NIST-style draft · SP-XXXX (Unofficial)

Action-Level Assurance for Autonomous AI Agents

A NIST-style control profile for the LLM-Agent Assurance Standard (LAAS)

KellerAI Open-Source Working Group

Consequence tiers · control catalog · AI RMF crosswalk

2026-06-22

Action-Level Assurance for Autonomous AI Agents: A NIST-Style Control Profile (LAAS)

Designation: LAAS-NIST-PROFILE-DRAFT-1.1 **Document type:** Control profile (NIST AI RMF crosswalk and SP 800-style control catalog) **Source standard:** LLM-Agent Assurance Standard (LAAS) v1.1, `standard/LAAS.md` **Machine source of truth:** `conformance/laas/data.json` (bundle `laas-fin-1.1.0`) **Enforcing policy:** `conformance/laas/laas.rego`, package `kellerai.laas.actions` **Status:** Draft, not approved; all thresholds cite `conformance/laas/data.json`

> **Disclaimer:** This is not an official NIST publication. It is a community-produced control profile that maps the LLM-Agent Assurance Standard (LAAS) to the NIST AI Risk Management Framework (AI RMF 1.0) and adopts SP 800-series control-register format to aid interoperability with existing federal and enterprise risk programs. No endorsement by NIST or any other standards body is expressed or implied.

Abstract

Autonomous AI agents commit actions (writing records, moving money, modifying access) at a rate and scale that exceed traditional governance checkpoints. Existing frameworks govern model providers, management systems, or periodic audits; none publishes a machine-checkable, per-action conformance standard that derives verification obligations directly from an action's observed blast radius.

This profile presents LAAS v1.1 in NIST control-register format and maps it to the four functions of the NIST AI Risk Management Framework (AI RMF 1.0): GOVERN, MAP, MEASURE, and MANAGE. LAAS gates individual agent actions by **Consequence Tier (CT0–CT4)**, computed by an out-of-process gate from the observed effect surface (reversibility, scope, consequence). Twelve normative obligations govern tier derivation, enforcement-plane integrity, cumulative blast-radius aggregation, input provenance, verifier independence, verifier qualification, residual escape rate, and human approval. An append-only, hash-chained decision trace is the conformance artifact.

Keywords: AI agents, action gating, consequence tier, escape rate, independent verification, decision trace, OPA, policy-as-code, AI RMF, DO-178C, SR 11-7, UL 4600

Audience: AI deployers, enterprise risk and compliance teams, government assessors, financial-services model-risk officers, software auditors, and AI agent implementers.

---

1. Introduction

1.1 Purpose and Scope

This profile serves two purposes. First, it renders the LAAS v1.1 normative obligations in SP 800-style control format so that organizations using NIST-aligned governance programs can adopt LAAS controls directly into their existing assessment frameworks. Second, it provides a crosswalk from each LAAS mechanism to the NIST AI RMF (AI 100-1, 2023) function and subcategory it instantiates, enabling organizations to demonstrate AI RMF alignment through LAAS conformance.

Scope. LAAS governs **individual actions taken by LLM-based agents** (tool calls, record writes, financial transfers, access changes) wherever those actions have an effect outside the agent's own sandbox (`standard/LAAS.md §1`). LAAS does not certify a model, a training pipeline, or a management system. It gates what an agent is allowed to commit.

This profile does not modify LAAS normative text. Where this profile and `standard/LAAS.md` conflict, the standard takes precedence. Where this profile and `conformance/laas/data.json` conflict on a threshold value, `data.json` is the machine source of truth (`standard/LAAS.md`, preamble).

1.2 Audience

This profile is written for two audiences simultaneously.

Risk and compliance practitioners who work within NIST-aligned programs and need to map agent governance controls to existing RMF subcategories and assessment objectives.

AI agent implementers and deployers who need a machine-checkable control set that can be evaluated by an OPA policy and attested by a signed decision-trace bundle rather than by periodic audit alone.

1.3 How to Use This Profile

Read **Section 2** for a plain-language description of the LAAS approach and its key concepts. Use **Section 3** for the RMF crosswalk; it answers which LAAS mechanisms satisfy which RMF subcategories. Use **Section 4** as the normative control catalog: one control per obligation, in SP 800 format, with assessment objectives an auditor or automated gate can evaluate. Read **Section 5** for the consequence-tier derivation function and escape-rate measurement method. Use **Appendix A** for a compact control-to-RMF mapping table. Use **Appendix B** for the mapping to aerospace and financial-services source standards. Use **Appendix C** for normative and informative references.

---

2. The LAAS Approach

2.1 Action Gating

LAAS applies **Zero-Trust to the model's outputs and to the apparatus around them** (`standard/LAAS.md §2`, `docs/laas/proposal-v1.1.md §0.1`). A conforming system does not trust:

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

Because of this invariant, a party other than the actor evaluates every control in LAAS. The gate, an out-of-process policy engine running the OPA policy in `conformance/laas/laas.rego`, derives the tier, selects obligations, and emits a decision trace the actor cannot rewrite.

2.2 Consequence Tiers

The gate assigns every agent action a **Consequence Tier (CT0–CT4)**, computed from the observed effect surface and never from the agent's self-report. CT numbers rise with consequence; CT4 is the highest (the inverse of DO-178C DAL letters, where A is most severe) (`docs/laas/proposal-v1.1.md §6.1`).

The tier function (`conformance/laas/data.json → tier_lattice`):

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

Rank tables (source: `conformance/laas/data.json → tier_lattice`):

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

Any unknown input defaults to the highest rank in its axis (`docs/laas/proposal-v1.1.md §6.2`).

