An append-only, content-addressed, hash-chained record emitted by the gate for every gated action, carrying the fields enumerated in Section 7.2 of this standard.

deployer The organization or individual that operates a conforming agent system against this standard.

effective CT The maximum of the gate-assigned CT and the aggregate-window CT; the tier actually enforced (`conformance/laas/laas.rego:47`).

enforcement gate (gate) An out-of-process component that (a) observes the action's effect surface, (b) derives the consequence tier, (c) invokes the required verifiers, (d) emits the decision trace, and (e) blocks or permits the action; the actor cannot disable, modify, or bypass the gate.

error correlation A measured statistic characterizing the degree to which two checkers (e.g., actor and verifier) fail on the same inputs; for a model verifier to satisfy independence, its error correlation with the actor shall not exceed 0.2 (`conformance/laas/data.json:14`).

escalation The act of routing a blocked or out-of-envelope action to a human review queue or equivalent authority, with the queue identifier and resulting ticket recorded in the decision trace.

escape rate The rate at which a wrong output from an agent passes every applicable check and is committed as an external effect; the governing metric for Bucket B conformance.

expected CT The consequence tier derived by the gate from the tier lattice applied to the observed effect surface; the floor below which the gate-assigned CT shall not fall (`conformance/laas/laas.rego:29–44`).

human approval An explicit authorization by a human operator, recorded in the decision trace, granting permission for a specific CT4 action or action class to proceed.

independence basis The documented reason a verifier satisfies the independence requirement; one of `deterministic` , `distinct_lineage_low_correlation` , or `human` .

independent verifier (verifier) A component that checks an action's correctness or safety properties without sharing the actor's model substrate or check logic; independence is established by checker type, lineage divergence and low error-correlation, or human judgment.

obligation A normative requirement in this standard identified by a stable obligation ID (e.g., `LAAS-OBL-TIER-001`), carrying a severity (`error` or `warning`), a CT floor, a precedence value, and a trigger predicate; violation of an `error` -severity obligation blocks conformance.

observed effect surface The actual tool, API, scope, amount, and metadata of the action as observed by the gate at runtime, not as declared by the actor.

out-of-process gate A gate component that executes in a process or trust boundary separate from the actor, such that the actor cannot disable or modify the gate even when operating with elevated permissions.

policy bundle Synonym for bundle; see that definition.

pre-commit verification Verification of an action that completes before the action is committed or takes effect; verification after commitment does not satisfy the pre-commit requirement.

qualification reference A stable identifier (`verifier_qualification_ref`) pointing to the verifier qualification record in the evidence store, comprising coverage documentation, a negative-test suite, and version information.

residual escape rate The measured escape rate for the Bucket B class of claims associated with a given action at the applicable CT; shall not exceed the tolerance declared in `conformance/laas/data.json` .

reversibility The extent to which an action can be undone; classified as `reversible` , `hard` , `irreversible` , or `none` ; unknown reversibility defaults to `irreversible` (`docs/laas/proposal-v1.1.md:§6.2`).

scope The breadth of parties affected by an action; classified as `single` , `multi` , `org` , or `public` ; unknown scope defaults to `public` (`docs/laas/proposal-v1.1.md:§6.2`).

self-reported CT The consequence tier asserted by the actor in the action descriptor; informational only; may not lower the gate-derived tier.

standing envelope A pre-authorized action class with explicit limits registered by a human operator, within which the gate enforces bounds and escalates only out-of-envelope actions.

verdict The outcome of verifier evaluation; one of `pass` , `fail` , `abstain` , or `indeterminate` .

verifier See independent verifier.

verifier qualification Evidence that a verifier meets the requirements of Section 6.3 of this standard; analogous to tool qualification under RTCA DO-330.

3.2 Acronyms and Abbreviations

Acronym	Expansion
AOE	Authorized Operating Envelope
CT	Consequence Tier
CT0–CT4	Consequence Tier levels 0 through 4
DAL	Design Assurance Level (DO-178C)
DO-178C	RTCA DO-178C, Software Considerations in Airborne Systems and Equipment Certification
DO-330	RTCA DO-330, Software Tool Qualification Considerations
GPAI	General-Purpose AI (EU AI Act)
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
LAAS	LLM-Agent Assurance Standard
LLM	Large Language Model
MC/DC	Modified Condition/Decision Coverage
MNPI	Material Non-Public Information
NIST	National Institute of Standards and Technology
ODD	Operational Design Domain (UL 4600)
OPA	Open Policy Agent
PII	Personally Identifiable Information
SR 11-7	Federal Reserve SR Letter 11-7, Supervisory Guidance on Model Risk Management
SR 26-2	Federal Reserve SR Letter 26-2 (successor to SR 11-7)
UL 4600	UL Standard for Safety for the Evaluation of Autonomous Products

