# ISO/IEC XXXXX

First edition · 2026 · UNOFFICIAL DRAFT

# Information technology - Artificial intelligence - Action-level assurance for autonomous agents (LAAS)

Reference number
**ISO/IEC XXXXX:2026(E)**

# ISO/IEC XXXXX:2026(E)

## Information technology - Artificial intelligence - Action-level assurance for autonomous agents (LAAS)

## Foreword

This document was prepared by an independent working group modelled on ISO/IEC JTC 1/SC 42 drafting conventions. The structure follows ISO/IEC Directives, Part 2, 2021 and the ISO Harmonized Structure (Annex SL).

Attention is drawn to the possibility that some elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

This document will be reviewed at five-year intervals and, if necessary, revised.

Direct any feedback or questions on this document to your national standards body. These bodies are listed at https://www.iso.org/members.html.

## Introduction

Autonomous agents powered by large language models (LLMs) act on behalf of natural persons and organisations by calling tools, writing records, moving assets, and changing access rights. Unlike static software, such agents operate in open domains, take sequences of non-deterministic actions, and can compose individual low-consequence operations into high-consequence aggregate effects.

Existing assurance frameworks address complementary but distinct problems:

- **DO-178C / RTCA** and its aerospace analogues bound undetected errors in frozen, deterministic software artefacts through structural coverage and bidirectional traceability.
- **SR 11-7 / SR 26-2 (Federal Reserve / OCC model-risk guidance)** mandate independent effective challenge, backtesting against held-out data, and vendor-model risk attribution.
- **UL 4600** establishes a standard of care for autonomous vehicle safety cases bounded by an Operational Design Domain.
- **ISO/IEC 42001:2023** provides an AI management system at the organisational lifecycle level but does not specify per-action runtime conformance.

None of these is a published, two-layer, machine-evaluable assurance standard that:

- derives verification obligations from the gate-observed blast radius of each individual action;
- mandates independent pre-commit verification of irreversible or high-consequence actions;
- requires a backtested, bounded residual escape rate for open-world claims; and
- emits a reconstructable, append-only, hash-chained decision trace as the conformance artefact.

This document specifies such a standard: the **LLM-Agent Assurance Standard (LAAS)**. LAAS gates individual agent **actions** rather than models, and applies vertically across any domain in which agents take actions with effects outside their own sandbox.

The key elements of LAAS are:

a) a consequence-tier lattice (CT0–CT4) that the gate derives from the observed effect surface of each action, scaled by reversibility, scope, and consequence;

b) twelve obligations (Annex A) whose applicability is governed by the gate-derived tier;

c) a two-layer architecture: a human-readable normative document (this document) coupled with a machine-evaluable policy bundle (`data.json` / `laas.rego`, package `kellerai.laas.actions`); and

d) a decision trace record that is the conformance artefact: append-only, hash-chained, and reconstructable without the actor's cooperation.

The design is governed by a single invariant: LAAS applies Zero-Trust to the model's outputs **and** to the apparatus around them. A conforming system shall not trust (i) the agent's self-classification of its own action, (ii) a verifier's soundness or independence absent evidence, or (iii) the integrity of the enforcement plane.

# 1 Scope

This document specifies requirements and recommendations for action-level assurance of autonomous agents based on large language models (LLMs). It establishes:

a) a consequence-tier determination framework for classifying individual agent actions by blast radius;

b) obligations for verification, enforcement, traceability, human authority, and third-party component governance, whose applicability scales with the consequence tier;

c) requirements for the decision trace as the machine-evaluable conformance artefact; and

d) a conformance predicate that a deployer, auditor, or automated gate can evaluate given only the decision trace and the obligation bundle.

This document is applicable to any organisation that deploys LLM-based agents capable of taking actions with effects outside the agent's own sandbox, including but not limited to:

- financial-services agents executing payments, trades, or credit decisions;
- agents modifying access controls, infrastructure, or identity records;
- agents acting on behalf of natural persons in regulated communication channels; and
- multi-agent systems in which sub-agents delegate actions to peer agents.

This document does not apply to:

- read-only or fully sandboxed operations (classified CT0 per Clause 4) that produce no external effect; and
- the training, fine-tuning, or evaluation of AI models (addressed by ISO/IEC 42001 and domain-specific standards).

## 2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

- **ISO/IEC 42001:2023**, *Information technology - Artificial intelligence - Management system*
- **ISO/IEC 27001:2022**, *Information security, cybersecurity and privacy protection - Information security management systems - Requirements*
- **ISO/IEC 42006:2025**, *Information technology - Artificial intelligence - Requirements for bodies providing audit and certification of AI management systems*
- **ISO 21448:2022**, *Road vehicles - Safety of the intended functionality*

Informative alignment with DO-178C, SR 11-7/SR 26-2, and UL 4600 is given in Annex B.