The **effective CT** is the maximum of the gate-assigned CT and the cumulative window CT (anti-structuring, `conformance/laas/laas.rego:47`):

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

2.3 Effect Surface and the Authorized Operating Envelope

The effect surface is the tuple `(reversibility, scope, consequence)` of the action as **observed by the gate** from the actual tool invocation, not as reported by the agent. The Authorized Operating Envelope (AOE) is the operator-declared set of permitted tools, scopes, data domains, action classes, and CT ceiling. Out-of-AOE actions produce a first-class `verdict: abstain → escalate` trace record (`docs/laas/proposal-v1.1.md §6.5`).

2.4 The Decision Trace

Every gated action emits one decision-trace record in an **append-only, content-addressed store** the actor cannot rewrite. Each actor maintains its own hash-chain (`actor_chain_prev_hash`), periodically anchored into a shared Merkle root (`merkle_anchor`) for multi-agent safety (`docs/laas/proposal-v1.1.md §7.4`). The trace is the conformance artifact: a signed, sampled trace bundle plus an independent auditor attestation constitutes the LAAS conformance attestation (`standard/LAAS.md §6`, `docs/laas/proposal-v1.1.md §11.3`).

---

3. Crosswalk to the NIST AI RMF

The NIST AI RMF (AI 100-1, 2023) organizes AI risk management into four functions (GOVERN, MAP, MEASURE, and MANAGE), each decomposed into categories and subcategories. The table below maps each LAAS mechanism to the RMF function and a representative subcategory. Section 4 provides the full normative controls; this crosswalk provides navigational orientation.

3.1 Crosswalk Table

LAAS Mechanism	Obligation(s)	RMF Function	RMF Subcategory
Signed bundle + out-of-process gate	ENF-001	GOVERN	GV-1.1 (Policies, processes, procedures)
Vendor attribution + scope limits	VEN-001	GOVERN	GV-6.1 (Organizational roles and responsibilities for third parties)
Human approval at CT4	HUM-001	GOVERN	GV-1.4 (Organizational teams empowered to escalate)
Obligation lifecycle + precedence	(all)	GOVERN	GV-1.7 (Continuous improvement)
Effect-surface derivation	TIER-001	MAP	MP-2.3 (AI system impact assessment)
Input provenance + injection resistance	INP-001	MAP	MP-2.5 (AI system context characterization)
Cumulative blast-radius aggregation	AGG-001	MAP	MP-3.5 (Aggregated risk characterization)
Self-report may not lower tier	SELF-001	MAP	MP-2.3 (AI system impact assessment)
Residual escape rate measurement	RES-001	MEASURE	MS-2.5 (AI system performance measurement)
Verifier independence + error-correlation	IND-001	MEASURE	MS-2.8 (Independence of evaluation)
Verifier qualification	VQ-001	MEASURE	MS-4.1 (Measurement practices improvement)
Append-only hash-chained trace	TRC-001	MEASURE	MS-2.6 (AI system outputs documented)
Pre-commit verification for CT≥3	IRR-001	MANAGE	MG-2.2 (Mechanisms to halt deployment)
Human approval at CT4	HUM-001	MANAGE	MG-2.4 (Mechanisms for human override)
Rollback + escalation paths	IRR-001, HUM-001	MANAGE	MG-3.2 (Risk treatment strategies)

3.2 GOVERN Function

LAAS contributes to GOVERN through enforcement-plane integrity controls (ENF-001), vendor accountability requirements (VEN-001), the human-approval gate at CT4 (HUM-001), and the obligation

lifecycle that supports continuous improvement of the control set. Together these establish the organizational policies and responsibilities that RMF GOVERN subcategories require.

3.3 MAP Function

LAAS contributes to MAP through its consequence-tier derivation function (TIER-001), which performs the impact characterization that MAP subcategories require for each AI action. The anti-structuring obligation (AGG-001) provides aggregated risk characterization: the MAP analogue of cumulative position-limit monitoring. The input-provenance obligation (INP-001) contributes context characterization by flagging untrusted input as a tier-raising signal.

3.4 MEASURE Function

LAAS provides the measurement backbone through three controls: the escape-rate tolerance and measurement protocol (RES-001), verifier independence and error-correlation requirements (IND-001, VQ-001), and the append-only decision trace (TRC-001). The escape rate measured by backtesting on a held-out adversarial set is the primary metric; `integrity = 1 − escape_rate` is a higher-is-better derived figure (`docs/laas/proposal-v1.1.md §5`).

3.5 MANAGE Function

LAAS contributes to MANAGE through the pre-commit gating mechanism for CT≥3 (IRR-001), which provides the ability to halt or block an action before it is committed. The human-approval requirement at CT4 (HUM-001) provides an explicit human-override mechanism. Escalation paths and rollback-plan requirements complete the MANAGE surface.

---

4. LAAS Control Catalog

This section presents all twelve LAAS obligations in SP 800-style control format. Each control entry includes: a Control ID, a normative Control Statement, Discussion, and Assessment / Verification Objectives.

Control IDs use the form `LAAS-<OBLIGATION-SHORT>-<N>` and correspond directly to obligation IDs in `conformance/laas/data.json → obligations`.

LAAS-TIER-1

Control ID: LAAS-TIER-1 **Obligation ID:** `LAAS-OBL-TIER-001` **Severity:** error **CT floor:** 0 (applies to all actions) **Precedence:** 100 (highest; source: `conformance/laas/data.json`)

Control Statement

The system shall derive the Consequence Tier of every agent action from the observed effect surface (the tuple (reversibility, scope, consequence) as measured by the out-of-process gate) using the tier lattice defined in `conformance/laas/data.json → tier_lattice`. The gate-assigned CT shall never be set below the lattice-derived value. The effective CT shall be the maximum of the gate-assigned CT and the cumulative window CT.