4. Conformance

4.1 Conformance predicate

A conforming system shall satisfy, for every gated action, the following predicate (stated normatively; implemented in `conformance/laas/laas.rego:154–191`):

> If any obligation's trigger condition matched the action, then the action either (a) passed an independent, qualified pre-commit verifier (plus human approval when the effective CT is 4), with measured residual escape rate not exceeding the declared tolerance, **or** (b) was blocked and escalated. No other outcome conforms.

4.1.1 A deployer's system shall be capable of evaluating this predicate mechanically, given only the decision-trace record and the obligation bundle from `conformance/laas/data.json`, without human explanation.

4.1.2 The predicate shall be evaluated by the gate at the time of the action, before commitment. Post-hoc evaluation does not satisfy this requirement.

4.1.3 An `error`-severity obligation violation makes the action non-conforming. A `warning`-severity violation shall be recorded in the decision trace and reported but does not, by itself, make the action non-conforming.

4.1.4 When two or more obligations fire with incompatible outcomes, the gate shall resolve the conflict deterministically: the obligation with the highest CT floor takes precedence; ties resolve to the most restrictive outcome (`block`). The resolution order shall be recorded in the decision trace. Obligation precedence values are defined in `conformance/laas/data.json:19–32`.

4.2 Conformance attestation

4.2.1 A deployer shall demonstrate conformance by producing a **signed, sampled trace bundle** (an integrity-verified representative slice of decision-trace records over a declared period), accompanied by an **independent auditor attestation** that the bundle's verdicts satisfy the obligation set.

4.2.2 The trace bundle shall be content-addressed and include sufficient records to support statistical inference about the escape rate at each applicable CT.

4.2.3 The auditor providing the attestation shall be independent of the deployer and shall not share a model lineage or organizational control relationship with the system under audit.

4.2.4 This attestation artifact is the conformance evidence reviewed by regulators, customers, and certification bodies. It shall graduate cleanly into a third-party certificate when an accrediting body adopts this standard.

4.3 Obligation lifecycle

4.3.1 Each obligation in the bundle carries a lifecycle status: `propose`, `active`, `deprecated`, or `retired`. Only obligations in `active` status constitute normative requirements.

4.3.2 Retiring an obligation removes it from the active set. A deployer's conformance evidence shall reference the bundle version in force at the time of the traced actions.

4.3.3 The bundle version shall be recorded in every decision-trace record (`policy_bundle_version`; see Section 7.2).

---

5. Consequence-Tier Derivation Requirements

5.1 Tier derivation, general (TIER-001)

5.1.1 The gate shall derive the consequence tier for every action from the **observed effect surface** (the actual tool, API, scope, and consequence attributes as observed at runtime), not from the actor's self-report.

5.1.2 The gate shall apply the following derivation function (`conformance/laas/data.json:6–10` , `conformance/laas/laas.rego:29–44`):

```
expected_ct = max( reversibility_rank, scope_rank, consequence_rank )   # external effect
expected_ct = 0                                                          # read-only / sandboxed
expected_ct = 4                                                          # any axis undetermined
```

where the rank mappings are:

Axis	Value	Rank
`reversibility`	`reversible`	1
`reversibility`	`hard`	3
`reversibility`	`irreversible`	4
`reversibility`	`none` (unknown)	4
`scope`	`single`	1
`scope`	`multi`	2
`scope`	`org`	3
`scope`	`public`	4
`consequence`	`none`	0
`consequence`	`low`	1
`consequence`	`material`	3
`consequence`	`high`	4

5.1.3 When any axis of the observed effect surface is undetermined or absent, the gate shall assign `expected_ct = 4` . This is the default when the gate cannot observe the effect surface (`conformance/laas/laas.rego:29`).

5.1.4 The gate-assigned CT (`gate.assigned_ct`) shall not be less than the lattice-derived `expected_ct` . Violation of this requirement constitutes a TIER-001 (`LAAS-OBL-TIER-001`) error-severity violation (`conformance/laas/laas.rego:98–103`).

