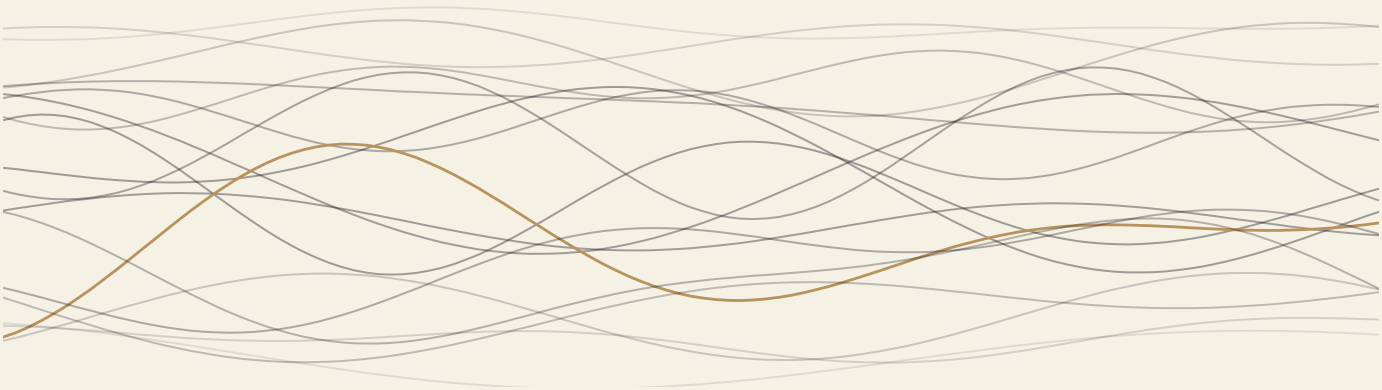
OVERT

OBSERVABLE VERIFICATION EVIDENCE FOR RUNTIME TRUST



An Open Standard for Runtime Trust in AI Systems

DATE	June 2026
PUBLISHED BY	GLACIS Technologies, Inc.
CONTACT	overt-review@glacis.io
STANDARD URL	overt.is
IPR POLICY	overt.is/ipr-policy

OVERT defines how runtime controls, boundary-enforcement decisions, measurement outputs, and response actions are bound to observable verification evidence that independent parties can validate without requiring protected-content egress. It serves both as infrastructure for verifiable AI governance and as a runtime trust layer for enforcement, detection, containment, and post-incident reconstruction.

Informative Foreword: The Verification and Security Gap

Existing AI governance frameworks specify what controls should exist. They generally do not specify how to produce independent, tamper-evident proof that those controls executed on a given interaction, under a given configuration, at a given time. That gap is not only a governance verification gap. It is also an AI security visibility and containment gap.

As AI systems move into consequential and adversarial settings, operators and relying parties increasingly need more than policy documents, self-generated logs, and periodic audit narratives. They need trustworthy evidence of what executed at the runtime boundary; what traffic was within mediation scope; what was allowed, denied, sampled, escalated, or overridden; whether the enforcing component was the expected one; and whether those records can be verified independently after an incident without creating a new protected-content disclosure channel.

Current practice often lacks five properties that mature security operations require:

- **Trusted execution evidence** showing which enforcing component and configuration were active when a governed action occurred.
- **Reliable runtime coverage accounting** showing what traffic and action classes were in scope, what was excluded, and how denominators were derived.
- **Tamper-evident telemetry** that is not reducible to operator-controlled logs.
- **Independent verification of enforcement events** including permit, deny, override, escalation, and response actions.
- **Post-incident reconstruction without routine content disclosure** so relying parties can verify event history without turning attestation into a new protected-data egress channel.

OVERT addresses that narrower problem. It does not adjudicate the merits of any particular governance, security, or legal dispute. It specifies how to produce independently assessable records of control execution, measurement, and response without requiring protected-content egress.

The case for the standard is, in the end, simple: as AI takes on consequential decisions, the people relying on it deserve more than assurances that controls exist. They deserve proof that the controls ran. This document is an attempt to make that proof routine, verifiable, and safe to produce. It is published in the open, under a royalty-free covenant, in the hope that it becomes a shared foundation rather than any one company's advantage.

Joe Braidwood
Co-Founder & CEO, GLACIS Technologies, Inc.
GLACIS Technologies, Inc.

What changed in 1.1

- Regulatory dates refreshed (Colorado, EU).
- All framework and regulatory crosswalks moved to the informative companion `OVERT_v1.1_CROSSWALKS.md`; the normative Attestation Boundary Declaration requirements remain in the standard, renumbered 29.4 → 22.10.
- Scanner and local classifier defined as supporting components.
- Governance language calibrated; non-normative post-quantum note added.
- No unregistered example Protocol Profile names are included in this version.
- New normative Annex G (Supplementary Requirements): Local CAS evidence retrieval and retention integrity (G.1), the HTTP transport binding for cross-boundary attestation (G.2), the automated auditor discovery / well-known endpoint protocol (G.3), and an informative reference schema (G.4) for the ControlAction artifact already mandated by Section 10.
- A versioning and errata policy is now stated in Section 22.11.
- The Part 1–22 conformance levels and existing obligations are otherwise unchanged from 1.0.

Keywords

The key words "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC 2119] [RFC 8174] when, and only when, they appear in all capitals, as shown here.

Requirements marked SHALL are normative and required for conformance. Requirements marked SHOULD or RECOMMENDED are normative recommendations. Requirements marked MAY indicate permitted behavior that is truly optional for conformance. Annexes A through F are informative. Annex G (Supplementary Requirements) is normative, except Section G.4, which is an informative reference. All notes, examples, architectural references, and case studies are informative.

Reading OVERT

Informative — an orientation, not a requirement.

Governance has always been able to say what ought to be done. It has rarely been able to prove what was. Policies, audit narratives, and self-reported logs record intentions and recollections; they

are not evidence. While the systems under governance were slow and human, the gap was tolerable. Systems that now act at machine speed, in consequential settings, make it tolerable no longer.

OVERT closes that gap. It specifies how a runtime control produces, as a by-product of doing its work, a signed record that an outside party can verify — without the protected data ever leaving the operator's environment. The result is not a better log. It is evidence: tamper-evident, independently checkable, and silent about everything it need not disclose.

Four commitments hold the standard together.

- **Evidence, not assertion.** A governed action yields a receipt a third party can check — not a claim it is asked to trust.
- **Containment by construction.** Only cryptographic fingerprints and signatures cross the boundary. The content stays home.
- **Independence by structure.** Whoever attests is separate from whoever is governed. Self-attestation is not independent attestation.
- **Measurement, not adjective.** Safety is stated in intervals and sample sizes an auditor can reproduce, not in words.

The standard is published as one volume and as six standalone fascicles. Read Part 1 (Foundations) for what is being proved and at what standard of proof. Read Part 2 (Governance Domains) and the Crosswalks companion to see OVERT mapped onto the frameworks you already answer to. Read Part 3 (Agentic AI Controls) to govern systems that act. Read Part 5 (Conformance) to learn what a claim must prove and who may verify it.

A word on how to read what follows. The standard's obligations are the numbered statements that use SHALL, SHOULD, and MAY; those, and only those, define conformance. The italic asides set against the margin rule — marked Plainly where they translate a dense idea — carry no requirements. They exist so the argument can be followed without the proof, and the proof consulted only where it is needed.

Table of Contents

Intellectual Property Rights Notice

Patent Covenant. GLACIS Technologies irrevocably covenants not to assert any patent claim it owns or controls that is necessarily infringed by implementing this standard, against any person or entity making, using, selling, offering to sell, importing, or distributing an implementation of this standard. This covenant applies to any such implementation whether or not it claims or achieves conformance, whether the implementation is commercial or non-commercial, whether complete or partial, and whether the attestation capabilities defined here are combined into a unified pipeline or implemented separately. The covenant is irrevocable, runs with the patents, and is binding on GLACIS Technologies and its successors and assigns.

The sole condition is defensive: this covenant terminates as to any person or entity that asserts, or voluntarily participates in asserting, a patent claim against GLACIS Technologies, its successors, or its assigns. Such termination affects only the asserting party; the covenant remains in full force as to every other person and entity.

Patent Disclosures. GLACIS Technologies holds patent filings related to certain methods described in this standard. Full patent disclosures, claim-scope details, and guidance for alternative implementations are published at overt.is/ipr-policy.

Standard Normative Requirements. The normative requirements of this standard — including the attestation envelope structure, conformance levels, auditor verification procedures, and agentic governance controls — define functional properties and interoperability requirements. They do not mandate the use of any specific patented method. Conformance with this standard can be achieved through multiple architectural approaches, and implementers are free to select cryptographic constructions and architectural patterns that satisfy the normative requirements.

Contributor Disclosure. Contributors to this standard who are aware of potentially essential patent claims are expected to disclose them. Protocol Profile registration, conformance criteria, and certification are governed by published, rule-based process rather than by discretion. The requirements for each conformance level, the registration process, the self-declaration fallback, and the mandatory conformance test suites are specified in the OVERT standard (Section 22.6), and self-declared profiles are barred from Level 3 and Level 4 conformance claims. Glacis Technologies is the current steward and maintainer. OVERT is on a defined path to multi-stakeholder stewardship: Glacis is establishing co-stewardship with domain partners and intends to formalize an independent, multi-party governance body that no single entity, Glacis included, can control. Until that body is constituted, governance follows the published process above together with the irrevocable royalty-free Patent Covenant in this Notice, both of which bind the steward. Multiple profiles are permitted.

Conformance requires exactly one declared Protocol Profile per deployment; registration requirements are level-dependent (Section 22.6), and Levels 3 and 4 require a registered profile.

PART 1: FOUNDATIONS

> *Everything that follows depends on what is settled here: what is being proved, to whom, and at what standard of proof. The vocabulary, the four-rung assurance ladder, and the trust model are established in this part.*

1. Purpose and Scope

1.1 Purpose

OVERT defines an open standard and certification framework for attested AI runtime control systems. It specifies requirements for generating, storing, preserving, and verifying cryptographic proof that declared governance and runtime control decisions executed under a defined configuration, within a bounded time interval, without requiring protected-content egress.

In this role, OVERT serves three related purposes. First, it supports verifiable AI governance by making policy execution, oversight actions, measurement outputs, and response activities independently assessable. Second, it provides a low-level control-and-evidence substrate for AI runtime security by enabling attested runtime identity, policy-mediated execution decisions, evidence that declared tool and boundary controls executed within the attested scope, tamper-evident telemetry, attested response actions, and post-incident reconstruction of control execution history. Third, it provides the conformance, independence, and assessment model by which relying parties can distinguish self-asserted deployment claims from independently assessed, evidence-grade runtime mediation.

OVERT is not itself a universal runtime-control product. Enforcement is performed by conformant arbiters, sidecars, gateways, proxies, or equivalent runtime-control implementations operating under a registered Protocol Profile. OVERT defines what those implementations SHALL prove, what evidence an independent attestation provider SHALL verify, and what a qualified assessor SHALL examine when a conformance claim is made.

OVERT does not replace governance frameworks, security engineering disciplines, runtime-control implementations, or legal analysis. Organizations remain responsible for defining policies, selecting controls, securing infrastructure, evaluating models, and satisfying applicable law. OVERT specifies how to produce temporally bound, tamper-evident, independently verifiable artifacts demonstrating that declared controls executed and that attested measurements and response actions can be reconstructed and checked by relying parties.

OVERT attests control execution and associated evidence quality. It does not attest the truthfulness of model outputs, the absence of hallucination, the absence of compromise, or the adequacy of the operator's policies. Attestation artifacts are designed to support authenticity, integrity, timing, auditability, and chain of custody. Their legal relevance, admissibility, and sufficiency remain questions of applicable law and context.

1.2 Limitations of Attestation

OVERT does not:

- Replace endpoint, cloud, network, application, platform, model, or software supply-chain security controls.
- Detect every attack, abuse path, or failure mode by itself.
- Guarantee that declared policies are adequate, lawful, or well configured.
- Make an unsafe, insecure, or poorly governed AI system safe merely because attestations are produced.
- Attest the quality, accuracy, truthfulness, fairness, robustness, or cybersecurity of model outputs as substantive properties.
- Eliminate the need for human incident response, forensic investigation, sector-specific controls, or domain-specific validation.
- Guarantee legal compliance, regulatory approval, evidentiary admissibility, or insurance coverage.
- Prove the absence of compromise, data poisoning, prompt-injection success, or unauthorized access outside the attested scope.
- Substitute attestation artifacts, managed deployment claims, or certification language for actual in-path runtime mediation.
- Treat operator or vendor assertions about deployment completeness as sufficient for AAL-4 or Level 4 conformance absent the independence requirements of this standard.

OVERT proves, within the claimed scope and assurance level, that certain controls, measurements, and response actions were executed or recorded in the manner specified by this standard. Whether those controls were sufficient for a given use case remains a separate question.

Training-time operations (data preparation, model training, experiment tracking, fine-tuning), data lifecycle management (versioning, freshness, deletion), and platform infrastructure security (vulnerability management, SDLC, patching, secrets management) are outside the OVERT attestation scope. These are important controls addressed by frameworks including DASF, NIST SP 800-53, and ISO 27001. OVERT complements but does not replace them.

A future OVERT Build Assurance Profile may define attestation requirements for training, data, and platform lifecycle controls. Until such a profile is published, conformance statements SHALL NOT state or imply that OVERT conformance covers training-time operations, data lifecycle management, or platform infrastructure security; a statement that does so is non-conformant.

1.3 Scope

This standard applies to:

- **AI system operators** deploying AI in regulated industries (healthcare, financial services, insurance, employment, education, housing)
- **AI system developers** building products subject to governance obligations
- **Security teams, incident responders, auditors, procurement reviewers, and regulators** who need to verify control execution without routine access to protected content
- **Insurers** who need quantitative, cryptographically verifiable data to price AI risk
- **Agentic AI systems** where autonomous agents execute tool calls, access external resources, and make decisions without step-by-step human oversight

1.4 Relationship to Existing Standards

OVERT operates beneath and alongside existing AI governance, security, attestation, and certification frameworks. Its role is to provide a trust, execution-control, telemetry, evidence, and conformity-assessment substrate for AI systems: a mechanism by which governance and security-relevant events can be bound to runtime state, recorded without protected-content egress, and independently verified.