Discussion

This control is the foundational invariant of LAAS: the constrained party (the agent) cannot grade its own action. The gate observes the actual tool invocation (the amount transferred, the counterparty identified, the scope of write), not the agent's description of what it intends to do. If any axis of the effect surface is undetermined, the tier defaults to CT4 (`conformance/laas/data.json → default_ct_when_undetermined: 4`, `conformance/laas/laas.rego:29`).

The Rego predicate for this control (`conformance/laas/laas.rego:98-103`):

```
violations contains obl("LAAS-OBL-TIER-001", ...) if {
    input.gate.assigned_ct < expected_ct
}
```

Assessment / Verification Objectives

1. Verify that the gate computes `expected_ct` using the tier lattice in `data.json` for every action with an external effect.
2. Verify that `input.gate.assigned_ct >= expected_ct` for every decision-trace record.
3. Verify that `effective_ct = max(gate_assigned_ct, aggregate_window_ct)` is computed before any obligation check.
4. Verify that undetermined inputs produce `expected_ct = 4`.
5. Sample ten decision-trace records; confirm no record has `assigned_ct < expected_ct`.

LAAS-SELF-1

Control ID: LAAS-SELF-1 **Obligation ID:** `LAAS-OBL-SELF-001` **Severity:** warning **CT floor:** 0 **Precedence:** 90

Control Statement

The system shall record the agent's self-reported Consequence Tier in the decision trace and shall flag any record where the self-reported CT is lower than the gate-derived CT. The gate-derived CT shall prevail in all cases.

Discussion

This control implements the warning tier of the Zero-Trust-on-classification invariant. Self-reports are recorded for diagnostic and audit purposes (a pattern of agents persistently under-reporting their tier is a governance signal), but they cannot lower the operative tier. This is a `warning`-severity obligation; it does not block the action but does appear in the `summary.warnings` count (`conformance/laas/laas.rego:106-113`).

Assessment / Verification Objectives

1. Verify that the decision-trace schema includes `self_reported_ct` as a declared field (see `docs/laas/proposal-v1.1.md §7.2`).
2. Verify that the Rego policy emits a `LAAS-OBL-SELF-001` warning when `self_reported_ct < gate.assigned_ct`.
3. Verify that the gate's enforcement logic uses `gate_derived_ct`, not `self_reported_ct`, for all obligation-selection decisions.
4. Review audit logs for systematic under-reporting patterns by any agent identity.

LAAS-ENF-1

Control ID: LAAS-ENF-1 **Obligation ID:** `LAAS-OBL-ENF-001` **Severity:** error **CT floor:** 0 **Precedence:** 99

Control Statement

The system shall enforce every LAAS obligation through an out-of-process gate that the agent cannot disable, bypass, or modify at runtime. The policy bundle evaluated by the gate shall be cryptographically signed and version-pinned. The decision-trace sink shall be append-only; the agent shall have no write access to existing trace records.

Discussion

Mechanical enforcement is only meaningful if the mechanism is outside the actor's reach. This control establishes three interlocking requirements derived from `docs/laas/proposal-v1.1.md §7.7`:

- `bundle_signed`: the policy bundle is signed before deployment; the gate verifies the signature before evaluating any action.
- `gate_out_of_process`: the gate runs in a separate process (or network service) from the agent; the agent cannot call overrides of the `--dangerously-skip-permissions` style to circumvent it.
- Append-only sink: the trace store accepts `APPEND` but not `UPDATE` or `DELETE` on committed records.

Configuration flags driving this control (`conformance/laas/data.json`): `require_bundle_signed: true`, `require_out_of_process_gate: true`.

The two Rego checks (`conformance/laas/laas.rego:114–122`):

```
violations contains obl("LAAS-OBL-ENF-001", "policy bundle is not signed") if {
    cfg.require_bundle_signed
    not input.gate.bundle_signed
}
violations contains obl("LAAS-OBL-ENF-001", "gate is not out-of-process") if {
    cfg.require_out_of_process_gate
    not input.gate.out_of_process
}
```

Assessment / Verification Objectives

1. Verify that `input.gate.bundle_signed == true` in every decision-trace record.
2. Verify that `input.gate.out_of_process == true` in every decision-trace record.
3. Confirm via architecture review that the gate process runs outside the agent process and that the agent has no API to disable it.
4. Confirm that the trace store's access-control configuration prevents `UPDATE` and `DELETE` on committed records.
5. Attempt to inject a record with `bundle_signed: false`; confirm the policy emits a `LAAS-OBL-ENF-001` error-severity violation.

LAAS-TRC-1

Control ID: LAAS-TRC-1 **Obligation ID:** `LAAS-OBL-TRC-001` **Severity:** error **CT floor:** 0 **Precedence:** 50

Control Statement

The system shall emit a decision-trace record for every gated action. Records shall be written to an append-only, content-addressed store. Each record shall include a `actor_chain_prev_hash` linking it to the prior record in the actor's hash-chain, and the chain shall be periodically anchored to a shared Merkle root (`merkle_anchor`) written to an append-only sink. Personally identifiable information and material non-public information in trace records shall be minimized and tokenized; the searchable index shall be separable from sensitive payloads.

Discussion

The decision trace is the LAAS conformance artifact: a signed, sampled trace bundle plus an independent auditor attestation demonstrates conformance (`standard/LAAS.md §6`, `docs/laas/proposal-v1.1.md §11.3`).