5.1.5 The effective CT enforced by the gate shall be:

```
effective_ct = max( gate.assigned_ct, aggregate_window_ct )
```

where `aggregate_window_ct` is the cumulative window effect tier defined in Section 5.3 (`conformance/laas/laas.rego:47–49`).

5.1.6 The operational definitions of the three axes shall be:

- **reversibility.** `reversible`: a tested, in-window, single-step undo exists; `hard`: undo exists but is costly, multi-step, or time-bounded; `irreversible`: no programmatic undo exists; `none`: unknown. Unknown reversibility shall be treated as `irreversible`.
- **scope.** `single`: one record or user; `multi`: many records or users; `org`: organization-wide; `public`: external parties or the public. Unknown scope shall be treated as `public`.
- **consequence.** Classified as `none`, `low`, `material`, or `high` by worst-case financial, safety, legal, or reputational impact. Unknown consequence shall be treated as `high`.

5.2 Self-report constraint (SELF-001)

5.2.1 An actor may include a self-reported consequence tier (`self_reported_ct`) in the action descriptor. This field is informational; it shall be recorded in the decision trace.

5.2.2 A self-reported tier that is lower than the gate-derived tier shall not reduce the effective CT. The gate tier prevails.

5.2.3 When `self_reported_ct` is less than `gate.assigned_ct`, the gate shall record a SELF-001 (`LAAS-OBL-SELF-001`) warning-severity violation in the decision trace (`conformance/laas/laas.rego:106-111`). This warning does not block the action.

5.3 Cumulative blast-radius aggregation (AGG-001)

5.3.1 The gate shall maintain a windowed aggregate effect tier per principal, session, and effect class (`aggregate_window_ct`).

5.3.2 When the aggregate effect of a sequence of actions within the window crosses a CT threshold, subsequent actions in that window shall be re-tiered to the aggregate's CT (`conformance/laas/laas.rego:47-49`, `conformance/laas/laas.rego:130-135`).

5.3.3 The gate-assigned CT for any action shall not be less than `aggregate_window_ct`. Violation constitutes an AGG-001 (`LAAS-OBL-AGG-001`) error-severity violation.

5.3.4 This requirement closes the structuring loophole by which an actor could decompose a high-blast-radius effect into a sequence of individually sub-threshold actions.

5.3.5 The `aggregate_window_ct` value shall be recorded in the decision trace for every action (`docs/laas/proposal-v1.1.md:§6.4`).

5.4 Untrusted-input tier elevation (INP-001)

5.4.1 The gate shall classify every input to an action as trusted or untrusted. Inputs from web content, inbound email, third-party data feeds, or any source outside the operator's control perimeter are untrusted unless the operator has explicitly certified them as trusted.

5.4.2 When an action is driven by untrusted input and the effective CT would otherwise be below 3, the gate shall either raise the effective CT to at least 3 or block the action (`conformance/laas/data.json:17`, `conformance/laas/laas.rego:138-145`).

5.4.3 Failure to satisfy Requirement 5.4.2 constitutes an INP-001 (`LAAS-OBL-INP-001`) error-severity violation.

5.4.4 The `input_trusted` field shall be recorded in the decision trace for every action.

5.4.5 This requirement protects tier derivation and verifier trust from prompt-injection attacks (`docs/laas/proposal-v1.1.md:§8.3`).

5.5 Regime requirements by tier

5.5.1 The gate shall enforce the following minimum regime for each effective CT.

Effective CT	Minimum regime
0	Trace only.
1	Self-check using an exact verifier if one exists, plus trace.
2	Independent automated check **or** rehearsed rollback, plus bounded residual escape rate, plus trace.
3	Mandatory independent, qualified pre-commit verification (no self-grading), plus residual escape rate ≤ tolerance, plus rollback plan, plus trace.
4	CT3 controls, plus human approval, plus abstention default, plus full evidence package, plus trace.

5.5.2 A rollback path substituted for pre-commit verification at CT2 shall be supported by **periodic rollback-rehearsal evidence**. An unrehearsed rollback does not satisfy the CT2 regime requirement (`docs/laas/proposal-v1.1.md:§6.3`).

5.5.3 At CT4, abstention shall be the default when the gate cannot confirm a passing verifier and an approved human escalation.

5.6 Authorized Operating Envelope

5.6.1 Operators should define an Authorized Operating Envelope enumerating the permissions, tool classes, data domains, action classes, and CT ceiling within which the agent may operate.

5.6.2 An action outside the AOE shall result in a first-class `abstain` verdict and escalation; it shall not be committed.