OVERT therefore complements, but does not replace, governance frameworks such as NIST AI RMF and ISO/IEC 42001; security frameworks such as NIST SP 800-53, FedRAMP, and zero-trust architectures; attestation architectures such as IETF RATS; and implementation products that actually mediate runtime actions. Those frameworks and products specify objectives, controls, management processes, trust relationships, or execution mechanisms at broader organizational and system levels. OVERT specifies how to generate and verify cryptographic records of declared control execution for AI systems and agentic workflows, and how those claims are assessed for conformance.

Conformance with OVERT is not a determination of compliance with any other standard, law, or regulatory regime. Rather, OVERT artifacts may support evidence for requirements defined elsewhere, subject to the scope, assurance level, and limitations of this standard.

OVERT operates beneath and is complementary to:

Standard	Role	OVERT Relationship
NIST AI 100-1 (AI RMF 1.0)	Risk management functions	OVERT provides attestation artifacts supporting evidence that GOVERN/MAP/MEASURE/MANAGE activities were executed
ISO/IEC 42001:2023	AI management system	OVERT supports evidence for A.6.2.8 (event logging) and extends event records to tamper-evident, third-party-verifiable attestation
EU AI Act (Regulation 2024/1689)	Regulatory requirements	OVERT supports evidence for Article 12 (automatic logging) and aspects of Article 9 (risk management) documentation requirements. Regulation (EU) 2024/1689 generally applies from 2 August 2026. Article 6(1) and corresponding obligations apply from 2 August 2027. Annex III systems (Article 6(2)) follow the general application date. Note: The Digital Omnibus (published by the Commission on November 19, 2025, with provisional political agreement reached on 6 to 7 May 2026, confirmed by Member State representatives in the Council on 13 May 2026, and expected Official Journal publication before 2 August 2026) defers standalone Annex III high-risk obligations to 2 December 2027, and product-embedded Annex I high-risk obligations (including medical devices) to 2 August 2028. Original dates legally stand until adoption.
IETF RATS (RFC 9334)	Remote attestation architecture	OVERT instantiates the Attester/Verifier/Relying Party model for AI attestation

Standard	Role	OVERT Relationship
NIST OSCAL	Machine-readable compliance	OVERT attestation packs are expressible as OSCAL assessment results
Registered OVERT Protocol Profile	Implementation specification	Specifies cryptographic constructions, envelope schemas, key derivation, and signal formats implementing this standard. Protocol Profile 1.0 is the initial registered profile; see Annex B
NIST SP 800-53 Rev 5	Security and privacy controls	OVERT maps to AU (Audit), SI (System Integrity), IA (Identification/Authentication) families
FedRAMP Moderate Baseline	Federal cloud authorization	OVERT attestation architecture supports evidence for FedRAMP AU and SI control families
NIST AI RMF GenAI Profile	GenAI-specific guidance	OVERT provides attestation artifacts for GenAI-specific GOVERN/MEASURE/MANAGE recommendations
NIST SP 800-207	Zero Trust Architecture	OVERT trust architecture is complementary; "untrusted SUT" model is distinct from ZTA network assumptions
OMB M-25-21 / M-25-22	Federal AI procurement	OVERT attestation packs support AI use case inventory and risk management documentation requirements. M-25-22 applies to solicitations issued on or after September 30, 2025 (180 days after its April 3, 2025 issuance) and to contract options exercised on or after October 1, 2025. Agency AI inventory/reporting obligations are in M-25-21 (reporting on the schedule set by OMB implementation instructions). M-25-22 excludes National Security Systems

> FRAMEWORK CROSSWALKS (INFORMATIVE COMPANION). *Detailed crosswalks to external frameworks and regulatory texts — NIST AI RMF, ISO/IEC 42001, the EU AI Act, AIUC-1/OWASP, NIST SP 800-53 Rev 5/FedRAMP, OMB M-25-21/M-25-22, the Databricks AI Security Framework (DASF) v3.0, the IMDRF N93 draft* Technical Framework for AI Life Cycle Management*, the CHAI Governance Playbooks, the Joint Commission RUAIH guidance, and the Databricks AI Governance Framework (DAGF) — are maintained in the informative companion document* `OVERT_v1.1_CROSSWALKS.md`.

> *The companion is informative and imposes no requirements; it is not required to determine OVERT conformance, and OVERT conformance does not determine compliance with any framework cross-walked there.*

1.5 Design Principles

1. **Attestation by construction, not assertion.** Controls produce cryptographic proof as a byproduct of execution, not as a separate documentation exercise.
2. **Privacy by architecture, not policy.** Protected content never leaves the operator's environment on the attestation path. Only cryptographic commitments cross trust boundaries; authorized disclosure under Section 20.4 is a separate, exceptional channel.
3. **Independence by structure.** The entity attesting to governance is structurally independent of the entity being governed. Self-attestation is not compliant.
4. **Statistical rigor by default.** Safety claims carry confidence intervals, sample sizes, and auditor-reproducible methodologies. Unquantified assertions are not attestation artifacts.
5. **Open by design.** This standard is open for implementation by any party under a royalty-free patent covenant. Open-source reference tooling, including an independent receipt verifier, is published under Apache-2.0.
6. **Security-supporting evidence by observation.** The architecture that produces governance evidence occupies the same inline position, binary identity measurement, behavioral monitoring, and tamper-evident recording paths that security detection requires. Within the attested scope, OVERT produces security-supporting evidence — not a complete security architecture. Whether that evidence is sufficient for a given security objective depends on mediation scope, denominator independence, arbiter isolation, IAP topology, and the operator's broader security posture.

2. Normative References

The following documents are referenced normatively within this standard:

2.1 Normative References

- RFC 2119 / RFC 8174: Key words for use in RFCs to Indicate Requirement Levels (BCP 14)
- NIST AI 100-1: AI Risk Management Framework 1.0 (January 2023)

- ISO/IEC 42001:2023: Information Technology — Artificial Intelligence — Management System
- ISO/IEC 22989:2022: Artificial Intelligence — Concepts and Terminology
- RFC 9334: Remote ATtestation procedureS (RATS) Architecture
- RFC 6962: Certificate Transparency
- RFC 8615: Well-Known Uniform Resource Identifiers (URIs)
- NIST SP 800-207: Zero Trust Architecture
- NIST SP 800-53 Rev 5: Security and Privacy Controls for Information Systems and Organizations
- A registered OVERT Protocol Profile (see Annex B for Protocol Profile 1.0, the initial registered profile)

2.2 Informative References

- RFC 8949: Concise Binary Object Representation (CBOR) — Section 4.2, Deterministic Encoding (used by Protocol Profile 1.0)
- RFC 5869: HMAC-based Extract-and-Expand Key Derivation Function (HKDF) (used by Protocol Profile 1.0)
- RFC 8785: JSON Canonicalization Scheme (JCS) (used by Protocol Profile 1.0)
- NIST SP 800-208: Recommendation for Stateful Hash-Based Signature Schemes
- FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA)
- FIPS 205: Stateless Hash-Based Digital Signature Standard (SLH-DSA)
- OWASP Top 10 for Agentic Applications (December 2025)
- NIST AI RMF Generative AI Profile (July 2024)
- OMB Memorandum M-25-21: Accelerating Federal Use of AI through Innovation, Governance, and Public Trust
- OMB Memorandum M-25-22: Driving Efficient Acquisition of Artificial Intelligence in Government
- RFC 9711: Entity Attestation Token (EAT)
- RFC 9162: Certificate Transparency Version 2.0
- AIUC-1: Artificial Intelligence Underwriting Company — Standard for AI Agent Security, Safety and Reliability (January 2026)
- EU Regulation 2024/1689: AI Act
- Colorado SB 26-189: Automated Decision-Making Technology (signed May 14, 2026, with key obligations beginning January 1, 2027; repeals and replaces SB 24-205; enforcement of prior Act stayed by federal court on April 27, 2026, with the state Attorney General stipulating to the stay, citing the rewrite; June 30, 2026 effective date from SB 25B-004 is superseded)

> NOTE: *Protocol Profiles SHOULD include a documented post-quantum cryptographic transition plan referencing NIST FIPS 204 (ML-DSA) or FIPS 205 (SLH-DSA). The informative references to FIPS 204 and FIPS 205 above are included to facilitate such planning.*

3. Terms and Definitions

For the purposes of this standard, the terms in ISO/IEC 22989:2022 and the following apply:

3.1 attestation: A cryptographically signed statement by an independent notary that a specific governance action occurred, at a specific time, under a specific system configuration.

3.2 arbiter: An enforcement component deployed at the boundary between an AI system and external resources that intercepts, evaluates, and gates actions against defined policy.

3.3 co-epoch binding: The cryptographic linkage of an attestation to the exact binary identity and network isolation state of the system during a bounded time interval (epoch).

> FORWARD EXTENSION POINT (INFORMATIVE). *Co-epoch binding presently covers the arbiter's binary identity and network state — not the model under attestation, which OVERT treats as the untrusted system. As GPU confidential computing (e.g., NVIDIA Confidential Computing, TDX-class runtimes) makes model weights and inference runtime measurable, a future Protocol Profile MAY extend the co-epoch binding to include a measured system-under-test identity (a model-weight or runtime hash) where the platform supports it. The receipt schema reserves this as an additive extension; normative model-identity binding is a candidate for a future minor release and does not affect Protocol Profile 1.0.*

3.4 digest publication ledger (DPL): A per-epoch publication of request commitments enabling third-party verification of sampling completeness.

3.5 epoch: A bounded time interval during which system configuration is attested as stable by the notary network. Duration is configurable; recommended values are specified in the registered Protocol Profile.

3.6 attestation assurance level (AAL): One of four tiers (AAL-1 through AAL-4) describing the cryptographic verifiability and independence of governance attestation artifacts. See Section 4.1.

3.7 non-egress attestation: An attestation generation architecture in which protected content never leaves the operator's environment on the attestation path; only cryptographic commitments cross trust boundaries. (Authorized content disclosure under Section 20.4 and Annex G.1 is a separate exception channel, not attestation egress.)

3.8 provisional receipt: A locally-signed attestation generated synchronously during enforcement, pending asynchronous counter-signature by the notary network.

3.9 receipt: A cryptographic artifact proving that a specific enforcement decision was made, at a specific time, under a specific configuration, and attested by an independent party.

3.10 statistical safety signal: A quantified statement of the form "with [confidence]% confidence, the evaluator-judged violation rate for [policy] did not exceed [bound]% during [epoch]," derived from cryptographically verifiable random sampling. The bound is on the rate of violations as judged by the deployment's version-attested evaluation instrument (Section 16.1, Section 19.2 note); sampling integrity and verdict integrity are distinct verification axes.

3.11 tool call: An action by an AI agent that invokes an external capability — API call, database query, file operation, code execution, communication, or any interaction with systems outside the model's internal computation.

3.12 human-in-the-loop (HITL) interaction: Any event where a human provides consent, approval, review, correction, override, or other governance-relevant input to an AI system workflow. HITL interactions are attestable events subject to the same attestation requirements as automated enforcement decisions.

3.13 notary network: One or more structurally independent nodes that validate attestations on behalf of relying parties. A single structurally independent node satisfies the AAL-4 independence requirement (Section 4.1.1). Where multiple, geographically distributed nodes are deployed, agreement of t-of-n nodes is required before a valid receipt can be issued, providing resilience such that no single node can forge or suppress attestation artifacts; the signature construction achieving the t-of-n property (threshold signature, multi-signature, or other scheme) is specified in the registered Protocol Profile. A multi-node set operated by a single entity (platform-operated, ATT-5.1(a)) remains a single trust entity: its node count provides operational redundancy, not the multi-party resilience property, and its topology disclosure is `Single-IAP` (Section 22.4). The distinction between attestation independence (met by a single independent node) and attestation resilience (provided by multi-node t-of-n sets) is set out in Section 4.1.1 and Annex A (A.34).

3.14 independent attestation provider (IAP): An entity structurally independent of the AI system operator that operates notary infrastructure, validates attestations, and publishes transparency log entries.

3.15 protocol profile: A registered implementation specification defining cryptographic constructions, envelope schemas, key derivation methods, and receipt formats that implement this standard. Multiple profiles may coexist. Conformance requires exactly one declared profile per deployment; registration requirements are level-dependent (Section 22.6): Level 1 may state `No Profile`, Levels 1–2 may use a validly self-declared profile (Section 22.6.3), and Levels 3–4 require a registered profile.

3.16 mediation scope statement: A signed declaration identifying the action types, components, tenants, and traffic paths covered by the attestation system. Published in machine-readable form and referenced in risk signal computation. The mediation scope statement defines what is "in scope" for coverage ratio, exposure window, and other signal denominators.

3.17 qualified risk officer: An individual with documented authority and competence to make risk classification and severity determination decisions under GOV-3. Competence criteria are defined by the operator's risk management policy and SHALL include documented training in AI risk management. The qualified risk officer for an AI system SHALL NOT be the system's sole developer. Referenced in GOV-3.5 as the required policy artifact signer.

3.18 baseline intent declaration: A machine-readable, versioned, hash-chained governance artifact specifying the permitted agent topology, behavioral bounds per agent class, permitted spawn relationships, model bindings, and human oversight checkpoints for a deployment. Published to the transparency log. The baseline intent declaration is the reference artifact against which behavioral drift (3.21) is measured.

3.19 graph complexity metric: A quantitative measure of agentic execution topology — including edge count, branching factor, and depth utilization — computed per execution and evaluated relative to thresholds declared in the baseline intent declaration (3.18).

3.20 causal drift attribution: The process of tracing a detected behavioral drift signal in one agent to a correlated change in an upstream agent via parent-child attestation linkages in the transparency log.

3.21 behavioral drift: A statistically significant change in an agent's output distribution, tool selection distribution, or interaction patterns that occurs within authorized behavioral bounds — distinct from a policy violation. Behavioral drift is detected by sequential statistical methods operating on measurement features produced by the evaluation instrument specified in the registered Protocol Profile.