The hash-chain design (`docs/laas/proposal-v1.1.md §7.4`) gives each actor its own chain, avoiding the single-appender bottleneck of a linear chain in multi-agent swarms. The Merkle-anchor step provides cross-actor tamper evidence.

A Bucket-A verdict (deterministic checker) is **deterministically reproducible** against the recorded `verifier_input_hash`. A Bucket-B verdict is **tamper-evidently re-inspectable**: the recorded verdict and

evidence references can be reviewed; the non-deterministic model verdict cannot be identically re-derived, but the evidence that supported it is available.

Data governance requirements (`docs/laas/proposal-v1.1.md §7.6`): tokenize PII/MNPI in payloads; store the searchable index separately; apply retention-by-tier (higher CTs retained longer; regulated verticals must satisfy applicable statutory retention minima).

Assessment / Verification Objectives

1. Verify that every action producing `trigger_matched == true` has a corresponding decision-trace record.
2. Verify that `input.trace.append_only == true` in every record (`conformance/laas/laas.rego:125-127`).
3. Verify continuity of `actor_chain_prev_hash` across sequential records for each actor.
4. Verify that `merkle_anchor` is present and updated at the declared anchoring interval.
5. Review data-governance controls: confirm PII/MNPI fields are tokenized before writing, confirm index-payload separation is implemented.
6. Confirm retention policy by CT: obtain the retention schedule and verify it meets or exceeds declared minimums per CT.

LAAS-AGG-1

Control ID: LAAS-AGG-1 **Obligation ID:** `LAAS-OBL-AGG-001` **Severity:** error **CT floor:** 0 **Precedence:** 85

Control Statement

The system shall maintain a windowed aggregate of the effect surface across all actions by a given principal or session. When the aggregate effect crosses a CT threshold, the system shall re-tier subsequent actions to the aggregate's CT (the effective CT). The gate-assigned CT shall never be set below the cumulative window CT.

Discussion

Per-action tiering is necessary but not sufficient. N individually sub-threshold actions can compose into a high-CT aggregate effect: a form of structuring analogous to transaction structuring in financial regulation (`docs/laas/proposal-v1.1.md §6.4`). The Rego policy computes `_agg_ct` from `input.aggregate.window_effect_ct` and takes `effective_ct = max(gate_assigned_ct, _agg_ct)` (`conformance/laas/laas.rego:47-49`):

```
effective_ct := max([input.gate.assigned_ct, _agg_ct])
_agg_ct := object.get(input, ["aggregate", "window_effect_ct"], 0)
```

The AGG-001 violation fires when `gate_assigned_ct < _agg_ct` (`conformance/laas/laas.rego:130-135`), preventing an operator from assigning a tier that ignores cumulative exposure.

Assessment / Verification Objectives

1. Verify that the gate maintains a windowed aggregate per principal/session/effect-class.

2. Verify that `effective_ct >= aggregate_window_ct` for every decision-trace record.
3. Construct a test sequence of N low-CT actions whose aggregate crosses CT3; confirm the gate re-tiers the (N+1)th action to CT3 and emits the appropriate obligations.
4. Verify that the AGG-001 violation fires when `assigned_ct < _agg_ct`.
5. Verify that `aggregate_window_ct` is a declared field in the trace (field name `aggregate_window_ct`, `docs/laas/proposal-v1.1.md §7.2`).

LAAS-INP-1

Control ID: LAAS-INP-1 **Obligation ID:** `LAAS-OBL-INP-001` **Severity:** error **CT floor:** 0 **Precedence:** 75

Control Statement

The system shall classify each action's driving input as trusted or untrusted. An action driven by untrusted input (web content, inbound email, third-party data) shall be gated at effective CT ≥ 3 or shall be blocked. The trust classification shall be recorded in the decision trace as `input_trusted`.

Discussion

Untrusted input is a tier-raising signal because it is the primary vector for prompt-injection attacks that attempt to manipulate the agent's tier self-report or verifier selection (`docs/laas/proposal-v1.1.md §8.3`). The floor CT for untrusted-input actions is `conformance/laas/data.json → untrusted_input_min_ct: 3`.

The Rego check (`conformance/laas/laas.rego:138-145`):

```
violations contains obl("LAAS-OBL-INP-001", ...) if {
    input.input.trusted == false
    effective_ct < cfg.untrusted_input_min_ct
    not blocked
}
```

An action is blocked when `input.action_blocked == true`; a blocked action satisfies the obligation regardless of tier.

Assessment / Verification Objectives

1. Verify that the input-classification mechanism labels each action as `trusted` or `untrusted` and that the classification is recorded as `input_trusted` in the trace.
2. Verify that `effective_ct >= 3` whenever `input_trusted == false` and the action is not blocked.
3. Submit a test action driven by web-fetched content; confirm the gate raises the tier to at least CT3 and emits the appropriate obligations.
4. Verify that the INP-001 violation fires when `trusted == false`, `effective_ct < 3`, and `action_blocked != true`.

LAAS-VEN-1

Control ID: LAAS-VEN-1 **Obligation ID:** `LAAS-OBL-VEN-001` **Severity:** error **CT floor:** 0 **Precedence:** 60

Control Statement

When an agent action is executed via a vendor model or third-party tool or API, the system shall record vendor provenance (attribution) and confirm that the vendor's scope is limited to the declared AOE. Residual escape-rate errors attributable to vendor-model outputs shall be charged against the deploying operator's escape-rate budget. Untrusted vendor dependencies shall fail closed.