5.6.3 At CT4, operators may register standing pre-authorized envelopes with explicit limits. The gate shall enforce those limits and escalate only out-of-envelope actions. Operators may also use batched approval for queued in-envelope actions. These mechanisms scale human oversight to exceptions rather than volume (`docs/laas/proposal-v1.1.md:§6.3`).

---

6. Verification Requirements

6.1 Independent pre-commit verification (IRR-001)

6.1.1 For every action whose effective CT is 3 or 4, the gate shall require a passing independent, qualified pre-commit verifier before the action is committed (`conformance/laas/data.json:13`, `conformance/laas/laas.rego:153–158`).

6.1.2 Pre-commit means verification completes and the gate evaluates the verdict before the action takes effect. Post-execution verification does not satisfy this requirement.

6.1.3 If the verifier returns a verdict of `fail`, `abstain`, or `indeterminate`, the gate shall block the action and escalate it. The block and escalation shall be recorded in the decision trace (`docs/laas/proposal-v1.1.md:§7.1`).

6.1.4 An action at CT≥3 that is neither passed by a conforming verifier nor blocked constitutes an IRR-001 (`LAAS-OBL-IRR-001`) error-severity violation.

6.1.5 For CT4 actions, the human approval requirement of Section 8.1 shall be satisfied in addition to the verifier requirement of this section.

6.1.6 The verifier's identity, type, independence basis, qualification reference, input hash, and verdict shall be recorded in the decision trace for every CT≥3 action.

6.2 Verifier independence (IND-001)

6.2.1 A verifier satisfies the independence requirement if exactly one of the following conditions holds (`conformance/laas/laas.rego:75–83`, `docs/laas/proposal-v1.1.md:§7.3`):

1. The verifier is a **deterministic or exact checker**: a different kind of checker from the actor (e.g., a schema validator, ledger reconciliation engine, hash comparison, or allowlist check). A deterministic verifier is independent at any CT.
2. The verifier is a **distinct model lineage** from the actor and its measured error correlation with the actor does not exceed 0.2 (`conformance/laas/data.json:14`). A shared-substrate verifier (same base model or fine-tune family) is presumed non-independent. This form of independence is permitted at CT3 but not at CT4 in isolation.
3. The verifier is a **human**. Human review satisfies independence at any CT and is required in addition to automated verification at CT4.

6.2.2 Where error correlation between a model verifier and the actor cannot be bounded empirically, the gate shall fall back to a deterministic or human verifier at CT≥3.

6.2.3 A verifier that does not satisfy any condition in Requirement 6.2.1 is non-independent. Using a non-independent verifier for a CT≥3 action, when the action was not blocked, constitutes an IND-001 (`LAAS-OBL-IND-001`) error-severity violation (`conformance/laas/laas.rego:161–166`).

6.2.4 The independence basis shall be recorded as one of the string values `deterministic`, `distinct_lineage_low_correlation`, or `human` in the decision trace.

6.3 Verifier qualification (VQ-001)

6.3.1 A verifier gating a CT≥3 action shall be qualified before use (`conformance/laas/data.json:13`, `conformance/laas/laas.rego:168–174`).

6.3.2 Qualification shall comprise, at minimum:

1. **Documented coverage** of the verifier's claim class: the set of properties or conditions the verifier asserts, and the scope of inputs over which it has been tested.
2. **A negative-test suite**: a set of known-bad inputs that the verifier must catch; the verifier shall fail (return `fail`) on all members of the negative-test suite.
3. **A change-controlled version**: a stable version identifier for the verifier, recorded in the trace as `verifier_qualification_ref`, such that a change to the verifier triggers re-qualification.

6.3.3 Qualification depth should scale with CT: higher consequence tiers warrant broader coverage and larger negative-test suites (`docs/laas/proposal-v1.1.md:§7.5`).

6.3.4 An unqualified verifier does not satisfy the IRR-001 pre-commit requirement. Using an unqualified verifier for a CT≥3 action, when the action was not blocked, constitutes a VQ-001 (`LAAS-OBL-VQ-001`) error-severity violation.

6.3.5 The qualification record shall be maintained in the evidence store (Section 7.3) and shall be referenced by `verifier_qualification_ref` in the decision trace.

6.4 Residual escape rate (RES-001)

6.4.1 For Bucket B claims at CT≥2, the deployer shall declare a maximum acceptable escape rate for each applicable CT (`conformance/laas/data.json:15`, `conformance/laas/laas.rego:183-191`).