3.22 scanner: A runtime monitoring sidecar or component that inspects inputs, outputs, and intermediate states of an AI system to detect policy violations, security threats, or behavioral drift.

3.23 local classifier: A local evaluation component that runs classification or inference models to categorize inputs, outputs, or agent behaviors for policy decision-making.

3.24 capability artifact: A notary-signed authorization for a bounded action scope — at minimum the tool or action class, the session, and an expiry — issued ahead of execution and verifiable at the enforcement point without a network dependency. A capability artifact provides independent authorization of a high-risk action in the blocking path without a synchronous notary round-trip; its encoding and verification procedure are specified in the registered Protocol Profile.

4. Architecture Overview

> PLAINLY — *The four assurance levels rank how little an outsider must trust you. At AAL-1 the evidence is your word; at AAL-4 it is mathematics — anyone can verify the record without trusting the operator, the vendor, or the auditor. Each rung removes a reason to take the system's word for it.*

OVERT architecture defines the trust model by which AI governance claims and AI runtime security claims can be made independently assessable. The architecture is designed to answer a bounded set of questions that existing governance documentation and operator-controlled logs answer poorly: what component enforced the decision, what policy state and network state were in effect, what event occurred at the boundary, what was measured or escalated, and whether those records can be verified without trusting the system under test.

The OVERT architecture intentionally separates four roles that are often collapsed in market messaging: the standard defines the normative requirements, runtime-control implementations mediate execution, independent attestation providers verify attestations and operate notary infrastructure, and qualified assessors certify conformance claims at the levels this standard requires. A single commercial offering MAY package more than one operational role, but packaging does not relax the independence requirements stated in this standard.

The architectural relationship to security is positional, not comprehensive. NGAV and EDR shifted endpoint security toward runtime behavior, policy-mediated execution, tamper-evident telemetry, containment, and post-incident reconstruction. OVERT occupies an analogous inline position for AI systems and produces security-supporting evidence within the attested scope: attested runtime identity, policy-mediated tool and boundary-control decisions, tamper-evident telemetry, inter-agent trust controls, capability mediation records, evidence-preserving response, and verification without routine protected-content egress (Design Principle 6). OVERT is not a complete security product. It does not by itself prove that every declared boundary was complete or uncompromised,

does not establish comprehensive defense, and does not guarantee that mediation scope covers all security-relevant traffic. The attestation infrastructure it specifies produces security-supporting evidence within the declared scope; whether that evidence is sufficient for a given security objective depends on scope completeness, denominator independence, arbiter isolation, IAP resilience, and the operator's broader security controls.

4.1 Attestation Assurance Levels (AAL)

OVERT defines four attestation assurance levels. Each level subsumes the requirements of all lower levels. The levels represent increasing degrees of verifiability: AAL-1 and AAL-2 provide documentation and process records suitable for policy declaration and organizational governance; AAL-3 adds machine-generated telemetry and measurement outputs that can be operationally useful for monitoring, but that remain operator-controlled; and AAL-4 adds independently verifiable, cryptographically bound runtime evidence of enforcement, measurement, and response events. Higher AAL tiers produce stronger evidence within the attested scope; they do not by themselves establish comprehensive security.

> NOTE — TERM COLLISION: *"AAL" in this standard means* Attestation Assurance Level*, defined here. It is unrelated to the NIST SP 800-63* Authenticator Assurance Level*, which tops out at AAL3. References elsewhere in this standard to "AAL-4 identity binding" (e.g., RES-3.1) refer to this standard's scale.*

Level	Name	Description	Verification Model
AAL-1	Policy Documentation	Written governance policies exist	Self-asserted; manual review
AAL-2	Process Records	Operational records of governance activities exist	Self-attested; auditor must trust operator
AAL-3	Automated Monitoring	System generates continuous governance telemetry	Machine-generated but operator-controlled
AAL-4	Cryptographic Attestation	Independent third party produces tamper-evident proof of control execution	Third-party verifiable; zero content access required

OVERT conformance requires AAL-4 attestation for all controls designated as AAL-4 in this standard. Controls designated AAL-1, AAL-2, or AAL-3 require the specified level. Conformance is assessed per-control, not globally.

AAL-1 through AAL-3 remain valid for organizational governance activities (policy drafting, training, culture) where cryptographic attestation is not architecturally applicable. For any control that in-

volves runtime AI system behavior — enforcement, monitoring, logging, incident detection — AAL-4 is the target assurance grade and is mandatory at any maturity level whose required architecture supports it (Levels 3 and 4 per Section 4.1.1). Section 22.1 specifies how AAL-4-designated controls are graded at lower maturity levels whose required architecture does not include an independent notary.

4.1.1 DEPLOYMENT ARCHITECTURE AND AAL MAPPING

The following table maps deployment architectures to the maximum attestation assurance level achievable under each architecture. The mapping is normative.

Deployment Architecture	Maximum AAL	Rationale
No attestation infrastructure	AAL-2	Operator-generated records only; no independent verification
Single notary, operator-controlled	AAL-3	Independent attestation present but operator controls the notary
Single notary with hardware-rooted measurement (TEE), operator-controlled	AAL-3	Hardware root of trust strengthens measurement; operator still controls the notary
Multiple notaries (t-of-n), single operating entity	AAL-3	Multi-notary verification present but organizational independence not met
Single notary, independent third party (IAP)	AAL-4	Independent attestation with third-party trust root
Multiple notaries (t-of-n), independent operating entities	AAL-4	Highest assurance; multi-entity independence with third-party verifiability

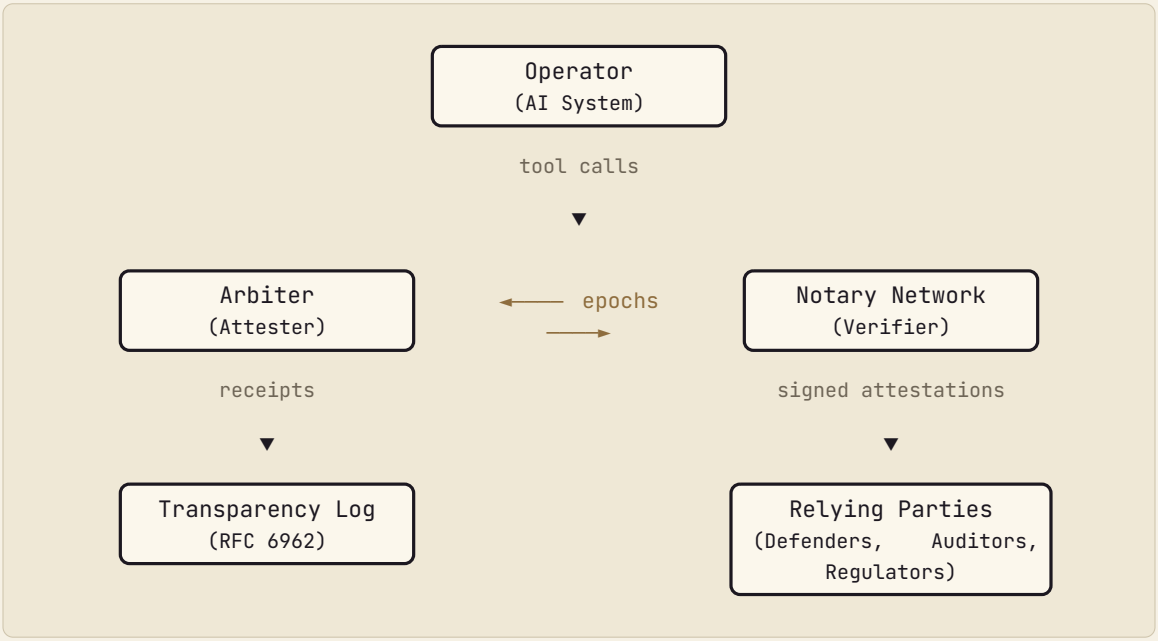
AAL-4 (cryptographic attestation with independent trust root) SHALL require that the notary service be operated by an entity structurally independent of the AI system operator — an Independent Attestation Provider (IAP) per Section 3.14. A single independent notary satisfies AAL-4. Multi-entity notary sets provide additional resilience against compromise but are not required for AAL-4 conformance. Single-IAP AAL-4 therefore establishes attestation independence, not attestation resilience.

Deployments MAY graduate from AAL-3 to AAL-4 by engaging an independent notary service as specified in ATT-5. The transition SHALL be attested in the transparency log with notary set attestations from both the pre-transition and post-transition configurations.

> NOTE: *AAL-1 through AAL-4 describe technical verifiability tiers. They do not correspond to legal burdens of proof, standards of admissibility, or regulatory compliance determinations. Whether an AAL-4 attestation artifact satisfies a particular legal or regulatory standard is a question of applicable law.*

4.2 The Attestation Model

OVERT adopts and extends the IETF RATS (RFC 9334) architecture:



Arbiter (Attester). Deployed at the operator's trust boundary. Intercepts AI system actions, evaluates them against policy, and generates attestation envelopes. The Arbiter sees plaintext — it operates within the operator's security perimeter, analogous to a firewall or security proxy.

Notary Network (Verifier). Structurally independent of the operator. Operated by an Independent Attestation Provider (IAP). Validates attestations using topology-appropriate notary verification — a single independent signer, or t-of-n for multi-notary sets (ATT-3.3) — as specified in the registered Protocol Profile. Derives the Arbiter's binary identity independently — the Arbiter cannot self-attest. Publishes epoch nonces and digest ledgers for auditor verification.

Transparency Log. An append-only Merkle tree (RFC 6962) of signed receipts. Provides inclusion proofs (receipt exists in log), consistency proofs (log was not tampered with between time points), and split-view detection.

Relying Parties. Defenders, incident responders, auditors, regulators, procurement teams, insurers, and other parties that need to verify AI control-execution claims without trusting the operator or accessing protected content.

The specific cryptographic constructions, envelope schemas, and protocol details implementing this architecture are specified in registered OVERT Protocol Profiles. Conformant implementations SHALL use exactly one declared OVERT Protocol Profile; registration requirements are level-dependent (Section 22.6 — Level 1 may state `No Profile`, Levels 1–2 may use a validly self-declared

profile per Section 22.6.3, Levels 3–4 require a registered profile). Protocol Profile 1.0 is the initial registered profile (see Annex B).

4.3 Trust Architecture

Component	Trust Requirement	Rationale
Arbiter	Operator trusts their own deployment	Same trust model as enterprise firewall
Notary Network	Independent third-party trust; multi-party (t-of-n) where deployed	No single notary can forge attestations; structural independence from operator required for AAL-4
AI Model/Provider	Untrusted (System Under Test)	The entity being governed is the System Under Test; the attestation system does not trust its self-reports
Transparency Log	Public verifiability	Anyone can audit log consistency

The "untrusted SUT" designation applies to the relationship between the attestation layer and the AI model/provider. The attestation system does not trust the model's self-reports, the provider's claims, or the operator's logs. It produces independent verifiable records.

> NOTE: *The "untrusted SUT" designation is specific to the OVERT attestation relationship and is distinct from the NIST SP 800-207 Zero Trust Architecture for network security. SP 800-207 addresses network access assumptions; OVERT addresses attestation independence assumptions. The two are complementary but operate at different layers.*

4.4 Deployment Topology

Mode 1: Sidecar. For self-hosted models. The Arbiter runs as an enforcement module adjacent to the model runtime within the operator's infrastructure. Tool calls are intercepted at the service boundary.

Mode 2: Gateway. For SaaS-based models (OpenAI, Anthropic, Google). The Arbiter operates as a forward proxy. The operator routes orchestration traffic through the gateway, which governs tool execution even when the model runs in a third-party environment. Mode 2 may also be used for self-hosted models where the operator prefers a proxy deployment over a sidecar deployment. The distinction is architectural topology, not hosting model.

Both modes produce identical attestation receipts. The attestation artifacts concern what the operator's system did — not about the model's internals.

In both topologies, the arbiter may operate alongside a **scanner** (running as a runtime monitoring sidecar for threat detection and drift measurement) and a **local classifier** (running local evaluation models to categorize inputs, outputs, or agent behaviors).

4.5 Threat Model and Trust Assumptions

OVERT assumes the following threat model. Conformant implementations SHALL address each threat vector through the specified mitigation. Where a mitigation is marked SHOULD in the normative body (e.g., reproducible builds, binary transparency logs), the threat is addressed through disclosure and compensating controls rather than a hard requirement. The "Required Mitigation" column describes the intended mitigation approach; the normative strength (SHALL, SHOULD, MAY) of each specific control is defined in the referenced section.