Discussion

The deploying operator owns the risk of vendor-model errors under the SR 11-7 principle ("you own the risk of models you buy", `docs/laas/proposal-v1.1.md §2`). This control implements that principle as an obligation: if a vendor model produces a wrong output that passes through the gate, the miss counts against the operator's Bucket-B escape rate.

The `vendor_ok` helper (`conformance/laas/laas.rego:68-71`):

```
vendor_ok if {
    input.vendor.attribution != null
    input.vendor.scope_limited == true
}
```

The VEN-001 violation fires when `vendor.used == true` and `vendor_ok` is false (`conformance/laas/laas.rego:148-150`).

Assessment / Verification Objectives

1. Verify that every action using a vendor model or third-party API records `vendor.attribution` (non-null) and `vendor.scope_limited == true` in the trace.
2. Verify that the escape-rate measurement methodology attributes vendor-model misses to the deploying operator's Bucket-B budget.
3. Confirm that vendor dependencies are configured to fail closed: a vendor API error or unavailability produces `action_blocked == true` rather than a degraded pass.
4. Verify that the VEN-001 violation fires when `vendor.used == true` and either `attribution` is null or `scope_limited` is false.

LAAS-IRR-1

Control ID: LAAS-IRR-1 **Obligation ID:** `LAAS-OBL-IRR-001` **Severity:** error **CT floor:** 3 **Precedence:** 80

Control Statement

For every action with effective CT ≥ 3, the system shall require a passing, independent, qualified pre-commit verifier before the action is committed. If the verifier returns `fail`, `abstain`, or `indeterminate`, the action shall be blocked and escalated. At CT4, human approval (LAAS-HUM-1) shall also be required.

The action, derived tier, verifier identity, verdict, independence basis, and qualification reference shall all be recorded in the decision trace.

Discussion

Pre-commit verification is the LAAS analogue of DO-178C independence objectives and SR 11-7 effective challenge: instructions do not prevent execution; an out-of-process mechanical gate does (`docs/laas/proposal-v1.1.md §7.1`).

The independence floor CT is `conformance/laas/data.json → independent_verification_floor_ct: 3`. Scalable CT4 oversight: operators may register **standing/pre-authorized envelopes** so that human approval applies to out-of-envelope actions rather than every individual action; **batched approval** of queued in-envelope items is permitted (`docs/laas/proposal-v1.1.md §6.3`).

The Rego check (`conformance/laas/laas.rego:154-158`):

```
violations contains obl("LAAS-OBL-IRR-001", ...) if {
    requires_independent_verification
    not blocked
    not verifier_passed
}
```

Where `requires_independent_verification` holds when `effective_ct >= 3` (`conformance/laas/laas.rego:62`).

At CT2, an **unrehearsed rollback** does not satisfy the obligation; rollback rehearsal evidence is required wherever rollback substitutes for pre-commit verification (`docs/laas/proposal-v1.1.md §6.3`).

Assessment / Verification Objectives

1. Verify that every decision-trace record with `effective_ct >= 3` and `action_blocked == false` has `verifier.verdict == "pass"`.
2. Verify that `trigger_matched == true` implies either `verdict == "pass"` (with independent, qualified verifier, human approval at CT4, and residual ≤ tolerance) or `action_blocked == true`: the LAAS conformance predicate (`docs/laas/proposal-v1.1.md §7.2`).
3. Submit a test CT3 action without a verifier; confirm the gate blocks it and emits `LAAS-OBL-IRR-001`.
4. Submit a CT3 action with verifier verdict `indeterminate`; confirm the gate blocks and escalates.
5. For deployments using rollback at CT2, obtain rollback-rehearsal evidence records and confirm they are current (not stale beyond the declared rehearsal interval).

LAAS-IND-1

Control ID: LAAS-IND-1 **Obligation ID:** `LAAS-OBL-IND-001` **Severity:** error **CT floor:** 3 **Precedence:** 80

Control Statement

For actions with effective CT ≥ 3, the pre-commit verifier shall be independent of the actor. Independence is satisfied by one of the following, applied in order of tier:

1. **Deterministic / exact checker:** a different kind of checker (e.g., schema validator, ledger reconciliation, invertible round-trip) is independent at any CT for Bucket A.
2. **Distinct model lineage with low error-correlation:** a verifier from a distinct model lineage whose measured error-correlation with the actor is ≤ 0.2 on the evaluation set is independent up to CT3.
3. **Human:** required in addition at CT4.

A verifier sharing the actor's model lineage is presumed non-independent. Where error-correlation cannot be bounded, CT ≥ 3 falls back to deterministic or human.

Discussion

A different model that fails the same way as the actor is not a check (`docs/laas/proposal-v1.1.md §7.3`). The error-correlation threshold is `conformance/laas/data.json → max_error_correlation: 0.2`.

The `independence_ok` rules (`conformance/laas/laas.rego:75-83`):

```
independence_ok if input.verifier.type == "deterministic"
independence_ok if input.verifier.type == "human"
independence_ok if {
    input.verifier.type == "model"
    input.verifier.model_lineage != input.action.actor_model_lineage
    input.verifier.error_correlation <= cfg.max_error_correlation
}
```

Assessment / Verification Objectives

1. Verify that every decision-trace record with `effective_ct >= 3`, `action_blocked == false`, and `verifier.verdict == "pass"` has `verifier.independent == true` and a valid `independence_basis` recorded.
2. For model verifiers: verify that `verifier.model_lineage != actor_model_lineage` and that `verifier.error_correlation <= 0.2` are recorded in the trace.
3. For model verifiers: obtain the evaluation-set correlation measurement report and confirm it is current (measured after the last model/prompt/policy change).
4. Verify the IND-001 violation fires when a passing verifier has the same lineage as the actor or has `error_correlation > 0.2`.
5. Confirm that at CT4, a human verifier is present in addition to any automated verifier.