6.4.2 The declared tolerances shall not exceed the following maximum values:

Effective CT	Maximum escape-rate tolerance
2	0.02 (2 %)
3	0.005 (0.5 %)
4	0.0 (0 %)

These values are normative and are sourced from `conformance/laas/data.json` key `escape_rate_tolerance_by_ct`.

6.4.3 The deployer shall estimate the escape rate by backtesting on a held-out, representative, and adversarially-stressed evaluation set, with a stated confidence interval.

6.4.4 The deployer shall re-measure the escape rate on any change to the model, prompt, tool set, or policy.

6.4.5 When the measured residual escape rate bound (`residual_error_bound`) of an action exceeds the tolerance for its effective CT, the gate shall record an RES-001 (`LAAS-OBL-RES-001`) error-severity violation.

6.4.6 For pure Bucket A claims where an exact verifier eliminates undetected error, `residual_error_bound` may be null; in that case, the RES-001 check does not apply to that action (`conformance/laas/laas.rego:183-191`).

6.4.7 Evidence of the escape-rate measurement, including the backtest report reference, shall be recorded in the decision trace under `evidence_refs`.

---

7. Enforcement-Plane and Evidence Requirements

7.1 Enforcement-plane integrity (ENF-001)

7.1.1 The gate shall operate as an out-of-process component in a trust boundary separate from the actor (`conformance/laas/data.json:18`, `conformance/laas/laas.rego:119–122`). The actor shall not be able to disable, modify, bypass, or influence the gate, even when the actor operates with elevated permissions (e.g., in modes that disable standard permission checks).

7.1.2 The policy bundle executed by the gate shall be **signed** and **version-pinned** (`conformance/laas/data.json:17`, `conformance/laas/laas.rego:114–117`). The gate shall verify the signature before executing the bundle.

7.1.3 The decision-trace sink shall be append-only; the actor shall not be able to modify or delete trace records (`conformance/laas/laas.rego:125–128`).

7.1.4 A decision trace asserting an in-process gate or an unsigned bundle is non-conforming regardless of the verdict recorded, and constitutes an ENF-001 (`LAAS-OBL-ENF-001`) error-severity violation (`docs/laas/proposal-v1.1.md:§7.7`).

7.1.5 `bundle_signed` and `gate_out_of_process` shall be recorded as boolean fields in every decision-trace record.

7.2 Decision-trace requirements (TRC-001)

7.2.1 The gate shall emit one decision-trace record for every gated action.

7.2.2 Each decision-trace record shall include, at minimum, the following fields (`docs/laas/proposal-v1.1.md:§7.2`):

Field	Description
`obligation_id`	The obligation(s) evaluated for this action
`obligation_version`	The bundle version of the obligation(s)
`action_ref`	A stable identifier for the action
`effect_surface_hash`	Cryptographic hash of the gate-observed effect surface
`gate_derived_ct`	The expected CT from the tier lattice
`effective_ct`	`max(gate.assigned_ct, aggregate_window_ct)`
`self_reported_ct`	The actor's self-declared tier (informational)
`trigger_matched`	Whether the obligation trigger condition fired
`verifier_id`	Identifier of the verifier invoked
`verifier_type`	One of `deterministic`, `model`, `human`
`verifier_independent`	Boolean: whether the verifier satisfies Section 6.2
`independence_basis`	One of `deterministic`, `distinct_lineage_low_correlation`, `human`
`verifier_qualified`	Boolean: whether the verifier satisfies Section 6.3
`verifier_qualification_ref`	Reference to the qualification record
`verifier_error_correlation`	Measured correlation with the actor; null for non-model verifiers
`verifier_input_hash`	Cryptographic hash of inputs provided to the verifier
`verdict`	One of `pass`, `fail`, `abstain`, `indeterminate`
`residual_error_bound`	Measured escape rate; null for pure Bucket A actions
`residual_tolerance`	Declared maximum tolerance for the effective CT
`evidence_refs`	References to backtest reports or other supporting evidence
`escalation`	Null, or `{queue, ticket_id}` when the action was escalated
`escalation_approved`	Boolean: whether a human approved the escalated action
`action_blocked`	Boolean: whether the gate blocked the action
`actor_id`	Stable identifier for the actor
`policy_bundle_version`	Version of the bundle in force at evaluation time
`bundle_signed`	Boolean: whether the gate verified the bundle signature
`gate_out_of_process`	Boolean: whether the gate is out-of-process
`aggregate_window_ct`	Cumulative blast-radius tier at the time of this action
`input_trusted`	Boolean: whether all inputs are trusted
`timestamp`	ISO 8601 timestamp of gate evaluation
`actor_chain_prev_hash`	Hash of the prior record in the actor's hash chain
`merkle_anchor`	Periodic shared Merkle root, when applicable