Threat Vector	Description	Required Mitigation
Arbiter compromise	Malicious operator modifies or replaces arbiter binary	Notary-derived binary identity via hardware-rooted or hypervisor-attested measurement (NOT client-supplied claims)
Epoch-nonce prediction	Operator predicts sampling nonce to game which requests are evaluated	CSPRNG generation + commitment-reveal scheme (nonce committed at epoch start, revealed after close)
Co-epoch forgery	Attacker fabricates attestation receipts for a prior epoch	Strict current-epoch rule with bounded skew; stale submissions rejected
PRF gaming	Operator manipulates request ordering/content to avoid sampling	Policy-scoped key derivation as specified in the Protocol Profile; PRF deterministic from request commitment
Notary collusion	Subset of notaries collude to forge or suppress attestations	t-of-n notary agreement requirement in multi-notary topologies — no single entity controls t nodes (ATT-5.3); single-IAP topologies disclose the single-trust-entity limitation (Section 22.4) and gate revocation on joint operator–IAP authorization (RES-4.2)
Transparency log manipulation	Log operator tampers with append-only log	Split-view detection via published Signed Tree Heads; independent monitors

Threat Vector	Description	Required Mitigation
Clock manipulation	Operator skews system clock to place events in wrong epochs	Notary-issued epoch tokens with independent timestamp; bounded skew tolerance
Key compromise	Operator's KMS keys are exfiltrated	Key rotation procedures; epoch-scoped key derivation limits blast radius
Replay/rollback	Attacker replays old valid attestations	Epoch binding prevents cross-epoch replay; receipt includes monotonic sequence
DPL omission	Operator omits requests from Digest Publication Ledger	Coverage ratio computation; gap detection by auditors
Notary censorship	Notary selectively refuses to sign valid attestations	In multi-notary topologies, the t-of-n requirement prevents single-notary censorship; in single-IAP topologies, censorship surfaces as attestation gaps and exposure windows (ATT-3.4, RES-5) and in published uptime metrics (ATT-5.4)
IAP compromise / coercion / acquisition	Compromised, coerced, acquired, or negligent IAP issues fraudulent receipts or suppresses anomaly evidence	IAP compromise response plan (Section 4.7.1); multi-IAP option for higher assurance; receipt quarantine for affected epochs; annual transparency reports
Transparency log equivocation	Log operator presents different views (different STHs or inclusion proofs) to different parties	Mandatory independent log monitors (min. 2 for AAL-4); STH gossip protocol; consistency verification publication (Section 4.7.2)
Arbiter side-channel / memory scrape	Attacker exploits arbiter runtime to exfiltrate plaintext content or extract tenant_pepper key material	Process isolation and memory protection; attested key injection channel; TEE (SHOULD for AAL-3/4); runtime integrity monitoring (Section 4.7.3)
Build pipeline compromise	Compromised CI/CD injects malicious arbiter binaries that bypass enforcement	Reproducible builds (SHOULD); binary transparency logs (SHOULD); provenance verification before deployment (e.g., in-toto/SLSA attestations)
Classifier evasion	Inputs crafted to pass the local classifier or scanner while violating	Output containment at the boundary (ATT-3.5(a)); classifier version

Threat Vector	Description	Required Mitigation
	policy intent — distinct from prompt injection of the agent	binding; MEA-3 third-party adversarial testing
Engineered exposure windows	Operator induces IAP unavailability to obtain conveniently timed fail-open periods	Exposure-window signal (Section 4.6) makes induced gaps visible and reportable; RECONSTRUCTED receipts excluded from contemporaneous coverage (RES-5.2, Section 21.5)
Operator–IAP collusion	Structurally independent parties cooperate to misattest	Multi-IAP deployment; independent transparency-log monitors; split-knowledge key hierarchy limits what either party can forge alone
Prompt-injection-induced tool abuse	Untrusted input induces the agent to invoke tools, destinations, or data flows that are syntactically valid but unauthorized for the requesting context	Input filtering, pre-execution policy enforcement, parameter validation, provenance-aware authorization, and architectural separation (PRO-4, TOOL-1, TOOL-2, CAP-1, CAP-2)
Delegated-capability abuse	An agent relays, inherits, or composes capabilities beyond those originally granted through delegation, spawning, or topology changes	Capability mediation, spawn authorization, agent topology attestation, and inter-agent trust boundaries (CAP-1, CAP-2, MULTI-1, MULTI-2, DRIFT-3.4)
Approval-path abuse	A sensitive action is pushed through a weak or fatigued human approval path, including rubber-stamping or misbound reviewer identity	Approval gates, reviewer identity binding, approval velocity controls, review-quality monitoring, and separation of duties (TOOL-4, HITL-2, HITL-4, DRIFT-5)
Mediation scope evasion (selection bias)	Operator narrows mediation scope to exclude unfavorable traffic, making signals appear cleaner	Scope statement published to transparency log; scope changes attested with justification; coverage ratio references independent ingress metrics (Section 4.7.4)
Coverage blind spots / denominator ambiguity	The implementation cannot independently demonstrate what traffic or action volume formed the denominator for coverage and measurement claims	Published mediation scope statement, denominator source declaration, independent ingress metrics or profile-defined equivalent, and explicit disclosure of unverifiable

Threat Vector	Description	Required Mitigation
		denominators (Sections 4.7.4, 19.7.4, 22.1)

Trust assumptions:

Conformant implementations SHALL anchor arbiter and configuration measurements in an independently verifiable root of trust. The measurement pipeline SHALL satisfy the properties defined in Section 18.2: not controlled by the attester, rooted in a hardware or cryptographic trust anchor, and reproducible by an independent auditor. Client-supplied identity claims alone are insufficient for AAL-4 conformance.

> NOTE: *See Section 18.2 for examples of acceptable measurement pipelines including hardware-rooted attestation, hypervisor-attested measurement, and equivalent infrastructure defined in a registered Protocol Profile.*

4.6 Risk Signal Architecture

OVERT is designed to produce quantitative runtime signals from the attestation stream within the declared mediation scope. Whether a given signal is independently verifiable depends on the denominator source. Denominators fall into three classes:

- **Independently-Attested** — the measurement source is outside the operator's unilateral administrative control, or is cryptographically co-attested by an independent party: counterparty-attested counts, provider co-attestation (note below), IAP-observed ingress commitments.
- **Operator-Infrastructure** — operator-administered infrastructure telemetry (load balancer request counts, API gateway metrics). Stronger than bare assertion because it can be cross-checked against DPL completeness, but it remains under the operator's administrative control and is not independent.
- **Operator-Declared** — asserted by the operator without an infrastructure measurement source.

Signals whose denominators are Independently-Attested are classified as **independently verifiable signals**; signals whose denominators are Operator-Infrastructure or Operator-Declared are classified as **operator-dependent signals**. Both classes are useful; the distinction determines what a relying party can verify without trusting the operator.

> PROVIDER CO-ATTESTATION (INFORMATIVE EXTENSION POINT). *The strongest independent denominator requires a counterparty, because ingress metrics measured inside the operator's own infrastructure remain operator-controlled. For Mode 2 (SaaS-gateway) deployments, the model API provider observes every request and could publish per-customer request-count commitments that reconcile against the operator's coverage claims — making mediation-scope evasion cryptographically detectable rather than merely disclosed. A registered Protocol Profile MAY define a provider co-attestation extension binding such provider-published commitments to the operator's coverage denominator. This is a forward-looking extension point, not a Protocol Profile 1.0 requirement.*

Conformant implementations SHALL produce risk signals satisfying the following properties:

1. **Content-free derivation.** All signals SHALL be derivable without access to the operator's protected content. Signals are computed from the transparency log, published epoch data, mediation scope statements, and the registered Protocol Profile.
2. **Verifiability classification.** Each signal SHALL be classified as independently verifiable or operator-dependent based on the denominator class. A signal whose denominator is Operator-Infrastructure or Operator-Declared SHALL NOT be presented as independently verifiable in conformance documentation or public claims. Verifiability classification addresses the denominator axis only: violation-rate signals additionally depend on evaluator verdicts, whose integrity is established separately (Section 16.1, GOV-3.5, the Section 19.2 note). A signal SHALL NOT be represented as fully independently verified unless its denominator is Independently-Attested **and** its verdicts have been independently reproduced (Section 20.4, Annex G.1).
3. **Temporal granularity.** Signals SHALL be expressible as time series at epoch-level granularity.
4. **Statistical rigor.** Signals derived from sampling SHALL carry exact confidence intervals (not approximate), sample sizes, and auditor-reproducible methodology.
5. **Scope binding.** All signals SHALL reference the mediation scope statement, which defines signal denominators, and SHALL disclose the denominator source classification.

Risk signals support governance monitoring, security operations, audit, regulatory reporting, and external risk analysis. Signal definitions, formulas, and derivation procedures are specified in the registered Protocol Profile or companion signal specification. See Annex D for the signal framework and design rationale.

Level 3 and Level 4 conformance SHALL produce, at minimum, the following mandatory signal set per epoch:

1. **Coverage ratio** — the ratio of attested actions to total in-scope actions, referencing the declared denominator class (Independently-Attested, Operator-Infrastructure, or Operator-Declared) and its resulting verifiability classification.
2. **Violation rate with confidence interval** — the estimated policy violation rate with exact confidence bounds (per MEA-2.4).
3. **Gap accounting** — the count and percentage of attestation gap events (per ATT-3.4).
4. **Optimistic residue ratio** — the percentage of in-scope actions executed as optimistic residue (where applicable; per ATT-3.5(d)), reported both as claim-period average and worst single-epoch value.
5. **Exposure-window duration** — total duration and percentage of the claim period during which attestation coverage lapsed (fail-open periods, IAP unavailability, or unattested operation).

Registered Protocol Profiles MAY define additional signals. A Protocol Profile that does not produce the mandatory signal set is non-conformant for Level 3 and Level 4 claims.

Interpretation of signals for contractual, legal, regulatory, or financial purposes remains external to this standard. OVERT defines the signal architecture; it does not prescribe operational, legal, or actuarial conclusions.

4.7 Security Considerations

This section defines the minimum operational security baseline for OVERT deployments. These controls address threats identified in Section 4.5 that are not resolved by cryptographic format requirements alone. They establish the baseline protections needed for the OVERT trust model itself to remain credible, including response to IAP compromise, transparency-log monitoring, arbiter hardening, mediation-scope attestability, and anomaly triage. All requirements in this section are normative.

> TRANSPARENCY-LOG METADATA. *Even though receipts are content-free, a transparency log discloses metadata — per-tenant request volumes, timing, policy identifiers, and violation, override, and gap rates. This is a side channel and may be commercially sensitive. For this reason OVERT requires only that an independent* monitor *have access to the log (4.7.2), not that the log be world-readable; deployments MAY operate access-controlled logs with independent monitors rather than fully public ones without weakening the trust model. At production volume the complete per-epoch commitment set (Annex G.3.3) is large; a registered Protocol Profile MAY satisfy retrieval through a Merkle root plus on-demand openings rather than bulk publication.*

4.7.1 IAP COMPROMISE RESPONSE

Operators SHALL maintain an IAP compromise response plan. The plan SHALL define, at minimum:

(a) Criteria for initiating a compromise response, including but not limited to: confirmed key compromise, suspected coercion, change-of-control event affecting the IAP, and notification from the IAP of a suspected compromise.

(b) Quarantine procedures for receipts issued during the suspected compromise period. Receipts issued during a suspected compromise period SHALL be quarantined and SHALL NOT be presented as evidence of conformance pending investigation and disposition.

(c) Notification procedures for downstream relying parties that have consumed receipts from the affected IAP during the compromise window.

(d) Re-attestation procedures for epochs affected by the compromise, using an unaffected IAP or through independent verification.

(e) Criteria for restoring trust in a previously compromised IAP, or for permanently revoking trust and transitioning to an alternative IAP.

IAPs SHALL notify affected operators within 72 hours of detecting or suspecting a compromise event. The notification SHALL include the earliest and latest times bounding the suspected compromise window, the nature of the suspected compromise, and a list of affected operator identities or a statement that all operators should be considered potentially affected.

Operators that rely on a single IAP SHALL document the residual risk of single-IAP dependence in their conformance declaration and SHALL satisfy the following resilience requirements:

(f) **Portability escrow.** The operator SHALL maintain a tested portability package (key material escrow, configuration artifacts, and transparency log export) sufficient to onboard a replacement IAP without loss of historical attestation data.

(g) **Migration rehearsal.** The operator SHALL conduct an IAP migration rehearsal at intervals not exceeding 12 months and SHALL attest the rehearsal execution and measured activation time. The rehearsal SHALL demonstrate that the portability escrow enables functional attestation under a replacement IAP.

(h) **Failover procedure.** The operator SHALL define an IAP failover procedure with a target activation time documented in the conformance declaration. The target activation time SHALL be informed by the operator's measured rehearsal results and the current availability of qualified replacement IAPs, not by a fixed calendar period. If no qualified replacement IAP is available at the time of conformance, the operator SHALL disclose this limitation.

During the failover period, the deployment operates under fail-open or fail-closed procedures (RES-5) and SHALL report the unattested duration as an exposure window. Such deployments satisfy

AAL-4 for attestation independence, not for attestation resilience. Level 4 conformance claims based on single-IAP deployments SHALL disclose the IAP topology (`Single-IAP` vs. `Multi-IAP (t,n)` ; Section 22.4), the most recent rehearsal date and measured activation time, and any period during the claim window in which no qualified replacement IAP was available.

4.7.2 TRANSPARENCY LOG MONITOR DIVERSITY

AAL-4 deployments SHALL engage at minimum two independent transparency log monitors. For the purposes of this requirement, "independent" means that the monitors: (a) are operated by distinct legal entities with no common controlling interest, (b) do not share signing key infrastructure, and (c) operate from network vantage points not co-located with the transparency log operator's primary infrastructure.

Monitors SHALL perform the following verification functions:

1. **Consistency verification.** Monitors SHALL verify that each Signed Tree Head (STH) is consistent with all previously observed STHs for the same log, at intervals not exceeding the epoch boundary frequency.
2. **Inclusion verification.** Monitors SHALL periodically verify that receipts known to have been submitted to the log are included in the published tree. The sampling rate for inclusion verification SHALL be documented.
3. **Cross-monitor gossip.** Monitors SHALL exchange observed STHs with at least one other independent monitor. Detection of an STH discrepancy SHALL be treated as a log equivocation event and SHALL trigger immediate notification to all affected operators.

Monitors SHALL publish consistency verification results at a location accessible to relying parties. AAL-3 deployments SHOULD engage at least one independent transparency log monitor.

4.7.3 ARBITER HARDENING

Arbiter deployments SHALL implement process isolation and memory protection appropriate to the sensitivity of the content processed. At minimum:

(a) The arbiter process SHALL execute in an isolated process boundary with restricted system call access. The arbiter SHALL NOT share a process address space with application code.

(b) Memory regions containing operator key material (tenant_pepper, content-binding keys) SHALL be protected from access by other processes.

(c) Operator key material SHALL be injected into the arbiter via an attested channel — one in which the recipient can be cryptographically verified to be running the expected binary in the expected isolation state before key material is transmitted. Acceptable mechanisms include hardware-attested

sealed channels, mutually authenticated TLS with identity bound to co-epoch state, or KMS with policy-gated release tied to arbiter binary hash.

(d) Operator key material SHALL NOT be passed via environment variables in production deployments, persisted to disk in plaintext, logged at any verbosity level, or included in core dumps or crash reports.

(e) The arbiter SHALL zeroize sensitive key material from memory upon epoch rotation and upon process termination.

(f) For AAL-4 deployments, the arbiter SHALL either execute within a hardware-attested trusted execution environment (TEE) or the conformance claim SHALL explicitly disclose that arbiter isolation is software-only and not hardware-rooted. For AAL-3 deployments, the arbiter SHOULD be executed within a hardware-attested trusted execution environment (TEE).