LAAS-VQ-1

Control ID: LAAS-VQ-1 **Obligation ID:** `LAAS-OBL-VQ-001` **Severity:** error **CT floor:** 3 **Precedence:** 80

Control Statement

A verifier gating actions at effective CT ≥ 3 shall be qualified. Qualification requires: (a) documented coverage of the verifier's claim class; (b) a negative-test suite of known-bad inputs the verifier must catch; and (c) a change-controlled version recorded in the decision trace as `verifier_qualification_ref`. Qualification depth shall scale with CT. An unqualified verifier does not satisfy LAAS-IRR-1.

Discussion

This control is the LAAS analogue of DO-330 (Tool Qualification) (`docs/laas/proposal-v1.1.md §7.5`). Just as DO-178C requires that tools used in software verification are themselves qualified, LAAS requires that verifiers used to gate high-CT actions are qualified.

The Rego check (`conformance/laas/laas.rego:169-174`):

```
violations contains obl("LAAS-OBL-VQ-001", "verifier is not qualified") if {
    requires_independent_verification
    not blocked
    verifier_passed
    not input.verifier.qualified
}
```

Qualification is version-controlled: a verifier updated after its last qualification cycle is treated as unqualified until re-qualified.

Assessment / Verification Objectives

1. Verify that every decision-trace record with `effective_ct >= 3`, `action_blocked == false`, and `verifier.verdict == "pass"` has `verifier.qualified == true` and a non-null `verifier_qualification_ref`.
2. Obtain the qualification dossier referenced by `verifier_qualification_ref`; confirm it includes claim-class coverage documentation and a negative-test suite.
3. Confirm that the verifier version recorded in the trace matches the current deployed version.
4. Verify that the VQ-001 violation fires when `verifier.qualified == false`.
5. Confirm that any verifier update triggers re-qualification before the verifier is used to gate CT ≥ 3 actions.

LAAS-RES-1

Control ID: LAAS-RES-1 **Obligation ID:** `LAAS-OBL-RES-001` **Severity:** error **CT floor:** 2 **Precedence:** 70

Control Statement

For Bucket-B (open-world) actions at effective CT ≥ 2, the operator shall declare a maximum acceptable residual escape rate per CT, shall estimate it by backtesting on a held-out adversarially-stressed set with a stated confidence interval, and shall re-measure on any model, prompt, tool, or policy change. Conformance requires measured escape rate ≤ declared tolerance at every CT, with traceable evidence referenced in the decision trace.

Declared tolerances (source: `conformance/laas/data.json → escape_rate_tolerance_by_ct`):

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

Discussion

The escape rate is the rate at which a wrong output passes every applicable check and is acted upon. Conformance asserts `measured_escape_rate ≤ tolerance`, not correctness (`standard/LAAS.md §4.3`, `docs/laas/proposal-v1.1.md §5`). The Rego check fires when `input.residual_error_bound > residual_tolerance` (`conformance/laas/laas.rego:184-192`); for pure Bucket-A actions (`residual_error_bound == null`), the comparison is undefined and the violation does not fire.

If a higher-is-better figure is needed, `integrity = 1 − escape_rate`: at CT3, `integrity ≥ 0.995`; at CT4, `integrity = 1.0` (Bucket B contributes zero undetected escapes; all CT4 actions are either deterministically verified or blocked).

Backtesting methodology requirements (`docs/laas/proposal-v1.1.md §5`): the held-out set must be representative of the operational action distribution and adversarially stressed; confidence intervals must be stated; the set must be withheld from model training and fine-tuning; the evaluation set is independently audited (`docs/laas/proposal-v1.1.md §10.1`).

Assessment / Verification Objectives

1. Obtain the operator's declared escape-rate tolerances; confirm they meet or are below the LAAS minimums (CT2 ≤ 0.02, CT3 ≤ 0.005, CT4 = 0.0).
2. Obtain the most recent backtest report; confirm it was produced after the last model/prompt/tool/policy change.
3. Verify that `residual_error_bound ≤ residual_tolerance` in every decision-trace record where `residual_error_bound` is non-null.
4. Confirm the evaluation set was adversarially stressed and independently audited.
5. Verify that the RES-001 violation fires when `residual_error_bound > residual_tolerance`.
6. Confirm re-measurement is triggered (and evidence is refreshed in the trace) upon any model, prompt, tool, or policy change.

LAAS-HUM-1

Control ID: LAAS-HUM-1 **Obligation ID:** `LAAS-OBL-HUM-001` **Severity:** error **CT floor:** 4 **Precedence:** 95

Control Statement

For every action with effective CT = 4, the system shall obtain human approval before committing the action, unless the action is blocked. Human approval shall be recorded in the decision trace as `human_approval.approved == true` and `escalation_approved == true`. Abstention is the default when no approval is received within the declared timeout.

Discussion

Human approval at CT4 implements the most restrictive control in the LAAS hierarchy: the two-person-rule analogue for irreversible, high-consequence agent actions (`standard/LAAS.md §3`, `docs/laas/proposal-v1.1.md §6.3`).

Human approval need not be per-action. Operators may register **standing/pre-authorized envelopes**: a human pre-approves a bounded action class with explicit limits (e.g., external transfers ≤ $X to allowlisted counterparties); the gate enforces the bounds and escalates only out-of-envelope actions. **Batched approval** of queued in-envelope items is also permitted. Human attention thereby scales with exceptions, not volume (`docs/laas/proposal-v1.1.md §6.3`).