7.2.3 Records shall be written to an append-only, content-addressed store that the actor cannot rewrite.

7.2.4 Each actor shall maintain its own hash chain using `actor_chain_prev_hash`, linking each record to the prior record in that actor's trace. Hash chains shall be periodically anchored into a shared Merkle root

(`merkle_anchor`) written to an append-only sink. This structure is safe for concurrent multi-agent environments (`docs/laas/proposal-v1.1.md:§7.4`).

7.2.5 Bucket A verdicts shall be deterministically reproducible against the recorded input hash. Bucket B verdicts shall be tamper-evidently re-inspectable: the recorded verdict and its evidence shall be re-readable, though re-deriving a non-deterministic model verdict is not required.

7.2.6 Failure to emit a conforming, append-only, chained trace record constitutes a TRC-001 (`LAAS-OBL-TRC-001`) error-severity violation (`conformance/laas/laas.rego:125-128`).

7.3 Evidence store

7.3.1 The deployer shall maintain an evidence store that is append-only and content-addressed.

7.3.2 Retention periods shall scale with CT: higher tiers shall be retained for longer periods; regulated verticals shall meet their applicable statutory minimum retention periods.

7.3.3 The deployer shall minimize payload size, tokenize or redact PII and MNPI, and separate the searchable index from sensitive payloads so that audit access does not require exposing regulated data.

7.3.4 Verifier qualification records, backtest reports, and escalation records shall be maintained in the evidence store and shall be accessible by reference from the decision trace.

7.4 Bidirectional traceability

7.4.1 A stable action or task identifier shall thread the obligation evaluation, verifier run, evidence record, escalation, and commit for every action.

7.4.2 Every committed CT≥3 action shall have either a passing or a blocked trace record.

7.4.3 Every active obligation in the bundle shall be exercised by at least one trace record in the conformance evidence, or shall be marked not-applicable with a documented reason.

7.4.4 These requirements are the analogue of bidirectional requirements-to-test traceability in DO-178C.

---

8. Human Authority and Third-Party Requirements

8.1 Human approval at CT4 (HUM-001)

8.1.1 An action whose effective CT is 4 shall not be committed without explicit human approval, unless the action is blocked (`conformance/laas/data.json:13` , `conformance/laas/laas.rego:177–181`).

8.1.2 Human approval shall be recorded in the decision trace as `human_approval.approved: true` and `escalation_approved: true` .

8.1.3 A CT4 action that is neither covered by human approval nor blocked constitutes an HUM-001 (`LAAS-OBL-HUM-001`) error-severity violation.

8.1.4 Operators may satisfy the human approval requirement through standing pre-authorized envelopes (Section 5.6.3). The gate shall enforce declared limits and escalate out-of-envelope actions for per-action human review.

8.1.5 Batched human approval of queued in-envelope actions is permitted. The gate shall record the approval batch reference in the decision trace for each covered action.

8.1.6 A human verifier satisfying the approval requirement at CT4 also satisfies the verifier independence requirement of Section 6.2 for that action.

8.2 Third-party and vendor obligations (VEN-001)

8.2.1 When the agent uses a vendor model, calls a third-party tool, or relies on a third-party API to produce or inform an action, the decision trace shall carry:

1. **provenance**: identification of the vendor model, tool, or API used;
2. **scope limits**: explicit bounds on the scope within which the vendor component is authorized to contribute to the action.

8.2.2 Errors attributable to vendor model outputs shall count against the deployer's escape-rate budget for the applicable CT. The deployer remains responsible for conformance; vendor indemnities do not reduce the deployer's escape-rate obligation.

8.2.3 Untrusted third-party dependencies shall fail closed: when the gate cannot verify the trustworthiness or scope limit of a dependency, it shall block the action and escalate.

8.2.4 Failure to record vendor attribution and scope limits when a vendor dependency is used constitutes a VEN-001 (`LAAS-OBL-VEN-001`) error-severity violation (`conformance/laas/laas.rego:148–151`).

Annex A (informative): Consequence-Tier Lattice and Obligation Map

This annex is informative. It provides reference tables for implementers.