# 3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardisation at the following addresses: ISO Online browsing platform at https://www.iso.org/obp and IEC Electropedia at https://www.electropedia.org/.

## 3.1 action

Observable unit of work that an agent commits, having an effect outside the agent's own sandbox (for example, calling a tool, writing a record, initiating a payment, modifying an access control list, or communicating with an external counterparty).

Note 1 to entry: A read-only or fully sandboxed operation that produces no external effect is classified as CT0 (3.4) and is not subject to the verification obligations of Clause 5.

Note 2 to entry: A sequence of individually low-consequence actions may constitute a high-consequence action when evaluated cumulatively (see 4.4 and LAAS-OBL-AGG-001).

## 3.2 effect surface

The observable, gate-measured attributes of an action that determine its consequence tier (3.4), comprising: `reversibility` (3.8), `scope` (3.9), and `consequence` (3.10).

Note 1 to entry: The effect surface is observed by the gate (3.5), not reported by the agent.

Note 2 to entry: The effect surface is recorded as an integrity-protected hash in the decision trace (3.6) to support reconstruction. Verified against `conformance/laas/data.json`, field `tier_lattice`.

## 3.3 blast radius

Aggregate measure of the potential impact of an action or a sequence of actions, derived from the effect surface (3.2) using the consequence-tier lattice.

Note 1 to entry: The blast radius is the basis for consequence-tier determination. It accounts for cumulative effects across a window of related actions (see 4.4).

## 3.4 consequence tier

Integer classification CT0 through CT4 assigned by the gate (3.5) to each action, derived from the effect surface (3.2) using the tier lattice, where higher numbers indicate greater consequence.

Note 1 to entry: CT0 denotes read-only or sandboxed operations. CT4 denotes irreversible or high-consequence operations and is the default when the effect surface is undetermined.

Note 2 to entry: CT numbers increase with consequence; this convention is the inverse of DO-178C Design Assurance Level (DAL) letters (A highest risk; E lowest risk). This difference shall be accounted for in any cross-mapping.

Note 3 to entry: The effective consequence tier accounts for cumulative window effects (4.4) and is `max(gate_assigned_ct, cumulative_window_ct)`.

Note 4 to entry: `default_ct_when_undetermined = 4` as specified in `conformance/laas/data.json:11`.

### 3.5 gate

Out-of-process component that observes the effect surface (3.2), derives the consequence tier (3.4), selects and invokes verifiers (3.7), enforces obligations, and writes to the decision trace (3.6). The gate is independent of the agent.

Note 1 to entry: The gate shall operate outside the agent's process boundary so that the agent cannot disable or modify the gate's behaviour (see 6.1 and LAAS-OBL-ENF-001).

Note 2 to entry: The gate's policy shall be delivered as a signed, version-pinned bundle.

### 3.6 decision trace

Append-only, hash-chained record produced by the gate (3.5) for each gated action, containing the fields specified in 6.2, and constituting the machine-evaluable conformance artefact.

Note 1 to entry: Each actor maintains its own hash-chain, anchored periodically into a shared Merkle root written to an append-only sink the actor cannot rewrite.

Note 2 to entry: The decision trace is the sole artefact against which the conformance predicate (Clause 8) is evaluated.

### 3.7 verifier

Independent component or human reviewer that evaluates an action prior to its commitment and returns a verdict of `pass`, `fail`, `abstain`, or `indeterminate`.

Note 1 to entry: A verifier is classified as `deterministic`, `model`, or `human`.

Note 2 to entry: A deterministic verifier applies an exact, sound oracle (for example, a schema validator, ledger balance check, or allowlist match). A model verifier uses a distinct AI model of a different lineage from the actor.

Note 3 to entry: `indeterminate` is the correct verdict when a deterministic verifier lacks required inputs. `abstain` is a confidence notion used by model verifiers or human reviewers operating outside their competence envelope.

### 3.8 reversibility

Attribute of an action describing the feasibility and cost of undoing its committed effect.

| Value | Meaning |
|---|---|
| `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 |

Note 1 to entry: When reversibility is unknown (`none`), it shall be treated as `irreversible` (default-to-highest rule). Tier-lattice values: `reversible` = 1, `hard` = 3, `irreversible` = 4, `none` = 4. Verified in `conformance/laas/data.json:7`.