4.7.4 MEDIATION SCOPE ATTESTABILITY

The mediation scope statement (as defined in Section 3.16) SHALL be published to the transparency log. The published scope statement SHALL include a machine-readable definition of the traffic classes within scope, any exclusions with stated justification, and the effective date.

Changes to the mediation scope SHALL be attested by the operator and logged to the transparency log with the previous scope statement hash, the new scope statement hash, a machine-readable justification, and the effective date of the change. Relying parties SHALL have access to the complete mediation scope history.

All Level 3 and Level 4 conformance claims SHALL identify the mediation scope statement hash, the declared coverage percentage of the mediation scope relative to its denominator, and the denominator class used for coverage and measurement claims (Section 4.6). Operator-administered infrastructure telemetry (e.g., load balancer request counts, API gateway telemetry) is Operator-Infrastructure class: it strengthens completeness checking against in-scope DPL metrics but SHALL NOT be represented as independently attested unless externally co-attested. Where no infrastructure measurement source outside the mediation scope exists, the denominator SHALL be marked Operator-Declared and the limitation disclosed. Level 4 claims SHALL use an Independently-Attested denominator — counterparty-attested counts, provider co-attestation (Section 4.6), IAP-observed ingress commitments, or a registered-Protocol-Profile equivalent whose independence from the operator is stated; absent such evidence, the implementation SHALL NOT claim Level 4 conformance for that scope.

> CONFORMANCE NOTE — LEVEL 4 DENOMINATOR FEASIBILITY. *Ordinary operator-administered API-gateway or load-balancer metrics do not satisfy this requirement, and Protocol Profile 1.0 does not define a registered equivalent denominator source. A Level 4 coverage claim therefore requires one of the independent sources above to actually exist for the deployment — for SaaS-gateway (Mode 2) topologies, typically IAP-observed ingress commitments or, once a registered profile defines the Section 4.6 extension point, provider co-attestation. Where no independent source exists yet, the correct claim is Level 3 with an Operator-Infrastructure (or Operator-Declared) denominator disclosure — not Level 4.*

4.7.5 ANOMALY TRIAGE OBLIGATION

Operators SHALL establish and maintain documented procedures for triaging, dispositioning, and escalating attested anomalies. Attested anomalies include but are not limited to: policy violations, override patterns exceeding baseline thresholds, drift signals breaching alert thresholds, exposure windows, coverage ratio shortfalls, and receipt verification failures.

The anomaly triage procedure SHALL define:

(a) **Classification criteria.** A severity classification scheme with defined criteria based on type, frequency, and potential impact.

(b) **Response timelines.** Maximum time-to-acknowledge and time-to-disposition for each severity level. Critical anomalies SHALL be acknowledged within 24 hours and dispositioned within 7 days. Security-critical anomalies — including binary identity mismatch, co-epoch binding violation, transparency log equivocation, and arbiter integrity failure — SHALL be acknowledged within 1 hour and SHALL trigger immediate containment action (circuit breaker, scope isolation, or fail-closed) pending disposition.

(c) **Disposition categories.** At minimum: confirmed violation (remediate), false positive (document rationale), accepted risk (document rationale and approval authority), and escalation (to identified authority).

(d) **Escalation paths.** Named roles or functions responsible for escalation decisions at each severity level.

(e) **Record retention.** Triage records SHALL be retained for the period defined in the operator's retention schedule and SHALL be available for audit.

> NOTE — ADVERSE INFERENCE IMPLICATIONS. *An attested anomaly constitutes a record of a condition observed and recorded by the system. Failure to act on attested anomalies — or failure to*

> *maintain triage procedures that ensure anomalies are reviewed — may constitute constructive notice of the conditions evidenced by those anomalies. Operators should consult legal counsel regarding the evidentiary implications of attested anomaly records in their jurisdiction. This note is informative and does not create legal obligations beyond those stated normatively in this section.*

4.8 Cross-Boundary Attestation Protocol

Many real-world AI deployments involve multiple trust boundaries in sequence — for example, an ambient scribe producing a clinical note that feeds a clinical decision support system, which queries a drug interaction database, which in turn calls a genomics API. Each boundary operator may independently deploy OVERT attestation. This section defines how attestation receipts are linked across trust boundaries to enable end-to-end verification without requiring protected content to cross any boundary.

4.8.1 PURPOSE

Cross-boundary attestation enables relying parties to verify that governance controls executed across an entire multi-provider workflow, not merely within a single operator's boundary. The protocol achieves this by linking receipts across trust boundaries using cryptographic references — specifically, by including the upstream receipt's attestation_id hash in the downstream receipt. No protected content crosses any trust boundary; only receipt hashes (attestation_id references) are exchanged. Throughout Section 4.8 and Annex G, a receipt's `attestation_id` is its attestation-hash field (Annex B.6) — the cryptographic digest of the attested envelope — under the name reflecting its role as a reference key.

4.8.2 PARENT ATTESTATION REFERENCE

Each receipt generated within a downstream trust boundary MAY reference an upstream receipt by including the upstream receipt's attestation_id hash as a `parent_attestation_id` field in the downstream receipt. The `parent_attestation_id` SHALL be the SHA-256 hash of the upstream receipt's attestation_id as published in the upstream operator's transparency log. The input octets for this computation SHALL be the upstream `attestation_id`'s published wire form — the 64-character lowercase hexadecimal ASCII string (Annex B.6) — with no separators or terminators; published cross-boundary test vectors SHALL cover this derivation (Section 22.6.2). Where multiple upstream receipts contributed to a single downstream action, the downstream receipt MAY include multiple `parent_attestation_id` entries.

The `parent_attestation_id` field is OPTIONAL for workflows that do not cross trust boundaries. For cross-boundary workflows at Level 3 or above, the `parent_attestation_id` field SHALL be populated when the downstream operator has access to the upstream receipt's attestation_id.

4.8.3 CROSS-BOUNDARY DAG RECONSTRUCTION

Relying parties can reconstruct an end-to-end directed acyclic graph (DAG) of attestation across providers by following the `parent_attestation_id` hash chains. Each node in the DAG represents a receipt within a single trust boundary; each edge represents a `parent_attestation_id` reference linking a downstream receipt to an upstream receipt. The DAG enables verification that governance controls executed at every boundary in a multi-provider workflow.

DAG reconstruction SHALL NOT require access to protected content from any boundary. Relying parties reconstruct the DAG using only: (a) receipt metadata and `parent_attestation_id` fields from each boundary's transparency log, (b) publicly verifiable receipt signatures and co-epoch bindings, and (c) published cross-boundary scope statements (Section 4.8.5).

4.8.4 NO CONTENT CROSSING

Only receipt hashes (attestation_id references) cross trust boundaries under this protocol. The `parent_attestation_id` is a hash of a receipt identifier — it does not contain, encode, or enable reconstruction of any protected content from the upstream boundary. The non-egress property (Section 17, Design Principle 2) is preserved across all trust boundaries in the chain.

4.8.5 CROSS-BOUNDARY SCOPE STATEMENT

Each boundary operator participating in cross-boundary attestation SHALL publish a cross-boundary scope statement to its transparency log. The cross-boundary scope statement SHALL declare:

(a) Which upstream attestation sources the operator accepts and links (identified by upstream operator identity and upstream transparency log URI).

(b) The upstream receipt validation policy: whether the operator performs the generation-time checks of Section 4.8.7 before linking (transparency-log inclusion is verified per Section 4.8.6 during DAG validation, not at link time), or accepts upstream attestation_id references without independent validation.

(c) The effective date and version of the cross-boundary scope statement.

Changes to the cross-boundary scope statement SHALL be attested and logged with the same change-attestation requirements as mediation scope statement changes (Section 4.7.4).

4.8.6 RECEIPT CHAIN VALIDATION

Verifiers performing cross-boundary DAG validation SHALL validate the full chain by checking, for each receipt in the DAG:

(a) The receipt's signature validity (per the receipt's boundary operator's notary network).

(b) The receipt's co-epoch binding integrity (binary identity, network state, epoch currency).

(c) The `parent_attestation_id` reference integrity: the referenced upstream receipt EXISTS in the upstream operator's transparency log and the `parent_attestation_id` value matches the SHA-256 hash of the upstream receipt's attestation_id.

(d) The downstream operator's cross-boundary scope statement declares acceptance of the upstream attestation source.

A cross-boundary verification is valid only if every receipt in the chain passes all four checks. Partial chain validation (where some links are verified and others are not) SHALL be reported as partial, not as full cross-boundary verification.

The checks in this subsection are **verification-time** procedures performed by relying parties during DAG reconstruction, asynchronously to receipt generation. The generation-time record is `parent_reference_status` (Section 4.8.7); in particular, check (c) — transparency-log inclusion — cannot generally be completed at generation time, because Phase 3 log inclusion is itself asynchronous (ATT-3.3).

4.8.7 FAILURE HANDLING

The downstream receipt SHALL include a `parent_reference_status` field recording the outcome of the **generation-time** checks — the structural checks the downstream arbiter can complete synchronously at intercept. Transparency-log inclusion is not a generation-time check: an upstream Phase 2 provisional receipt's log entry may legitimately not yet exist when the downstream receipt is generated (ATT-3.3). Log-inclusion verification (Section 4.8.6(c)) is performed by relying parties during asynchronous DAG validation.

Status	Description
`VALID`	Upstream reference was available and passed the generation-time checks: well-formed reference, `parent_attestation_id` computed per Section 4.8.2, and — where the upstream receipt itself was presented — signature and co-epoch binding verified
`UNAVAILABLE`	Upstream receipt or reference was not presented at intercept; `parent_attestation_id` could not be populated
`INVALID`	Upstream receipt or reference was available but failed a generation-time check: malformed reference, signature verification failure, or co-epoch binding failure

Status	Description
`TIMEOUT`	Upstream endpoint did not respond within the profile-defined timeout for the generation-time checks

When the `parent_reference_status` is anything other than `VALID`, the downstream receipt SHALL still be generated (attestation of the downstream boundary's own governance controls proceeds regardless of upstream availability), but the cross-boundary chain is incomplete at that link. Relying parties SHALL treat incomplete links as gaps in cross-boundary verification, not as failures of the downstream boundary's own attestation. `parent_reference_status` is a generation-time record, not an assertion of transparency-log inclusion; relying parties SHALL NOT treat `VALID` as a substitute for the full Section 4.8.6 chain validation.

4.8.8 NORMATIVE REQUIREMENTS

All cross-boundary attestation controls in this section are normative at AAL-4 for Level 3 and Level 4 conformance claims involving cross-boundary workflows. Level 1 and Level 2 claims are not required to implement cross-boundary attestation. Implementations that do not participate in cross-boundary workflows are not required to implement this section.

The specific `parent_attestation_id` field encoding, `parent_reference_status` enumeration, cross-boundary scope statement schema, and DAG reconstruction procedures are specified in the registered Protocol Profile.

OVERT is an open standard for public adoption and co-development. Implementation details are specified in registered OVERT Protocol Profiles.

Editorial contact: overt-review@glacis.io Protocol Profile Registry: See Annex B and Section 22.6

PART 2: GOVERNANCE DOMAINS

> *Six obligations turn an intention to govern into evidence that it happened — govern, identify, protect, attest, measure, respond. A domain is not satisfied by having a control; it is satisfied by being able to prove the control ran.*

OVERT organizes its core requirements into governance domains that together define the organizational and infrastructure control plane for verifiable AI governance and AI runtime defense. Part 2 covers organizational governance, system identification, boundary enforcement, attestation generation and verification, measurement, and response. Read together, these domains define how an operator declares policy, constrains AI actions at the boundary, measures in-scope behavior, and preserves evidence of control execution.

Where a control table in Part 2 or Part 3 restates an architecture requirement specified in Part 4, the Part 4 specification prevails in any conflict. Control tables summarize; Part 4 defines.

5. Domain 1: GOVERN — Organizational Governance

Scope: Policies, accountability structures, training, culture, and supply chain governance. Maps to NIST AI RMF GOVERN and ISO 42001 Clauses 4–7.

These controls are organizational in nature. Attestation assurance level requirements are AAL-1–AAL-2 for policy and process controls, with AAL-4 required where machine-verifiable artifacts are possible.

GOV-1: AI Governance Policy

Requirement: The organization SHALL establish, document, and maintain an AI governance policy covering all AI systems within scope.

Attestation Assurance Level: AAL-1 (policy document) + AAL-4 (machine-readable policy artifact published to transparency log)

ID	Control	Attestation Artifact	Level
GOV-1.1	Publish AI governance policy covering intended uses, risk tolerances, accountability structures, and applicable regulations	Policy document in human-readable format	AAL-1
GOV-1.2	Publish machine-readable policy artifact (OSCAL or OVERT policy schema) to transparency log with cryptographic timestamp	Signed policy artifact with transparency log inclusion proof	AAL-4
GOV-1.3	Review and update policy at planned intervals (minimum: annually) with documented change justification	Transparency log entries showing versioned policy updates	AAL-4

GOV-2: Accountability and Roles

Requirement: The organization SHALL assign and document roles and responsibilities for AI risk management, including a designated accountable individual for each AI system in scope.

Attestation Assurance Level: AAL-2

ID	Control	Attestation Artifact	Level
GOV-2.1	Assign accountable owner for each AI system with documented authority and responsibility	Organizational chart or RACI matrix	AAL-2
GOV-2.2	Define and document change approval authority — which system changes require formal review and by whom	Change approval policy with designated approvers per change type	AAL-2
GOV-2.3	Ensure separation of duties: personnel responsible for AI development SHALL NOT serve as sole approver of their own work	Documented approval records showing independent sign-off	AAL-2

GOV-3: Risk Taxonomy

Requirement: The organization SHALL establish and maintain a risk taxonomy categorizing AI-specific risks with severity levels, examples, and remediation procedures.