A.1 Tier-lattice rank values

The following values are normative in Section 5 and are sourced from `conformance/laas/data.json` key `tier_lattice`.

Reversibility ranks

`reversibility` value	CT rank
`reversible`	1
`hard`	3
`irreversible`	4
`none` (unknown)	4

Scope ranks

`scope` value	CT rank
`single`	1
`multi`	2
`org`	3
`public`	4

Consequence ranks

`consequence` value	CT rank
`none`	0
`low`	1
`material`	3
`high`	4

A.2 Worked derivation example

An agent calls `payments.transfer(amount=250000, dest=external)`. The gate observes: `reversibility=irreversible` (rank 4), `scope=public` (rank 4), `consequence=high` (rank 4). `expected_ct = max(4, 4, 4) = 4`. The actor's self-reported tier is irrelevant; the gate-derived tier is CT4. The system requires independent pre-commit verification (Section 6.1), verifier independence and qualification (Sections 6.2–6.3), residual escape rate of 0 % (Section 6.4, `data.json` key `escape_rate_tolerance_by_ct["4"]`), and human approval (Section 8.1). Source: `docs/laas/proposal-v1.1.md:§6.1`.

A.3 Obligation-to-clause map

All twelve obligations and their normative clause numbers in this standard.

Obligation ID	Title	Sev	CT floor	Clause	Precedence
LAAS-OBL-TIER-001	Tier is gate-derived from observed effect surface	error	0	5.1	100
LAAS-OBL-SELF-001	Self-reported tier may not lower gate tier	warning	0	5.2	90
LAAS-OBL-ENF-001	Enforcement-plane integrity	error	0	7.1	99
LAAS-OBL-TRC-001	Append-only chained decision trace	error	0	7.2	50
LAAS-OBL-AGG-001	Cumulative blast-radius aggregation and re-tiering	error	0	5.3	85
LAAS-OBL-INP-001	Untrusted input raises tier or blocks	error	0	5.4	75
LAAS-OBL-VEN-001	Vendor attribution and scope limits	error	0	8.2	60
LAAS-OBL-IRR-001	Independent pre-commit verification for CT≥3	error	3	6.1	80
LAAS-OBL-IND-001	Verifier independence and low error-correlation	error	3	6.2	80
LAAS-OBL-VQ-001	Verifier qualification (DO-330 analogue)	error	3	6.3	80
LAAS-OBL-RES-001	Bounded residual escape rate (Bucket B)	error	2	6.4	70
LAAS-OBL-HUM-001	Human approval required at CT4	error	4	8.1	95

Source: `conformance/laas/data.json` key `obligations`.

A.4 Minimum regime by effective CT

CT	Trace	Self-check	Ind. verifier	Qualified verifier	Residual ≤ tolerance	Human approval	Rollback plan
0	yes	no	no	no	no	no	no
1	yes	yes	no	no	no	no	no
2	yes	no	yes (or rehearsed rollback)	no	yes	no	yes
3	yes	no	yes	yes	yes	no	yes
4	yes	no	yes	yes	yes (0 %)	yes	yes

Annex B (informative): Mapping to DO-178C, SR 26-2/SR 11-7, and UL 4600

This annex is informative. It identifies the structural analogues between this standard and the three mature standards that informed its design. Source: `docs/laas/proposal-v1.1.md:§2`, `§3`.

B.1 Common structural object

All four standards share a common governing structure across five mechanics.

Mechanic	DO-178C	SR 11-7 / SR 26-2	UL 4600	This standard (LAAS)
Scale rigor to consequence	DAL A–E from failure effect	Materiality tiers	Risk-based case depth	CT0–CT4 from blast radius (Sec 5)
Bound undetected error	Structural coverage (MC/DC)	Backtesting	Safety Performance Indicators	Escape rate per CT (Sec 6.4)
Independent verification	Objectives "with independence" (verifier ≠ author)	Effective challenge	Independent safety assessment	No self-grading; lineage + low correlation (Sec 6.2)
Trace to evidence	Bidirectional trace + data package	Validation docs	GSN + evidence	Decision-trace JSONL, hash-chained (Sec 7.2)
Declared envelope + abstain	Config/operational envelope	Approved use	ODD	AOE (Sec 5.6)

B.2 Specific analogues

DO-178C (RTCA)

- DAL A–E ? CT4–CT0: severity runs in opposite directions (DAL A is most severe; CT4 is most severe). Note: do not conflate the letter ordering with CT ordering.
- MC/DC structural coverage ? verifier qualification coverage (Section 6.3).
- Data package / certification artifact ? signed trace bundle + auditor attestation (Section 4.2).
- Tool qualification (DO-330) ? verifier qualification, `LAAS-OBL-VQ-001` (Section 6.3).
- Bidirectional requirements-to-test traceability ? bidirectional obligation-to-trace traceability (Section 7.4).