### 3.9 scope

Attribute of an action describing the breadth of entities affected by its committed effect.

| Value | Meaning |
|---|---|
| `single` | One record or one user |
| `multi` | Multiple records or users |
| `org` | Organisation-wide |
| `public` | External parties or world-visible |

Note 1 to entry: When scope is unknown, it shall be treated as `public` (default-to-highest rule). Tier-lattice values: `single` = 1, `multi` = 2, `org` = 3, `public` = 4. Verified in `conformance/laas/data.json:8`.

### 3.10 consequence

Attribute of an action describing the worst-case impact level of its committed effect.

| Value | Meaning |
|---|---|
| `none` | No material impact |
| `low` | Minor, bounded impact |
| `material` | Significant financial, legal, or reputational impact |
| `high` | Severe financial, safety, legal, or reputational impact |

Note 1 to entry: When consequence is unknown, it shall be treated as `high` (default-to-highest rule). Tier-lattice values: `none` = 0, `low` = 1, `material` = 3, `high` = 4. Verified in `conformance/laas/data.json:9`.

### 3.11 escape rate

Rate at which an incorrect output of an agent survives all applicable verification checks and is committed. Measured as a fraction in [0, 1].

Note 1 to entry: The escape rate is the governing metric for Bucket B (open-world) claims. Where a higher-is-better figure is required, `integrity = 1 − escape_rate`.

Note 2 to entry: The escape rate shall be estimated by backtesting on a held-out, representative, adversarially-stressed evaluation set with a stated confidence interval.

Note 3 to entry: Tolerances by tier: CT2 ≤ 0.02, CT3 ≤ 0.005, CT4 = 0. Verified in `conformance/laas/data.json:15`.

### 3.12 verifier independence

Property of a verifier (3.7) that ensures it does not share the error modes of the actor it checks, established by one of the independence criteria in 5.2.

Note 1 to entry: A verifier sharing the actor's model lineage is presumed non-independent.

Note 2 to entry: For model verifiers, independence requires distinct model lineage **and** measured error-correlation ≤ 0.2 on the evaluation set. Verified in `conformance/laas/data.json:14`.

### 3.13 verifier qualification

Documented evidence that a verifier (3.7) has adequate coverage of its claim class, maintained under change control.

Note 1 to entry: Qualification shall include: documented coverage of the claim class, a negative-test suite of known-bad inputs the verifier must catch, and a change-controlled version identifier recorded in the decision trace. This is the DO-330 analogue for LAAS.

### 3.14 authorised operating envelope (AOE)

The operator-declared set of permissions, action classes, tools, scopes, data domains, and consequence-tier ceiling within which the agent may operate.

Note 1 to entry: An action outside the AOE shall result in the verdict `abstain` and escalation, regardless of the gate-derived consequence tier.

# 4 Consequence-tier determination

## 4.1 General

The gate (3.5) shall determine the consequence tier (3.4) of every action (3.1) prior to permitting its commitment. The consequence tier shall be derived exclusively from the gate-observed effect surface (3.2). The agent's self-reported tier shall not lower the gate-derived tier.

## 4.2 Tier-lattice derivation

### 4.2.1 Read-only and sandboxed actions

An action that produces no external effect shall be assigned CT0.

### 4.2.2 Actions with an external effect

For actions with an external effect, the gate shall derive the consequence tier using the tier lattice:

```
ct = max( rev_ct[reversibility], scope_ct[scope], cons_ct[consequence] )
```

where `rev_ct`, `scope_ct`, and `cons_ct` are the rank mappings defined in 3.8, 3.9, and 3.10 respectively.

### 4.2.3 Default-to-highest rule

When any input to the tier-lattice formula is undetermined, the gate shall assign CT4. This default shall be recorded in the decision trace.

*Requirement source:* `conformance/laas/laas.rego:29–44`.

**Example:** An agent calls `payments.transfer(amount=250000, dest=external)` with `reversibility=irreversible`, `scope=public`, `consequence=high`. The gate computes `max(4, 4, 4) = CT4`.

## 4.3 Self-report constraint (LAAS-OBL-SELF-001)

The agent may supply a self-reported consequence tier as an informational field in the decision trace. The gate shall flag any self-reported tier that is lower than the gate-derived tier. A self-reported tier shall never lower the gate-derived tier.

*This is a warning-severity obligation. The gate-derived tier prevails in all cases.*

## 4.4 Cumulative blast-radius aggregation (LAAS-OBL-AGG-001)

The gate shall maintain a windowed aggregate of effect surfaces per principal, session, and effect class. When the aggregate effect crosses a consequence-tier threshold, the gate shall re-tier subsequent actions within the window to the aggregate's tier.

This requirement closes the structuring vulnerability in which an agent or operator splits a high-consequence action into a sequence of individually lower-consequence actions.