The human-approval floor CT is `conformance/laas/data.json → human_approval_floor_ct: 4`. The Rego check (`conformance/laas/laas.rego:177-181`):

```
violations contains obl("LAAS-OBL-HUM-001", ...) if {
    effective_ct >= cfg.human_approval_floor_ct
    not blocked
    not human_approved
}
```

Assessment / Verification Objectives

1. Verify that every decision-trace record with `effective_ct == 4` and `action_blocked == false` has `human_approval.approved == true` and `escalation_approved == true`.
2. For deployments using standing envelopes: obtain the envelope definition and confirm it specifies explicit limits (amounts, counterparty allowlists, action classes).
3. Confirm that out-of-envelope actions produce escalation and are not committed without explicit human approval.
4. Confirm that the timeout behavior defaults to `verdict: abstain` when no approval is received.
5. Verify that the HUM-001 violation fires when `effective_ct >= 4`, `action_blocked != true`, and `human_approval.approved != true`.

---

5. Consequence-Tier Model and Escape-Rate Measurement

5.1 Tier Derivation

The gate derives the Consequence Tier using the lattice defined in `conformance/laas/data.json → tier_lattice`. The derivation function (`docs/laas/proposal-v1.1.md §6.1`, `conformance/laas/laas.rego:32-44`):

```
# Read-only or fully sandboxed: CT0
if not external_effect:
    ct = 0

# All three axes known: take the max
elif reversibility and scope and consequence are all known:
    ct = max(rev_ct[reversibility], scope_ct[scope], cons_ct[consequence])

# Any axis unknown: default to CT4
else:
    ct = 4
```

The three axes and their ranks:

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

Worked example (`docs/laas/proposal-v1.1.md §6.1`): An agent calls `payments.transfer(amount=250000, dest=external)`. Observed surface: reversibility=`irreversible` (rank 4), scope=`public` (rank 4), consequence=`high` (rank 4). `ct = max(4, 4, 4) = CT4`. The agent's self-reported tier is irrelevant; the gate's derivation stands (LAAS-TIER-1).

5.2 Effective CT and Anti-Structuring

The effective CT is `max(gate_assigned_ct, aggregate_window_ct)` (`conformance/laas/laas.rego:47`). The cumulative window prevents structuring: N individually low-CT actions that compose into a high-CT aggregate trigger re-tiering of subsequent actions (LAAS-AGG-1).

5.3 Escape-Rate Measurement Method

Bucket A (deterministically checkable). An exact oracle exists (compile, schema-validate, ledger reconciliation, hash match, invertible round-trip). Run the exact verifier; escape rate approaches zero, bounded by verifier soundness (itself an obligation, LAAS-VQ-1). Deterministic verifiers return `pass`, `fail`, or `indeterminate` (when required inputs are missing); they never return `abstain`.

Bucket B (open-world). No exact oracle. The measurement protocol (`docs/laas/proposal-v1.1.md §5`):

1. **Bound:** the operator declares a maximum escape rate per CT (tolerances in §5.1 above).
2. **Measure:** estimate by backtesting on a held-out, representative, adversarially-stressed set with a stated confidence interval.
3. **Control:** route to independent verification or block when above tolerance; abstain/ escalate when below confidence or out of envelope.
4. **Re-measure:** after any model, prompt, tool, or policy change.
5. **Audit:** the evaluation set is independently audited to prevent gaming.

The escape rate is the governing metric for Bucket B. Conformance asserts `measured_escape_rate ≤ tolerance` with traceable evidence. LAAS is a standard of care, not a correctness guarantee (`standard/LAAS.md §1`).

5.4 Obligation Lifecycle and Conflict Resolution

Obligations move through the lifecycle `propose → active → deprecated → retired` (`docs/laas/proposal-v1.1.md §9.6`). Retired obligations leave the active set. When two obligations fire with incompatible outcomes, precedence is deterministic: the highest CT wins; ties resolve to the most restrictive outcome (block). Each obligation carries a `precedence` field in `conformance/laas/data.json`; the gate's resolution order is recorded in the trace.

---

6. Implementation and Assurance Tiers

6.1 Relationship to RMF Organizational Tiers

The NIST AI RMF defines four organizational Tiers of AI risk management maturity (Tier 1 Partial through Tier 4 Adaptive). LAAS controls apply at the technical-implementation level regardless of organizational RMF maturity, but the depth of implementation naturally aligns with organizational maturity.

RMF Org Tier	Expected LAAS Implementation Depth
Tier 1 (Partial)	CT0–CT2 controls operational; CT3–CT4 controls planned
Tier 2 (Risk Informed)	All 12 controls operational; backtest evidence available
Tier 3 (Repeatable)	Qualified verifiers; documented AOE; signed bundles; attestation bundles produced
Tier 4 (Adaptive)	Continuous escape-rate measurement; standing envelopes; automated re-qualification trigger on change

6.2 Phased Adoption

Operators new to LAAS may adopt controls in the following sequence:

1. **Phase 1, Trace and enforcement plane (LAAS-ENF-1, LAAS-TRC-1):** stand up the append-only trace and the out-of-process gate before adding obligation checks.
2. **Phase 2, Tier derivation (LAAS-TIER-1, LAAS-SELF-1, LAAS-AGG-1):** implement the tier-lattice derivation and cumulative-window aggregation.
3. **Phase 3, Input and vendor controls (LAAS-INP-1, LAAS-VEN-1):** add input-trust classification and vendor attribution.
4. **Phase 4, Independent verification (LAAS-IRR-1, LAAS-IND-1, LAAS-VQ-1):** qualify verifiers and wire them to the CT≥3 gate.
5. **Phase 5, Escape rate and human approval (LAAS-RES-1, LAAS-HUM-1):** run initial backtests, declare tolerances, and implement the CT4 human-approval path.