Attestation Assurance Level: AAL-2 + AAL-4 (taxonomy published as machine-readable artifact)

ID	Control	Attestation Artifact	Level
GOV-3.1	Define risk categories covering: harmful outputs, out-of-scope outputs, hallucinated outputs, unauthorized tool actions, data leakage, bias, and domain-specific risks	Risk taxonomy document	AAL-2

ID	Control	Attestation Artifact	Level
GOV-3.2	Assign severity levels to each risk category with escalation criteria	Severity matrix with escalation procedures	AAL-2
GOV-3.3	Publish machine-readable risk taxonomy to transparency log and reference it in attestation policy configuration	Signed taxonomy artifact with log inclusion proof	AAL-4
GOV-3.4	Review taxonomy at intervals defined in the operator's risk management policy, not to exceed 12 months or as defined by applicable regulation; update based on incidents, emerging threats, and regulatory changes	Transparency log entries showing versioned taxonomy updates	AAL-4
GOV-3.5	Require that all attestation policy artifacts (the policy_hash referenced in enforcement receipts) be signed by a designated Qualified Risk Officer and reference a published safety baseline (e.g., the NIST AI RMF Generative AI Profile (NIST AI 600-1), OWASP Agentic Top 10, or sector-specific baseline). The receipt validates enforcement; the signature validates the rule set. [See Annex C: Design Rationale, "Policy-Quality Gap" analysis]	Policy artifact signed by named risk officer with baseline reference in transparency log	AAL-4

GOV-4: Supply Chain and Third-Party Governance

Requirement: The organization SHALL establish governance processes for third-party AI components, including foundation models, data sources, and tools.

Attestation Assurance Level: AAL-2

ID	Control	Attestation Artifact	Level
GOV-4.1	Conduct documented due diligence on foundation model providers covering: data handling, security practices, safety testing, and contractual commitments	Vendor assessment records	AAL-2
GOV-4.2	Maintain inventory of all third-party AI components with version tracking and provenance documentation	Component inventory with version history	AAL-2
GOV-4.3	Establish contractual requirements for third-party components including: notifica-	Contract excerpts or attestation from legal review	AAL-2

ID	Control	Attestation Artifact	Level
	tion of material changes, incident disclosure, and cooperation with audits		

GOV-5: AI Disclosure

Requirement: The organization SHALL implement disclosure mechanisms informing users when they interact with AI systems.

Attestation Assurance Level: AAL-2 (product demonstrations) + AAL-4 (receipt reference in response metadata, GOV-5.6)

ID	Control	Attestation Artifact	Level
GOV-5.1	Implement disclosure for text-based AI interactions ("You are chatting with an AI")	Product screenshot or recording	AAL-2
GOV-5.2	Implement disclosure for voice-based AI interactions (spoken notification at session start)	Audio recording or transcript	AAL-2
GOV-5.3	Label AI-generated content in machine-readable format (C2PA Content Credentials, metadata, or watermarks)	Content sample with embedded metadata	AAL-2
GOV-5.4	Disclose when autonomous AI agents perform actions without step-by-step human oversight	Product demonstration showing agent disclosure	AAL-2
GOV-5.5	Respond accurately when users ask "Are you AI?"	Product demonstration	AAL-2
GOV-5.6	Include a receipt reference (receipt_id or receipt_hash) in AI response metadata, enabling end users to dispute a specific interaction by citing its cryptographic identifier. The operator can then locate the exact attestation record, verifiable record, and policy evaluation for that transaction. Where AI-generated content carries C2PA Content Credentials (GOV-5.3), the receipt reference SHOULD be embedded as a C2PA assertion, so that any downstream consumer of the content can walk from the artifact to its attested enforcement record — end-to-end output provenance	Receipt reference in response metadata; dispute resolution procedure; C2PA assertion where content is labeled	AAL-4

6. Domain 2: IDENTIFY — Risk Identification and Mapping

Scope: Context establishment, AI system categorization, impact assessment, and risk mapping. Maps to NIST AI RMF MAP and ISO 42001 Clause 6.

IDE-1: System Context and Categorization

Requirement: The organization SHALL document the intended purpose, deployment context, capabilities, and limitations of each AI system in scope.

ID	Control	Attestation Artifact	Level
IDE-1.1	Document intended purposes, target users, deployment settings, and applicable laws/regulations	System context document	AAL-1
IDE-1.2	Categorize system capabilities: text generation, voice generation, image generation, automation/agentic, or multimodal	Capability classification in machine-readable format	AAL-2
IDE-1.3	Document system knowledge limits and conditions under which outputs may be unreliable	Technical limitations document	AAL-1

IDE-2: AI System Impact Assessment

Requirement: The organization SHALL assess and document potential consequences of each AI system on individuals, groups, and society.

ID	Control	Attestation Artifact	Level
IDE-2.1	Conduct impact assessment for each AI system covering: potential benefits, potential harms, affected populations, and severity of adverse outcomes	Impact assessment document	AAL-2
IDE-2.2	Consider domain-specific and jurisdictional requirements in impact assessments	Jurisdictional analysis	AAL-2
IDE-2.3	Incorporate impact assessment results into risk treatment planning	Risk treatment plan referencing impact assessment	AAL-2

7. Domain 3: PROTECT — Boundary Enforcement and Containment

Scope: Runtime enforcement of governance policy at the boundary between AI systems and external resources. This is the domain where OVERT departs from existing frameworks by requiring enforcement infrastructure, not just policy documentation.

All controls in this domain require AAL-4 attestation artifacts.

PRO-1: Boundary Enforcement

Requirement: All AI system interactions with external resources (tool calls, API requests, data access, network egress) SHALL pass through an enforcement layer that evaluates actions against defined policy before execution.

ID	Control	Attestation Artifact	Level
PRO-1.1	Deploy an enforcement arbiter at the boundary between AI system and external resources	Co-epoch attested binary hash proving arbiter deployment	AAL-4
PRO-1.2	Evaluate every outbound action against customer-defined policy before execution	Per-action attestation receipt (permit or deny)	AAL-4
PRO-1.3	Block actions that violate policy; generate denial receipt with policy reference	Denial receipts in transparency log	AAL-4
PRO-1.4	Generate permit receipt for allowed actions, cryptographically bound to policy version and system configuration	Permit receipts with co-epoch binding	AAL-4

PRO-2: Network Isolation and Egress Control

Requirement: AI system network egress SHALL be restricted to approved destinations and attested at each epoch.

ID	Control	Attestation Artifact	Level
PRO-2.1	Implement destination allowlists restricting AI system network egress to approved endpoints	Attested network policy hash (NETATT)	AAL-4
PRO-2.2	Attest network isolation state at each epoch covering, at minimum: the effective egress policy, the enforcement component identity, and the TLS certificate pin set (Section	Co-epoch NETATT covering the Section 18.3 minimum set	AAL-4

ID	Control	Attestation Artifact	Level
	18.3). Operators MAY include additional deployment-specific inputs — network policy definitions (hashing the policy controller input, not dynamic ephemeral rules), network controller identity, eBPF programs, CNI configuration, runtime environment variables affecting AI behavior. The minimum input set is specified in the registered Protocol Profile; Section 18.3 prevails in any conflict		
PRO-2.3	Detect and attest any network configuration changes within an epoch	Configuration drift detection via NETATT hash comparison	AAL-4

PRO-3: Rate Limiting and Velocity Controls

Requirement: AI system actions SHALL be subject to rate limits and velocity controls with attested enforcement.

ID	Control	Attestation Artifact	Level
PRO-3.1	Enforce per-action, per-user, and per-epoch rate limits on tool calls and API requests	Rate limit enforcement receipts	AAL-4
PRO-3.2	Implement escalating restrictions for anomalous velocity patterns	Velocity enforcement attestations	AAL-4
PRO-3.3	Require human approval gates for actions exceeding defined thresholds	Approval gate attestations with identity binding	AAL-4

PRO-4: Input and Output Filtering

Requirement: AI system inputs and outputs SHALL be filtered for safety policy violations with attested enforcement.

ID	Control	Attestation Artifact	Level
PRO-4.1	Filter inputs for adversarial content, prompt injection, and policy violations before model processing	Filter enforcement receipts	AAL-4
PRO-4.2	Filter outputs for harmful content, out-of-scope content, PII leakage, and policy violations before delivery	Filter enforcement receipts	AAL-4
PRO-4.3	Sanitize outputs to prevent security vulnerabilities (XSS, injection, unsafe URLs) in downstream systems	Sanitization enforcement receipts	AAL-4

PRO-5: Data Isolation

Requirement: Customer data SHALL be isolated with attested enforcement of tenant boundaries.

ID	Control	Attestation Artifact	Level
PRO-5.1	Enforce logical and/or physical separation of customer data across tenants	Data isolation attestation	AAL-4
PRO-5.2	Attest that AI system prompts and responses do not cross tenant boundaries	Cross-tenant isolation receipts	AAL-4
PRO-5.3	Implement PII detection and filtering with attested enforcement	PII detection receipts (no content egress)	AAL-4

8. Domain 4: ATTEST — Attestation Generation and Verification

Scope: The core attestation infrastructure. This domain specifies how attestation artifacts are generated, stored, attested, and made verifiable by third parties.

ATT-1: Non-Egress Attestation Architecture

Requirement: The attestation protocol SHALL NOT require transmission of protected content outside the operator environment. Conformant receipt-service interfaces SHALL accept only cryptographic commitments and profile-defined metadata.

ID	Control	Attestation Artifact	Level
ATT-1.1	Canonicalize AI request/response payloads using deterministic encoding as specified in the registered Protocol Profile	Documented encoder specification with version-pinned encoder_id	AAL-4
ATT-1.2	Compute request digests as cryptographic hashes of canonical encodings; derive keyed commitments using a keyed cryptographic function with tenant-scoped keys held exclusively in the operator's KMS. Only keyed commitments cross the trust boundary — never raw digests. This prevents rainbow table reversal of low-entropy content (PII, SSNs) by any party with ledger access	Receipt service schema accepts only keyed commitments; raw digests rejected; closed schema (unknown fields rejected)	AAL-4

ID	Control	Attestation Artifact	Level
ATT-1.3	Store attestation artifacts (full payloads, policy evaluations, metadata) in content-addressable storage within the operator's environment	Local CAS deployment with retention policy	AAL-4
ATT-1.4	Constrain attestation egress to a single receipt service endpoint over TLS with certificate pinning as defined in the registered Protocol Profile	Attested certificate pin set in NETATT	AAL-4

> NOTE: *For streaming outputs (Server-Sent Events), implementations MAY use rolling commitment constructions or chunked attestation as defined in the registered Protocol Profile. The full-payload commitment model described here is the normative baseline; streaming extensions are profile-specific.*

> NOTE: *The keyed commitment requirement (ATT-1.2) specifies properties, not constructions. The keyed function SHALL be computationally infeasible to invert without knowledge of the operator secret. Protocol Profile 1.0 satisfies this requirement using HMAC-SHA256 with keys derived via HKDF. Alternative profiles MAY use different keyed commitment schemes provided they satisfy the non-egress and irreversibility properties defined above.*

ATT-2: Co-Epoch Binding

Requirement: Every attestation receipt SHALL be cryptographically bound to the system's binary identity and network isolation state during a bounded time interval.

ID	Control	Attestation Artifact	Level
ATT-2.1	Establish heartbeat epochs with configurable epoch duration (recommended: 300 seconds) with notary-issued bearer tokens	Epoch heartbeat receipts with notary signatures	AAL-4
ATT-2.2	Arbiter binary identity SHALL be derived by the notary through a measurement pipeline that is (a) not controlled by the attester, (b) rooted in a hardware or cryptographic trust anchor, and (c) reproducible by an independent auditor given the measurement policy. Client-supplied identity claims are insuffi-	Notary-derived binary identity in receipt	AAL-4

ID	Control	Attestation Artifact	Level
	cient for AAL-4 conformance. See Section 18.2 for acceptable measurement pipelines		
ATT-2.3	Bind every receipt to the current epoch, binary identity, and network attestation hash	Co-epoch receipt schema with all three bindings	AAL-4
ATT-2.4	Reject any attestation submission not in the current epoch (strict current-epoch rule; bounded skew tolerance as defined in the registered Protocol Profile, recommended: <=2 seconds)	Deterministic rejection: ERR_STALE_EPOCH	AAL-4

ATT-3: Three-Phase Attestation

Requirement: The attestation system SHALL support synchronous enforcement, synchronous provisional receipts, and asynchronous full attestation to meet latency requirements without compromising attestation artifact quality.

ID	Control	Attestation Artifact	Level
ATT-3.1	**Phase 1 — Enforcement:** Evaluate action against policy synchronously. [Informative targets: <5ms P50 local, <25ms P50 distributed. Specific latency requirements are defined in the registered Protocol Profile.]	Enforcement decision recorded in arbiter	AAL-4
ATT-3.2	**Phase 2 — Provisional Receipt:** Generate locally-signed attestation commitment synchronously with explicit provisional status	Provisional receipt with arbiter signature	AAL-4
ATT-3.3	**Phase 3 — Full Attestation:** Notary network validates and counter-signs asynchronously using topology-appropriate notary verification — a single independent signer for Single-IAP deployments, or threshold/multi-signature t-of-n verification for Multi-IAP deployments (ATT-5.3, Section 22.4) — with cryptographic constructions specified in the registered Protocol Profile. Implementations SHALL support cryptographic agility including post-quantum migration paths. After January 1, 2031, pure classical signature schemes are	Full receipt with topology-appropriate notary signature (single independent signer or t-of-n) and transparency log inclusion proof	AAL-4

ID	Control	Attestation Artifact	Level
	non-conformant; hybrid classical + post-quantum constructions, or pure post-quantum constructions, as specified in the registered Protocol Profile SHALL be used		
ATT-3.4	Track and report provisional receipts that are not upgraded to full attestation within the SLA window as explicit "attestation gap" events	Gap accounting in audit reports	AAL-4
ATT-3.5	**Optimistic Enforcement Mode:** Because Phase 3 notary counter-signature is asynchronous by architecture (ATT-3.3), execution proceeds after Phase 2 (Provisional Receipt). The normative provisions (a)–(f) following this table govern which actions may proceed on a provisional receipt alone: side-effecting action classes (provision (a)) SHALL carry independent authorization before execution, and the optimistic-residue caps bound what may execute with neither	Optimistic mode declaration in policy with explicit tool-call classification; capability artifacts or synchronous attestation for side-effecting classes; circuit breaker on notary rejection	AAL-4

ATT-3.5 normative provisions. Phase 3 notary counter-signature is asynchronous (ATT-3.3); execution after Phase 2 is therefore the universal execution model of this standard. The provisions below govern what may proceed on a provisional receipt alone, and bound the **optimistic residue** — side-effecting actions (Write, Transact, Delete, Modify, and operator-defined side-effecting classes) that execute with neither independent pre-authorization nor synchronous attestation. Read-class actions executing on Phase 2 alone under the output-containment conditions of (a) are the intended optimistic path, not residue.