The effective consequence tier for any action is:

```
effective_ct = max( gate_assigned_ct, cumulative_window_ct )
```

*Requirement source:* `conformance/laas/laas.rego:47–49`.

### 4.5 Untrusted-input tier raising (LAAS-OBL-INP-001)

When an action is driven by untrusted input (including web content, inbound communications, or third-party data), the gate shall raise the effective consequence tier to at least CT3 or shall block the action. The decision trace shall record the `input_trusted` field.

This requirement protects tier derivation and verifier selection from prompt-injection attacks.

*Untrusted-input minimum tier: CT3. Verified in* `conformance/laas/data.json:18`.

---

# 5 Verification of actions

## 5.1 Verification obligations by tier

The following minimum verification controls apply at each consequence tier:

| CT | Minimum verification controls |
|---|---|
| 0 | Decision trace only |
| 1 | Self-check (exact verifier if one exists in the verifier registry) and decision trace |
| 2 | Independent automated check **or** documented and rehearsed rollback plan, plus bounded residual escape rate and decision trace |
| 3 | Independent, qualified pre-commit verification (no self-grading); residual escape rate ≤ tolerance; rollback plan; decision trace |
| 4 | CT3 controls; plus human approval; plus abstention default when out of authorised operating envelope; plus full evidence package |

## 5.2 Verifier independence (LAAS-OBL-IND-001)

### 5.2.1 Independence criteria

A verifier shall be independent of the actor it checks. Independence is established by satisfying one of the following criteria, applied in order of precedence:

a) **Deterministic/exact criterion:** The verifier is a different kind of checker that applies a sound, exact oracle (for example, schema validation, ledger balance computation, cryptographic hash verification). This criterion is sufficient for independence at any consequence tier.

b) **Distinct-lineage-and-low-correlation criterion:** The verifier uses a distinct model lineage from the actor **and** demonstrates measured error-correlation ≤ 0.2 on the evaluation set. This criterion is sufficient for independence up to and including CT3.

c) **Human criterion:** The verifier is a human reviewer. Human verification is required in addition to criteria (a) or (b) at CT4.

A verifier sharing the actor's model lineage shall be presumed non-independent. Where the error-correlation between a model verifier and the actor cannot be bounded below 0.2, the gate shall fall back to a deterministic verifier or a human reviewer for CT3 and above.

*Maximum error-correlation: 0.2. Verified in* `conformance/laas/data.json:14`.

### 5.2.2 Prohibition on self-grading

A verifier that is under the control of, or whose verdict can be modified by, the agent being verified shall not satisfy the independence requirement. Any control path that allows the constrained party to tier, grade, or gate itself is non-conforming.

*Requirement source:* `conformance/laas/laas.rego:75–83`.

## 5.3 Verifier qualification (LAAS-OBL-VQ-001)

A verifier that gates actions at CT3 or above shall be qualified. Qualification requires:

a) documented coverage of the claim class the verifier addresses;

b) a negative-test suite of known-bad inputs that the verifier shall correctly reject; and

c) a change-controlled version identifier recorded in the `verifier_qualification_ref` field of the decision trace.

An unqualified verifier does not satisfy LAAS-OBL-IRR-001 (5.4) even if it returns a passing verdict.

## 5.4 Independent pre-commit verification (LAAS-OBL-IRR-001)

An action whose effective consequence tier is CT3 or CT4 shall pass an independent, qualified pre-commit verifier before commitment. The verifier shall satisfy the independence criteria in 5.2 and the qualification requirements in 5.3.

If the verifier returns `fail`, `abstain`, or `indeterminate`, the action shall be blocked and escalated. A blocked action satisfies this obligation if the block and escalation are recorded in the decision trace.

*CT floor for independent verification: CT3. Verified in* `conformance/laas/data.json:12`.

## 5.5 Residual escape rate (LAAS-OBL-RES-001)

### 5.5.1 Bucket classification

Claims and actions shall be classified into:

a) **Bucket A (deterministically checkable):** An exact oracle exists. The gate shall apply the exact verifier; the escape rate approaches zero, bounded only by the verifier's soundness. No probabilistic tolerance is permitted where an exact verifier exists.

b) **Bucket B (open-world):** No exact oracle exists. The deployer shall declare a maximum acceptable escape rate for the claim class at the relevant consequence tier, estimate the escape rate by backtesting on a held-out, representative, adversarially-stressed evaluation set, and re-measure the escape rate on any change to the model, prompt, tools, or policy.