Appendix A: Control-to-RMF-Subcategory Mapping

This table is a compact reference. The full discussion is in Section 3.

Control ID	Obligation ID	RMF Function	RMF Subcategory
LAAS-TIER-1	LAAS-OBL-TIER-001	MAP	MP-2.3
LAAS-SELF-1	LAAS-OBL-SELF-001	MAP	MP-2.3
LAAS-ENF-1	LAAS-OBL-ENF-001	GOVERN	GV-1.1
LAAS-TRC-1	LAAS-OBL-TRC-001	MEASURE	MS-2.6
LAAS-AGG-1	LAAS-OBL-AGG-001	MAP	MP-3.5
LAAS-INP-1	LAAS-OBL-INP-001	MAP	MP-2.5
LAAS-VEN-1	LAAS-OBL-VEN-001	GOVERN	GV-6.1
LAAS-IRR-1	LAAS-OBL-IRR-001	MANAGE	MG-2.2
LAAS-IND-1	LAAS-OBL-IND-001	MEASURE	MS-2.8
LAAS-VQ-1	LAAS-OBL-VQ-001	MEASURE	MS-4.1
LAAS-RES-1	LAAS-OBL-RES-001	MEASURE	MS-2.5
LAAS-HUM-1	LAAS-OBL-HUM-001	MANAGE / GOVERN	MG-2.4 / GV-1.4

Appendix B: Mapping to Source Standards

LAAS inherits its structural mechanics from three mature assurance standards (`docs/laas/proposal-v1.1.md §2, §3`). This appendix maps LAAS mechanisms to their source-standard analogues.

LAAS Mechanism	DO-178C analogue	SR 11-7 analogue	UL 4600 analogue
CT0–CT4 (consequence tiering)	DAL A–E (design assurance level)	Risk-tiering by materiality	Risk-based safety case depth
Escape rate (Bucket B)	Structural coverage (MC/DC)	Backtesting on held-out set	Safety Performance Indicators (SPIs)
Independent pre-commit verifier (IRR-001)	Independence objectives ("with independence")	Effective challenge	Independent assessment
Verifier qualification (VQ-001)	DO-330 tool qualification	Validator qualification	Safety case evidence quality
Decision trace (TRC-001)	Bidirectional trace + data package	Validation documentation	GSN + evidence
AOE + abstention	Configuration / operational envelope	Approved use scope	Operational Design Domain (ODD)
Bucket B open-world bounding	None (weak analogue only)	Conceptual soundness	SOTIF (ISO 21448)
HUM-001 (CT4 human approval)	None	Human sign-off for high-impact	None

DO-178C direction note: DO-178C DAL letters fall A→E with decreasing consequence (A is most severe); LAAS CT numbers rise with consequence (CT4 is most severe). These are inverses; cross-mapping requires explicit reversal. (`docs/laas/proposal-v1.1.md §6.1`).

SR 11-7 (Federal Reserve Supervisory Guidance on Model Risk Management, 2011, extended by SR 26-2 for AI/ML) provides the primary regulatory-pull pathway for LAAS adoption in financial-services contexts. LAAS escape-rate measurement maps directly to SR 11-7 ongoing backtesting and outcome analysis; the VEN-001 control maps to SR 11-7's vendor-risk ownership principle.

UL 4600 (Standard for Safety for the Evaluation of Autonomous Products) provides the safety-case and standard-of-care culture for Bucket-B bounding. LAAS is a standard of care in the same sense as UL 4600: conformance asserts that the right checks ran by the right party with evidence, not that no error can occur (`standard/LAAS.md §1`).

Appendix C: References

Normative References

- `standard/LAAS.md`: LLM-Agent Assurance Standard v1.1 (normative prose)
- `conformance/laas/data.json`: LAAS bundle `laas-fin-1.1.0` (machine source of truth for thresholds and obligation registry)
- `conformance/laas/laas.rego`: OPA policy, package `kellerai.laas.actions` (enforcing policy)

Informative References

- `docs/laas/proposal-v1.1.md`: LAAS Design Rationale and Proposal v1.1 (design record)
- NIST AI 100-1 (2023): Artificial Intelligence Risk Management Framework (AI RMF 1.0). National Institute of Standards and Technology.
- NIST AI 600-1 (Jul 2024): Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile.
- RTCA DO-178C (2011): Software Considerations in Airborne Systems and Equipment Certification. RTCA, Inc.
- RTCA DO-330 (2011): Software Tool Qualification Considerations. RTCA, Inc.
- Board of Governors of the Federal Reserve System (2011): SR 11-7: Guidance on Model Risk Management.
- UL 4600 (2020, Edition 1): Standard for Safety for the Evaluation of Autonomous Products. UL Solutions.
- ISO 21448:2022: Road vehicles. Safety of the intended functionality (SOTIF).
- ISO/IEC 42001:2023: Information technology. Artificial intelligence. Management system.
- Five Eyes (Apr–May 2026): *Careful Adoption of Agentic AI Services* (joint advisory, six agencies, 29 pp).
- CSA MAESTRO threat model, CSA ATF Zero-Trust governance framework, CSA STAR-for-AI (Phase 1, in progress as of mid-2026).
- OWASP Agentic Top 10; OWASP AIVSS.