SR 11-7 / SR 26-2 (Federal Reserve)

- Effective challenge ? independent pre-commit verification (`LAAS-OBL-IRR-001`, Section 6.1).
- Materiality-based risk tiering ? CT tiering by blast radius (Section 5.1).
- Ongoing monitoring and backtesting ? escape-rate measurement and re-measurement on change (Section 6.4).
- Vendor model risk ? `LAAS-OBL-VEN-001` vendor attribution (Section 8.2).

UL 4600 (Underwriters Laboratories)

- Standard of care, not pass/fail ? this standard is a standard of care (Section 1.2).
- Operational Design Domain (ODD) + abstention ? Authorized Operating Envelope (Section 5.6).
- Safety Performance Indicators ? residual escape rate (Section 6.4).
- ISO 21448 (SOTIF) "no failure, still wrong" ? Bucket B open-world bounding and independent review (Section 6.4, `docs/laas/proposal-v1.1.md:§5`).

B.3 Key differences from predecessor standards

- **Scope:** DO-178C and ARP6983/ED-324 are explicitly scoped to frozen, supervised ML in embedded avionics; they do not address generative LLMs or online agents. UL 4600 addresses autonomous vehicles; its ODD concept is a direct ancestor of the AOE. SR 11-7/SR 26-2 addresses model risk in financial services but does not define a per-action runtime gate.
- **Machine-actionability:** this standard ships as a two-layer artifact (normative prose plus machine-evaluable OPA policy) that a fresh agent can evaluate without human explanation. The predecessor standards produce human-reviewed artifacts.
- **Governing metric:** this standard names the escape rate as the explicit governing metric; the predecessor standards each use analogous but differently named quantities.

Annex C (informative): Bibliography

This annex is informative.

The following references informed the design of this standard. Dates, titles, and availability reflect information current as of June 2026. Readers should confirm current versions before relying on specific document details.

Aviation and embedded software

- RTCA DO-178C, *Software Considerations in Airborne Systems and Equipment Certification*, RTCA Inc., 2011.
- RTCA DO-330, *Software Tool Qualification Considerations*, RTCA Inc., 2011.
- ARP6983 / ED-324, SAE G-34 / EUROCAE WG-114, lifecycle standard for ML in airborne systems (approximately Q1 2026; confirm current status before citing).

Financial-services model risk

- Board of Governors of the Federal Reserve System, SR Letter 11-7, *Supervisory Guidance on Model Risk Management*, April 2011.
- Board of Governors of the Federal Reserve System, SR Letter 26-2 (successor to SR 11-7; confirm issuance status as of date of use).

Autonomous-vehicle safety

- UL 4600, *Standard for Safety for the Evaluation of Autonomous Products*, Underwriters Laboratories, edition current at time of use.
- ISO 21448, *Road vehicles: Safety of the intended functionality (SOTIF)*, ISO, 2022.

AI governance and management systems

- ISO/IEC 42001:2023, *Artificial Intelligence: Management system*.
- NIST AI 100-1, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, National Institute of Standards and Technology, 2023.
- NIST AI 600-1, *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*, National Institute of Standards and Technology, July 2024.
- European Parliament and of the Council, *Regulation (EU) 2024/1689 (EU AI Act)*, July 2024.

Agentic AI security and governance

- CISA / NCSC / NSA / ASD / CCCS / GCSB, *Careful Adoption of Agentic AI Services*, April–May 2026, approximately 29 pages.
- Cloud Security Alliance, MAESTRO Threat Modeling Framework, 2025–2026.
- OWASP, *Agentic AI Security Top 10*, 2025–2026.

LAAS internal sources (normative)

- `standard/LAAS.md`: canonical prose specification, LAAS v1.1.
- `conformance/laas/data.json`: machine-readable obligation bundle `laas-fin-1.1.0`.
- `conformance/laas/laas.rego`: OPA policy, package `kellerai.laas.actions`.
- `docs/laas/proposal-v1.1.md`: rationale and design record, LAAS v1.1.

End of IEEE P-XXXX/D1, LAAS Draft Standard