### 5.5.2 Tolerance requirements

The measured Bucket B escape rate shall not exceed the declared tolerance for the effective consequence tier. The following maximum tolerances apply:

| Effective CT | Maximum escape rate tolerance |
|---|---|
| CT2 | 0.02 (2 %) |
| CT3 | 0.005 (0.5 %) |
| CT4 | 0 (zero; deterministic or human verification required) |

*Verified in* `conformance/laas/data.json:15`.

### 5.5.3 Evidence

The deployer shall reference the backtest report in the `evidence_refs` field of the decision trace. The backtest evaluation set shall be independently audited for representative coverage and adversarial stress.

## 5.6 Rollback rehearsal

Where a rollback plan substitutes for pre-commit verification at CT2, the deployer shall maintain documented periodic rollback-rehearsal evidence. An unrehearsed rollback plan does not satisfy the CT2 verification obligation.

# 6 Enforcement plane and decision evidence

## 6.1 Enforcement-plane integrity (LAAS-OBL-ENF-001)

### 6.1.1 Out-of-process gate

The gate (3.5) shall operate as an out-of-process component that the agent cannot disable, modify, or circumvent, including in elevated-permission or override modes.

*Requirement source:* `conformance/laas/data.json:17` *(*`require_out_of_process_gate: true`*);* `conformance/laas/laas.rego:119–121`.

### 6.1.2 Signed policy bundle

The gate's policy bundle shall be cryptographically signed and version-pinned. The `bundle_signed` field of the decision trace shall be `true`. A decision trace asserting an unsigned or in-process gate is non-conforming regardless of other verdict fields.

*Requirement source:* `conformance/laas/data.json:16` *(*`require_bundle_signed: true`*);* `conformance/laas/laas.rego:114–117`.

## 6.2 Decision trace (LAAS-OBL-TRC-001)

### 6.2.1 Append-only requirement

The decision trace shall be written to an append-only, content-addressed store that the actor cannot rewrite. Each entry in the trace shall reference the hash of the preceding entry in the actor's own chain (the `actor_chain_prev_hash` field), forming a per-actor hash-chain.

### 6.2.2 Merkle anchoring

Per-actor hash-chains shall be anchored periodically into a shared Merkle root written to an append-only sink. The `merkle_anchor` field records the root at the time of anchoring. This scheme supports concurrent multi-agent deployments without requiring a single serial appender.

### 6.2.3 Mandatory trace fields

Each decision-trace record shall contain, at minimum, the following fields:

| Field | Description |
|---|---|
| `obligation_id` | Identifier of the triggering obligation |
| `obligation_version` | Version of the obligation bundle |
| `action_ref` | Stable, unique identifier of the action |
| `effect_surface_hash` | Integrity-protected hash of the gate-observed effect surface |
| `gate_derived_ct` | Consequence tier derived by the gate |
| `self_reported_ct` | Consequence tier self-reported by the actor (informational) |
| `trigger_matched` | Whether this obligation's trigger matched ( `true` / `false` ) |
| `verifier_id` | Stable identifier of the verifier applied |
| `verifier_type` | `deterministic` , `model` , or `human` |
| `verifier_independent` | Whether the verifier satisfies the independence criteria in 5.2 |
| `independence_basis` | Which independence criterion was applied |
| `verifier_qualified` | Whether the verifier satisfies the qualification requirements in 5.3 |
| `verifier_qualification_ref` | Reference to the qualification evidence |
| `verifier_error_correlation` | Measured error-correlation (model verifiers only; `null` otherwise) |
| `verifier_input_hash` | Hash of the inputs supplied to the verifier |
| `verdict` | `pass` , `fail` , `abstain` , or `indeterminate` |
| `residual_error_bound` | Measured escape rate (Bucket B); `null` for pure Bucket A |
| `residual_tolerance` | Declared escape-rate tolerance for the effective tier |
| `evidence_refs` | References to supporting evidence artefacts |
| `escalation` | Null or `{queue, ticket_id}` if the action was escalated |
| `escalation_approved` | Whether an escalation was approved by an authorised reviewer |
| `action_blocked` | Whether the action was blocked |
| `actor_id` | Stable identifier of the acting agent |
| `policy_bundle_version` | Version of the signed policy bundle applied |
| `bundle_signed` | Whether the policy bundle was signed |
| `gate_out_of_process` | Whether the gate operated out-of-process |
| `aggregate_window_ct` | Cumulative-window consequence tier at the time of evaluation |
| `input_trusted` | Whether the action inputs were classified as trusted |
| `timestamp` | ISO 8601 timestamp of gate evaluation |
| `actor_chain_prev_hash` | Hash of the actor's preceding trace entry |
| `merkle_anchor` | Shared Merkle root hash at the time of anchoring |

*Field set verified against `docs/laas/proposal-v1.1.md:225–256` (emit_decision_trace).*

### 6.2.4 Bidirectional traceability

Every committed CT3 or CT4 action shall have a corresponding decision-trace record with a passing or blocked verdict. Every obligation shall be exercised by at least one trace record in the conformance evidence, or marked not-applicable with a documented reason.