(a) **Action classification and independent authorization.** Optimistic execution on Phase 2 alone is permitted for actions classified Read-only, subject to output containment: a Read-class action that returns sensitive content (PII, PHI, classified data, or credentials) remains optimistic-eligible only where delivery is to the authenticated, authorized subject of the session over a sink declared in the mediation scope statement. Routing of sensitive content to any undeclared or uncontrolled sink SHALL be blocked regardless of action classification — the restriction binds on exfiltration paths, not on governed delivery to the authorized subject. Deployments SHALL ensure that actions classified Write, Transact, Delete, or Modify carry independent authorization before execution — satisfied by either (i) a valid capability artifact (3.24) covering the tool or action class, the session, and the time of execution, verified at the enforcement point without a network dependency; or (ii) synchronous Phase 3 notary attestation, where the deployment elects to block — **except within**

the bounded optimistic residue governed by provisions (c) through (f). An action in these classes executed with neither is optimistic residue: permitted only within the (e) caps, gap-classified under (c), and disclosed under (d). Misclassification of a side-effecting action as Read-only is a governance failure and SHALL be reported as a conformance deviation.

(b) **Circuit breaking.** If Phase 3 subsequently rejects a provisional receipt (notary detects drift, binary mismatch, or policy violation), the system SHALL trigger a circuit breaker (TOOL-3.3) halting subsequent requests from the same agent or session.

(c) **Gap classification.** If Phase 3 does not complete within the profile-defined SLA window, the action SHALL be classified as an attestation gap event under ATT-3.4 and SHALL NOT count as fully attested coverage.

(d) **Reporting.** Provisional-only periods SHALL be reported as a distinct coverage class in risk signals, separated from fully attested periods. For Level 3 and Level 4 conformance claims, the percentage of in-scope actions executed as optimistic residue during the claimed period SHALL be disclosed in the conformance statement, as both the claim-period average and the worst single-epoch value. The disclosure SHALL additionally include the absolute residue counts per action class for the claim period and for the worst single epoch: fraction caps bound the rate, and counts make in-class volume dilution visible.

(e) **Residue caps.** The caps below apply to the optimistic residue, independently per side-effecting action class (Write, Transact, Delete, Modify, and any operator-defined side-effecting classes; Read-class optimistic execution is governed by (a) and is not residue). Actions executed under a valid capability artifact or under synchronous Phase 3 attestation are independently authorized and do not count toward the residue.

Conformance Level	Worst Single Epoch	Claim-Period Average
Level 4	25% of in-scope actions	15% of in-scope actions
Level 3	40% of in-scope actions	25% of in-scope actions

Deployments exceeding a cap for any action class SHALL claim Level 3 for the affected scope until the residue is reduced; a Level 3 deployment exceeding either of its caps SHALL report the excess as a conformance deviation. This structure ensures that Evidence-Grade conformance reflects predominantly independently authorized or independently verified execution, and that optimistic execution cannot be concentrated in high-risk epochs and diluted by low-risk traffic. Because every side-effecting class is capped at the same fractions, the aggregate side-effecting residue is bounded by the same values — splitting or merging operator-defined classes cannot raise it; the absolute counts disclosed under (d) expose dilution within a class.

(f) **Eligibility governance.** The operator's action classifications and capability-issuance policy SHALL be documented in the mediation scope statement and SHALL be subject to review by the IAP or auditor upon request. For Level 4 conformance claims, this review is part of the conformance assessment procedure (Section 22.8.3(e)), not merely on request.

ATT-4: Transparency Log

Requirement: All receipts SHALL be recorded in an append-only transparency log providing inclusion proofs, consistency proofs, and split-view detection.

ID	Control	Attestation Artifact	Level
ATT-4.1	Operate an append-only Merkle tree log (RFC 6962) for all attestation receipts	Signed Tree Heads (STH) with root hash, tree size, and timestamp	AAL-4
ATT-4.2	Provide inclusion proofs for any receipt on demand	Merkle inclusion proof path	AAL-4
ATT-4.3	Provide consistency proofs between any two Signed Tree Heads	Merkle consistency proof	AAL-4
ATT-4.4	Publish Signed Tree Heads at regular intervals for independent monitoring and split-view detection	Published STH records	AAL-4

ATT-5: Notary Network Governance

Requirement: The governance, composition, and independence of the notary network SHALL be explicitly defined, documented, and verifiable. The credibility of AAL-4 attestation artifacts depends entirely on the structural independence of the notaries from the operator being attested.

ID	Control	Attestation Artifact	Level
ATT-5.1	Define and publish the notary network governance model. Four models are normative: **(a) Platform-operated:** An Independent Attestation Provider (IAP) operates all notary nodes. Satisfies the AAL-4 independence requirement where the IAP is structurally independent of the AI system operator; regardless of node count, a platform-operated deployment is a single trust entity and SHALL NOT claim the multi-party t-of-n resilience property (ATT-5.3) — its topology disclosure is `Single-IAP` (Section 22.4). **(b) Consortium:** Nodes operated	Governance model documentation with transparency log proof	AAL-4

ID	Control	Attestation Artifact	Level
	by a combination of operator, insurer, auditor, and IAP. Satisfies AAL-4 where the structurally independent members control sufficient nodes that no single entity controls t or more (ATT-5.3). **(c) Customer-operated:** Full sovereignty for maximum-security environments. Satisfies AAL-3 (the operator controls the notary). **(d) Hardware-enforced (First-Party Enclave):** Cloud provider or operator uses native hardware-attested TEEs (e.g., AWS Nitro Enclaves, Azure Confidential Computing) as the notary, with attestation quotes verifiable by any relying party. Satisfies AAL-3 with enhanced measurement properties; satisfies AAL-4 where the enclave attestation is validated by an independent third party. Any additional governance model MAY be proposed for registration provided it satisfies the independence and publishability requirements defined in this standard. Publish governance model to transparency log		
ATT-5.2	Publish the current notary set: node identities, operating entities, geographic distribution. Attest any changes to the notary set with topology-appropriate signatures from both the outgoing and incoming set (t-of-n where multi-notary; the single independent signer's outgoing and incoming keys in single-IAP deployments)	Notary set transition attestation in transparency log	AAL-4
ATT-5.3	AAL-4 SHALL require that the notary service be operated by an entity structurally independent of the AI system operator. For deployments claiming the multi-party resilience property (topology `Multi-IAP (t,n)`, including consortium models): no single organizational entity SHALL control t or more notary nodes, and no two notary nodes in the same threshold set SHALL share a common ulti-	Notary independence attestation	AAL-4

ID	Control	Attestation Artifact	Level
	mate corporate parent, common hosting infrastructure provider, or common jurisdiction of incorporation. For platform-operated models: publish geographic and infrastructure diversity guarantees (operational resilience; not the multi-party trust property). For hardware-enforced models: publish TEE attestation quote verification procedures		
ATT-5.4	Publish notary network uptime, availability, and attestation latency metrics at regular intervals	Notary health metrics in transparency log	AAL-4

Architectural note: AAL-4 requires structural independence between the notary service and the AI system operator. A single independent third-party notary satisfies this requirement. Multi-entity notary sets (consortium models) provide additional resilience — no single entity compromise can forge attestations — but are not required for AAL-4 conformance. The claim string and ABD disclose which topology applies (`Single-IAP` or `Multi-IAP (t,n)`; Section 22.4, Section 22.10); the t-of-n trust property may be claimed only where ATT-5.3's entity-diversity requirements hold, and revocation authorization follows the topology (RES-4.2). Platform-operated notaries are operationally simpler but introduce a dependency on the IAP's integrity. Customer-operated notaries satisfy AAL-3, not AAL-4. Hardware-enforced models satisfy AAL-3 unless validated by an independent party. The standard does not mandate a single model — it mandates that the chosen model is explicit, published, and auditable, and that the AAL claim matches the achieved independence.

9. Domain 5: MEASURE — Statistical Safety Assessment

Scope: Continuous, quantitative measurement of AI system behavior with cryptographically verifiable sampling. Maps to NIST AI RMF MEASURE but adds the rigor that MEASURE 2.x references but does not specify.

This standard defines a single normative auditor-reproducible sampling and measurement method: the Statistical Safety Signal Protocol (S3P) defined in MEA-2. MEA-1 specifies the deterministic sampling infrastructure that S3P relies upon. Alternative sampling constructions SHALL be specified in a registered Protocol Profile and demonstrated to preserve completeness verification and auditor reconstruction.

MEA-1: Deterministic Sampling Infrastructure

Requirement: All sampling for AI system monitoring SHALL use deterministic, cryptographically verifiable selection — ensuring that the operator cannot selectively monitor favorable interactions. MEA-1 specifies the key derivation and sampling infrastructure that feeds the S3P measurement method (MEA-2).

ID	Control	Attestation Artifact	Level
MEA-1.1	Derive per-policy sampling keys using a key derivation function scoped by policy identifier, as specified in the registered Protocol Profile	Key derivation documented; key fingerprint published	AAL-4
MEA-1.2	Compute pseudorandom function (PRF) tag for each request using a keyed function with domain separation including policy_id and the request commitment produced per ATT-1.2 (the keyed value, not the raw content digest). This ensures an auditor can verify sampling fairness using only the sampling key and published commitments, without requiring access to a key capable of reversing content. Specific PRF construction is defined in the registered Protocol Profile	PRF tags included in attestation envelopes	AAL-4
MEA-1.3	Determine sample membership by comparing PRF tag against threshold: sampled iff the tag value falls within the configured sampling rate boundary	Deterministic threshold computation	AAL-4
MEA-1.4	Publish per-epoch Digest Publication Ledger (DPL) enabling auditors to verify sample completeness via the S3P epoch nonce reveal (MEA-2.5)	DPL with notary signature	AAL-4

MEA-2: Statistical Safety Signal Protocol (S3P)

Requirement: Safety monitoring SHALL produce quantified statistical statements with exact confidence intervals, derived from cryptographically unbiased sampling. S3P is the single normative auditor-reproducible measurement method defined by this standard.

ID	Control	Attestation Artifact	Level
MEA-2.1	Generate secret epoch nonce via CSPRNG; withhold during epoch to prevent gaming	Epoch nonce commitment (cryptographic hash of epoch_nonce) published at epoch start	AAL-4
MEA-2.2	Compute S3P sampling tag using a keyed function with epoch_nonce and the request commitment (per ATT-1.2) as specified in the registered Protocol Profile; sample iff tag falls within the configured sampling boundary	S3P tag computation	AAL-4
MEA-2.3	Conduct full guardrail evaluation on sampled requests; record n_total, n_sampled, n_violations per policy per epoch	Per-epoch S3P attestation	AAL-4
MEA-2.4	Compute exact binomial confidence intervals using conservative methods free of normal-approximation assumptions (e.g., the Clopper-Pearson method, exact under the S3P Bernoulli sampling model with binary evaluator verdicts). The bound form (`one-sided-upper` for upper-bound safety claims; `two-sided` for interval reporting) SHALL be declared in the attestation (Section 19.3, Annex B.8). Specific formulas and minimum credibility thresholds are defined in the registered Protocol Profile	CI bounds in S3P attestation	AAL-4
MEA-2.5	Publish epoch nonce with notary signature after epoch close to enable auditor reconstruction of all sampling decisions	Published nonce matching commitment	AAL-4
MEA-2.6	Emit S3P attestation with closed schema as defined in the registered Protocol Profile, including at minimum: epoch, violation_type, n_total, n_sampled, sampling_rate, n_violations, observed_rate, confidence_level, bound_form, CI_lower, CI_upper, denominator_class, sampling_threshold, epoch_nonce_commitment, status, and signature	Notary-signed S3P attestation	AAL-4

MEA-3: Third-Party Testing

Requirement: AI systems SHALL undergo independent third-party testing at regular intervals across all risk taxonomy categories.

ID	Control	Attestation Artifact	Level
MEA-3.1	Conduct third-party adversarial robustness testing at intervals defined in the operator's risk management policy, not to exceed 12 months or as defined by applicable regulation	Third-party evaluation report	AAL-2
MEA-3.2	Conduct third-party safety testing (harmful outputs, out-of-scope, hallucination, bias) at intervals defined in the operator's risk management policy, not to exceed 12 months or as defined by applicable regulation	Third-party evaluation report	AAL-2
MEA-3.3	Conduct third-party tool-call security testing at intervals defined in the operator's risk management policy, not to exceed 12 months or as defined by applicable regulation (agentic systems only)	Third-party evaluation report	AAL-2
MEA-3.4	Publish testing scope, methodology, and results summary to transparency log (redacting sensitive details)	Transparency log entry	AAL-4

MEA-4: Pre-Deployment Testing

Requirement: AI systems SHALL undergo internal testing prior to deployment and prior to any material change.

ID	Control	Attestation Artifact	Level
MEA-4.1	Conduct pre-deployment testing covering: adversarial robustness, safety (all risk taxonomy categories), hallucination, and tool-call authorization (for agentic systems)	Test results with pass/fail criteria	AAL-2
MEA-4.2	Define material change threshold (e.g., +/-10% on evaluation metrics) requiring re-testing and re-approval	Change threshold definition in policy	AAL-2
MEA-4.3	Document test results and approval sign-offs before deployment proceeds	Approval records	AAL-2

10. Domain 6: RESPOND — Adaptive Control and Incident Response

Scope: Bounded, cryptographically gated response to detected violations. Maps to NIST AI RMF MANAGE and ISO 42001 Clauses 8–10.

RES-1: Cryptographically Gated Control Loop

Requirement: When the attestation system detects violations exceeding policy thresholds, adaptive control actions SHALL be cryptographically gated to prevent unauthorized or unbounded modifications.