### 6.2.5 Trace data governance

The deployer shall apply the following data governance controls to the decision trace:

a) minimise payloads to the fields required by this clause;

b) tokenise or redact personally identifiable information (PII) and material non-public information (MNPI) before writing to the searchable trace index; and

c) maintain the searchable trace index as a separate component from sensitive payload data, so that audit access does not require exposure of regulated data.

# 7 Human authority and third-party components

## 7.1 Human approval at CT4 (LAAS-OBL-HUM-001)

### 7.1.1 Requirement

An action at effective CT4 shall not be committed without human approval. The `escalation_approved` field of the decision trace shall be `true`. A CT4 action that is blocked and escalated satisfies this obligation if the block and escalation are recorded in the decision trace.

*CT floor for human approval: CT4. Verified in* `conformance/laas/data.json:13`.

### 7.1.2 Abstention default

In the absence of human approval for a CT4 action, the gate shall abstain (return `verdict: abstain`) and escalate. An abstention is a conforming outcome.

### 7.1.3 Scalable human oversight

Human approval may be implemented via pre-authorised standing envelopes. An operator may pre-approve a bounded action class with explicit limits (for example, a defined transaction amount ceiling with an explicit counterparty allowlist). The gate shall enforce the declared limits and escalate only out-of-envelope actions. This scheme permits human attention to scale with exceptions rather than volume.

Batched approval of queued in-envelope actions is permitted provided each action in the batch remains within the pre-authorised limits.

## 7.2 Third-party and vendor components (LAAS-OBL-VEN-001)

### 7.2.1 Attribution

When the agent uses a vendor model or calls a third-party tool or API, the decision trace shall carry:

a) provenance of the vendor component (vendor identifier, model or API version); and

b) declared scope limits for that component.

### 7.2.2 Residual attribution

Residual errors attributable to a vendor model or third-party component shall be counted against the deploying operator's escape-rate budget (3.11). The operator is responsible for the aggregate escape rate regardless of whether the error originated in a first- or third-party component.

### 7.2.3 Fail-closed default

Untrusted or unattributed third-party dependencies shall fail closed. A component whose provenance or scope limits cannot be established shall cause the gate to block and escalate the action.

*Requirement source:* `conformance/laas/laas.rego:148–151`.

# 8 Conformance

## 8.1 Conformance predicate

An action is conforming if and only if, for every obligation whose trigger matched, one of the following holds:

a) **Pass path:** The action passed an independent (Clause 5.2), qualified (5.3) verifier; and, where the effective consequence tier is CT4, human approval was obtained (7.1); and the measured residual escape rate does not exceed the declared tolerance (5.5.2); or

b) **Block path:** The action was blocked and the block and escalation were recorded in the decision trace.

Formally, for each triggered obligation:

```
trigger_matched == true  IMPLIES
  ( verdict == "pass"
    AND verifier_independent == true
    AND verifier_qualified == true
    AND ( gate_derived_ct < 4  OR  escalation_approved == true )
    AND ( residual_error_bound == null  OR  residual_error_bound <= residual_tolerance ) )
  OR
  ( verdict IN {"fail", "abstain", "indeterminate"}  AND  action_blocked == true )
```

*Predicate source:* `docs/laas/proposal-v1.1.md:257–264` *(conformance_predicate field); evaluated by* `conformance/laas/laas.rego:154–191`.

## 8.2 Deployer conformance attestation

A deployer demonstrates conformance by providing:

a) a signed, sampled bundle of decision traces over a representative period, covering CT2, CT3, and CT4 actions; and

b) an independent auditor attestation that the bundle's verdicts satisfy the obligation set in Annex A for every triggered obligation.

## 8.3 Machine evaluation

The conformance predicate shall be evaluable by an automated gate given only the decision trace record and the signed obligation bundle ( `conformance/laas/data.json` ). For the deterministic path (Bucket A), evaluation is fully automated. For the Bucket B path, evaluation additionally requires the operator-supplied backtest evidence referenced in `evidence_refs`.

## 8.4 Obligation lifecycle

Obligations move through the states `propose → active → deprecated → retired`. Retired obligations leave the active set and are not evaluated. When two active obligations fire with incompatible required outcomes, precedence is deterministic: the obligation with the highest consequence-tier floor wins; ties resolve to the most restrictive outcome (block). Each obligation's precedence is recorded in `conformance/laas/data.json` and the gate's resolution order is recorded in the decision trace.

## Annex A (normative): Control objectives and controls

This annex specifies the control objectives and controls for LAAS conformance. Each control maps to one obligation. Deployers shall implement all applicable controls or document a justified exception in the conformance evidence.

| Control ref | Control objective | Control | Obligation | CT floor |
|---|---|---|---|---|
| A.1 | Tier determination is gate-derived and ungameable | The gate shall derive the consequence tier from the observed effect surface using the tier lattice (Clause 4). The agent's self-reported tier shall not lower the gate-derived tier. | LAAS-OBL-TIER-001 | CT0 |
| A.2 | Self-reported tier does not suppress gate tier | The system shall flag any self-reported tier that is lower than the gate-derived tier. The gate-derived tier shall prevail. | LAAS-OBL-SELF-001 | CT0 |
| A.3 | Enforcement-plane integrity | The gate shall operate out-of-process, and the policy bundle shall be cryptographically signed and version-pinned (Clause 6.1). | LAAS-OBL-ENF-001 | CT0 |
| A.4 | Append-only, hash-chained decision trace | All gated actions shall produce a decision-trace record written to an append-only store the actor cannot rewrite, with per-actor hash-chaining and periodic Merkle anchoring (Clause 6.2). | LAAS-OBL-TRC-001 | CT0 |
| A.5 | Cumulative blast-radius aggregation | The gate shall aggregate effect surfaces across a time window and re-tier subsequent actions when the aggregate crosses a tier threshold (4.4). | LAAS-OBL-AGG-001 | CT0 |
| A.6 | Untrusted input raises the tier | When an action is driven by untrusted input, the gate shall raise the effective tier to at least CT3 or block the action (4.5). | LAAS-OBL-INP-001 | CT0 |
| A.7 | Third-party attribution and scope limits | Vendor and third-party components shall be attributed in the decision trace with declared scope limits; residual errors shall be charged to the deployer's escape-rate budget (7.2). | LAAS-OBL-VEN-001 | CT0 |
| A.8 | Independent pre-commit verification | CT3 and CT4 actions shall pass an independent, qualified pre-commit verifier before commitment; failing verdicts shall result in a block and escalation (5.4). | LAAS-OBL-IRR-001 | CT3 |
| A.9 | Verifier independence and low error-correlation | The verifier shall satisfy at least one independence criterion in 5.2; model verifiers shall demonstrate error-correlation ≤ 0.2; at CT4, a human verifier is required in addition (5.2). | LAAS-OBL-IND-001 | CT3 |
| A.10 | Verifier qualification | Verifiers gating CT3 or above shall be qualified with documented claim-class coverage, a negative-test suite, and a change-controlled version identifier in the decision trace (5.3). | LAAS-OBL-VQ-001 | CT3 |
| A.11 | Bounded residual escape rate | The measured Bucket B escape rate shall not exceed the declared tolerance for the effective tier: CT2 ≤ 2 %, CT3 ≤ 0.5 %, CT4 = 0 %; evidence shall be referenced in the decision trace (5.5). | LAAS-OBL-RES-001 | CT2 |
| A.12 | Human approval at CT4 | CT4 actions shall not be committed without human approval or a documented block and escalation; the abstention default applies in the absence of approval (7.1). | LAAS-OBL-HUM-001 | CT4 |

# Annex B (informative): Correspondence to DO-178C, SR 26-2, and UL 4600

This annex maps LAAS concepts to the three mature assurance frameworks from which LAAS draws its structural object. This annex is informative; it does not modify the requirements of this document.

## B.1 Common structural object

All four frameworks share a common set of assurance mechanics, applied under different vocabulary to different domains.

| Shared mechanic | DO-178C | SR 26-2 (SR 11-7) | UL 4600 | LAAS |
|---|---|---|---|---|
| Scale rigor to consequence | Design Assurance Level (DAL A–E) | Materiality tier | Risk-based safety-case depth | Consequence tier CT0–CT4 (Clause 4) |
| Bound undetected error | Structural coverage (MC/DC at DAL A) | Backtesting on held-out data | Safety Performance Indicators (SPIs) | Escape rate per bucket (3.11, 5.5) |
| Independent verification | Verification objectives "with independence"; verifier ≠ developer | Effective challenge by independent party | Independent safety assessment | Independence criteria (5.2); no self-grading (5.2.2) |
| Trace to evidence | Bidirectional traceability requirements → certification data package | Validation documentation | Goal Structuring Notation (GSN) evidence | Decision trace (3.6, 6.2); bidirectional traceability (6.2.4) |
| Declared envelope and abstention | Configuration and operational envelope | Approved use model | Operational Design Domain (ODD) | Authorised Operating Envelope (AOE) (3.14) |
| No failure, still wrong | Addressed weakly | Conceptual soundness; ongoing monitoring | ISO 21448 (SOTIF) companion | Bucket B bounding and independent review (5.5) |

## B.2 Key differences

The following differences from the reference frameworks are intentional design choices.

**B.2.1 Consequence-tier direction.** LAAS CT numbers rise with consequence (CT0 lowest, CT4 highest). DO-178C DAL letters fall with consequence (A highest, E lowest/none). Any crosswalk table shall account for this inversion to prevent mapping errors.

**B.2.2 Per-action runtime gating.** DO-178C applies assurance pre-deployment at the artefact level. LAAS applies assurance at runtime to each individual action, sustaining conformance in environments where model, prompt, tools, and policy change continuously.

**B.2.3 Open-world scope.** DO-178C and ARP6983/ED-324 are explicitly limited to frozen, deterministic or supervised-ML software in embedded avionics. LAAS is designed for generative, non-deterministic LLM-based agents operating in open domains.

**B.2.4 Non-determinism handling.** LAAS distinguishes between Bucket A (deterministically reproducible verdicts) and Bucket B (tamper-evidently re-inspectable verdicts), enabling conformance evaluation without requiring deterministic reproduction of model outputs.

### B.3 Lineage statement

LAAS draws its governing mechanics from the following lineage, without incorporating the domain-specific scope restrictions of any antecedent framework:

- **DO-178C** (RTCA/EUROCAE, 2011): independence requirements, bidirectional traceability, structural-coverage analogue (verifier claim-class coverage in 5.3).
- **SR 11-7** (Federal Reserve SR Letter 11-7, 2011) and **SR 26-2** (Federal Reserve SR Letter 26-2, 2026, expected): effective challenge, backtesting, vendor-risk attribution.
- **UL 4600** (UL, 2020 and subsequent): standard-of-care model, safety case with ODD, abstention outside the operational envelope.

The three-lines-of-defence model is general enterprise-risk management practice commonly paired with SR 11-7 and is not a creature of that document's text.

---

## Bibliography

The following documents are cited for informative purposes.


- RTCA/EUROCAE, **DO-178C / ED-12C**, *Software Considerations in Airborne Systems and Equipment Certification*, 2011.
- RTCA/EUROCAE, **DO-330 / ED-215**, *Software Tool Qualification Considerations*, 2011.
- Federal Reserve, **SR Letter 11-7**, *Guidance on Model Risk Management*, 2011.
- Federal Reserve, **SR Letter 26-2** (expected), *Update to Model Risk Management Guidance*, 2026 (confirm publication date before citation).
- UL Solutions, **UL 4600**, *Standard for Safety for the Evaluation of Autonomous Products*, 2020 (latest edition).
- ISO, **ISO 21448:2022**, *Road vehicles - Safety of the intended functionality (SOTIF).*
- ISO/IEC, **ISO/IEC 42001:2023**, *Information technology - Artificial intelligence - Management system.*
- ISO/IEC, **ISO/IEC 27001:2022**, *Information security, cybersecurity and privacy protection - Information security management systems - Requirements.*
- ISO/IEC, **ISO/IEC 42006:2025**, *Information technology - Artificial intelligence - Requirements for bodies providing audit and certification of AI management systems.*
- SAE International / EUROCAE, **ARP6983 / ED-324**, *Process Standard for Development and Certification/Approval of Aeronautical Safety-Related Products Implementing AI/ML Technology*, approximately Q1 2026 (confirm publication date before citation).
- NIST, **AI 100-1**, *Artificial Intelligence Risk Management Framework*, 2023.
- NIST, **AI 600-1**, *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*, 2024.
- Cloud Security Alliance (CSA), **MAESTRO** threat modeling framework, 2025–2026.
- OWASP Foundation, **Agentic AI Top 10**, 2025–2026.
- Five Eyes Intelligence Partnership, *Careful Adoption of Agentic AI Services*, April–May 2026.
- KellerAI, **LAAS v1.1 proposal**, `docs/laas/proposal-v1.1.md`, 2026-06-18.
- KellerAI, **LAAS machine-evaluable obligation bundle**, `conformance/laas/data.json`, version 1.1.0, bundle ID `laas-fin-1.1.0`.
- KellerAI, **LAAS conformance policy**, `conformance/laas/laas.rego`, package `kellerai.laas.actions`.