ID	Control	Attestation Artifact	Level
RES-1.1	Aggregate verified receipts and NETATTs per epoch to compute violation metrics	Epoch metrics bundle	AAL-4
RES-1.2	Upon threshold exceedance, emit signed ControlAction specifying parameter changes (sampling_prob, queue_max, rate_limit)	ControlAction attestation	AAL-4
RES-1.3	Validate ControlAction through five cryptographic gates before application: (1) signature verification, (2) epoch currency, (3) parameter bounds, (4) co-epoch receipt for metrics, (5) co-epoch NETATT	Five-gate validation receipt	AAL-4
RES-1.4	Enforce parameter bounds: p_min <= sampling_prob <= p_max; 0 <= queue_max <= q_max; 0 <= rate_limit <= r_max. Reject ControlActions exceeding bounds regardless of signature validity	Bounded parameter attestation	AAL-4

The normative minimum ControlAction content is: the control identifier, the parameter delta (`params_before` → `params_after`, covering at minimum `sampling_prob`, `queue_max`, and `rate_limit` where applicable), epoch binding, and signature. The complete field-level schema, types, and encodings are profile-defined; Annex G.4 is an informative reference to that schema, and the glossary fields of A.11 (action type, timestamp, scope) are carried by the parameter delta, the epoch binding, and the referenced co-epoch receipt as described there.

RES-2: Incident Response

Requirement: The organization SHALL maintain and exercise AI incident response plans with attested verifiable record collection.

ID	Control	Attestation Artifact	Level
RES-2.1	Document AI failure plans for: security breaches, harmful outputs, hallucinations causing financial loss, and tool-call authorization failures	Incident response plans	AAL-1
RES-2.2	Assign accountable owner for each incident type with documented escalation criteria	Accountability matrix	AAL-2
RES-2.3	Upon incident detection, generate attestation pack: all attestation receipts, NETATT states, S3P signals, and ControlActions for the affected time period	Attestation pack with transparency log proofs	AAL-4
RES-2.4	Report critical incidents to designated authorities within required timeframes with cryptographic attestation artifacts	Incident report with attached receipts	AAL-4

RES-3: Emergency Override ("Break Glass")

Requirement: Emergency overrides SHALL be cryptographically attested, not hidden.

ID	Control	Attestation Artifact	Level
RES-3.1	Implement emergency override requiring enhanced authentication meeting AAL-4 identity binding requirements + reason code	Override authentication attestation	AAL-4
RES-3.2	Generate override receipt with full attestation (action taken, reason code, identity, timestamp)	Override receipt in transparency log	AAL-4
RES-3.3	Automatically schedule compliance review within SLA defined in operator's policy	Review scheduling attestation	AAL-4
RES-3.4	Surface all override events in audit dashboards and risk signal feeds	Override frequency in risk signals	AAL-4

RES-4: Scoped Revocation and Circuit Breaking

Requirement: The system SHALL support scoped, time-bounded revocation or equivalent circuit breaking of specific binaries, policies, or agent identities within a tenant boundary. Revocation SHALL be designed as a circuit breaker — local, bounded, self-healing — not as a centralized kill switch.

Security principle: The revocation mechanism SHALL NOT create a centralized control plane capable of network-wide propagation. No single entity — including the attestation platform, any notary node, or any operator — SHALL be able to unilaterally trigger revocation, and revocation SHALL NOT cross tenant boundaries. Every revocation is tenant-scoped, gated on multi-party

agreement per RES-4.2 — t-of-n notary agreement where multiple notaries are deployed; joint operator–IAP authorization in single-IAP topologies — time-bounded, and rate-limited.

ID	Control	Attestation Artifact	Level
RES-4.1	Implement tenant-scoped revocation: operators SHALL support publication of a signed Revocation Receipt revoking a specific binary_hash, policy_id, or agent identity within their own tenant boundary only. Revocation signals SHALL NOT propagate across tenant boundaries	Revocation Receipt with tenant-scoped multi-party authorization signature (per RES-4.2's topology rule)	AAL-4
RES-4.2	Gate revocation on multi-party agreement. In multi-notary topologies, Revocation Receipts SHALL require the same t-of-n notary verification as full attestation; no single notary, operator, or platform entity can unilaterally trigger revocation. In single-IAP topologies (Section 4.1.1), where a t-of-n threshold cannot exist, Revocation Receipts SHALL require joint authorization by the operator and the IAP, preserving the no-unilateral-party property; the topology and its revocation authorization model SHALL be stated in the attestation-topology disclosure (Section 22.4, Section 22.10)	Multi-party gated revocation verification	AAL-4
RES-4.3	Time-bound all revocations: Revocation Receipts SHALL include an expiration (current epoch + configurable TTL, maximum: 24 hours). Expired revocations automatically reset — circuit breaker model. Permanent decommissioning requires explicit policy republication, not perpetual revocation	Time-bounded revocation with automatic reset	AAL-4
RES-4.4	Rate-limit revocation signals: maximum one revocation per policy_id per epoch. The notary network SHALL reject revocation attempts exceeding rate limits, preventing denial-of-service via revocation spam	Rate-limited revocation enforcement	AAL-4
RES-4.5	Test revocation mechanism at intervals defined in the operator's risk management policy, not to exceed 12 months or as defined by applicable regulation; attest test	Revocation test receipt with scope verification	AAL-4

ID	Control	Attestation Artifact	Level
	execution, propagation latency, automatic reset behavior, and scope containment (verify no cross-tenant effect)		

RES-5: Failure Mode Declaration

Requirement: Operators SHALL declare and attest their system's default behavior when the attestation infrastructure itself becomes unavailable. The standard mandates the decision be explicit and attested, not a particular choice.

ID	Control	Attestation Artifact	Level
RES-5.1	Declare failure mode for attestation infrastructure unavailability: **fail-open** (AI system continues operating unattested) or **fail-closed** (AI system halts until attestation resumes). Publish declaration to transparency log	Failure mode declaration with transparency log proof	AAL-4
RES-5.2	For fail-open declarations: log all unattested operations locally; generate retroactive attestation receipts when the notary network resumes; report unattested duration as an explicit exposure window in risk signals. Retroactive receipts generated after fail-open periods SHALL carry the `RECONSTRUCTED` temporality flag (Section 21.5, bit 1) and SHALL NOT be counted as contemporaneous attestation coverage for conformance, risk-signal reporting, or litigation reporting purposes. Reconstructed receipts are reconstruction artifacts, not contemporaneous attestation; they are submitted under the late-submission semantics of Section 18.5 and Section 21.5	Exposure window accounting with RECONSTRUCTED classification	AAL-4
RES-5.3	For fail-closed declarations: implement graceful degradation (queue requests, display maintenance notification, route to human fallback) rather than silent failure	Fail-closed behavior documentation and test records	AAL-2
RES-5.4	Require review of failure mode declaration at intervals defined in the operator's risk	Failure mode review attestation	AAL-4

ID	Control	Attestation Artifact	Level
	management policy with sign-off from designated risk officer		

Note: For healthcare deployments, the fail-open vs. fail-closed decision is clinically material. A fail-closed AI system in an emergency department may cause harm through unavailability. A fail-open system may cause harm through unmonitored operation. OVERT does not prescribe the answer — it requires the decision to be documented, attested, and priced. [See Annex C: Design Rationale for healthcare deployment considerations.]

Note — containment during attestation outage: Local containment does not depend on notary availability. Arbiter circuit breakers (TOOL-3.3), loop termination (TOOL-3.4), emergency override (RES-3), security-critical anomaly containment (Section 4.7), and the fail-closed mode itself all execute locally. What requires the notary is attested revocation (RES-4) — the cross-binary, cross-policy, or cross-agent circuit-breaker receipts. During an attestation outage, operators retain local containment; revocation-class actions resume when attestation resumes, and the outage interval is reported as an exposure window (RES-5.2).

PART 3: AGENTIC AI CONTROLS

Part 3 defines the AI-specific execution controls required when systems invoke tools, coordinate with other agents, operate under delegated capability grants, route decisions through human approval paths, or exhibit behavioral drift. These sections provide AI-layer execution control, inter-agent boundary enforcement, capability mediation, privileged action authorization, transparency to relying parties, and behavioral anomaly monitoring for agentic workflows. Per Design Principle 6 (Security by Observation), the same inline enforcement position and tamper-evident recording that produce governance evidence also produce the detection, containment, and forensic reconstruction capabilities that security operations require. Where a control is satisfied at AAL-2 or AAL-3, the resulting claim is documentation- or operator-telemetry-grade evidence rather than cryptographically independent proof.

These controls apply to AI systems where autonomous agents execute tool calls, access external resources, and make decisions without step-by-step human oversight. They are mandatory for any system classified as "Automation" capability under IDE-1.2.

11. Tool-Call Governance

TOOL-1: Pre-Execution Policy Enforcement

Requirement: Every tool call by an AI agent SHALL be evaluated against policy and attested before execution. No tool call SHALL execute without a governance decision.

ID	Control	Attestation Artifact	Level
TOOL-1.1	Intercept all tool calls at the enforcement boundary before execution reaches the external resource	Per-call attestation receipt	AAL-4
TOOL-1.2	Evaluate tool calls against a capability policy specifying: permitted tools, permitted parameter ranges, permitted destinations, and required approval gates	Policy evaluation result in receipt	AAL-4
TOOL-1.3	Block tool calls that violate policy; generate denial receipt with policy reference and violation type	Denial receipt	AAL-4

ID	Control	Attestation Artifact	Level
TOOL-1.4	For permitted calls, generate provisional receipt before execution; upgrade to full attestation after notary validation	Three-phase receipt per Section 8	AAL-4

Architectural reference: Tool calls SHOULD be validated against information flow policies that consider the provenance and capabilities of all arguments, not just the tool name. Where the system tracks data provenance (source and allowed readers), policy checks SHOULD verify that argument capabilities permit the intended data flow.

TOOL-2: Function Authorization and Parameter Validation

Requirement: AI agents SHALL be restricted to approved functions with validated parameters.

ID	Control	Attestation Artifact	Level
TOOL-2.1	Maintain an explicit function allowlist: only approved tool functions may be invoked	Allowlist hash in policy attestation	AAL-4
TOOL-2.2	Validate function parameters against defined schemas before execution (type checking, range checking, format validation)	Parameter validation result in receipt	AAL-4
TOOL-2.3	Reject function calls with parameters outside defined bounds	Rejection receipt with parameter violation detail	AAL-4

TOOL-3: Tool-Call Rate Limiting and Circuit Breaking

Requirement: AI agent tool calls SHALL be subject to rate limits, velocity caps, and circuit breakers with attested enforcement.

ID	Control	Attestation Artifact	Level
TOOL-3.1	Enforce per-tool rate limits (calls per epoch, calls per minute)	Rate limit enforcement receipts	AAL-4
TOOL-3.2	Enforce per-session and per-user velocity caps for cumulative tool actions	Velocity enforcement receipts	AAL-4
TOOL-3.3	Implement circuit breakers: halt tool execution when error rates or violation rates exceed defined thresholds within an epoch	Circuit breaker activation receipt	AAL-4
TOOL-3.4	Track tool-call recursion depth per trace_id; terminate agent execution when depth exceeds a configurable threshold de-	Loop termination receipt with trace_id, depth, and termination reason	AAL-4

ID	Control	Attestation Artifact	Level
	fined in deployment policy. Agents caught in retry loops (call Tool A -> error -> call Tool A) are a common failure mode. The Arbiter SHALL detect repeated identical tool calls within a trace and terminate after configurable repetition limit		

TOOL-4: Human Approval Gates

Requirement: Sensitive tool operations SHALL require explicit human approval with attested identity binding.

ID	Control	Attestation Artifact	Level
TOOL-4.1	Define which tool operations require human-in-the-loop approval (financial transactions, data deletion, external communications, privilege modifications)	Approval-required policy in attestation	AAL-4
TOOL-4.2	Gate execution pending human approval; attest approval with authenticated identity, timestamp, and action reference	Approval receipt with identity binding	AAL-4
TOOL-4.3	Implement timeout for pending approvals; attest timeout as denial if approval not received	Timeout receipt	AAL-4
TOOL-4.4	Enforce maximum approval velocity for human reviewers (configurable approvals-per-minute threshold). Approvals exceeding the velocity cap SHALL be attested as potentially fatigued and flagged for secondary review. This mitigates rubber-stamping under high volume	Approval velocity enforcement receipt	AAL-4

TOOL-5: Tool-Call Logging and Audit Trail

Requirement: All AI agent tool calls SHALL be logged with sufficient detail for retrospective analysis.

ID	Control	Attestation Artifact	Level
TOOL-5.1	Log every tool call: tool name, parameters, caller identity, timestamp, epoch, policy evaluation result, and execution outcome	Tool-call log entries	AAL-3

ID	Control	Attestation Artifact	Level
TOOL-5.2	Ensure tool-call logs are tamper-evident: write-once storage, cryptographic hashing of entries, sequence integrity enabling gap detection	Tamper-evident log with hash chain	AAL-4
TOOL-5.3	Attest tool-call logs at each epoch boundary with notary signature over log digest	Epoch log attestation	AAL-4

11.5 MCP Server Trust Governance

The Model Context Protocol (MCP) enables AI agents to invoke tools hosted on local or remote servers. Because MCP servers mediate between the agent and external resources — databases, APIs, file systems, credentials — the trust posture of the MCP server is itself a first-class governance surface. An agent's tool-call attestation is only as strong as the trust chain to the server executing the call.

This subsection defines evidence requirements for three MCP deployment patterns: managed (vendor-hosted), custom (operator-hosted), and external (third-party-hosted). Implementations that do not use MCP or equivalent tool-hosting protocols MAY omit this subsection; the omission SHALL be declared in the conformance statement Exclusions field.

MCP-1: MANAGED MCP SERVER POSTURE EVIDENCE

Requirement: When an agentic system invokes tools through a managed (vendor-hosted) MCP server, the conformant implementation SHALL attest the server's governance posture at each co-epoch boundary.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
MCP-1.1	Record the managed MCP server identity (vendor, server version, configuration hash) in the co-epoch binding at each attestation epoch	Server identity fields in co-epoch record	AAL-4
MCP-1.2	Attest the transport security state between the arbiter and the managed MCP server (TLS version, certificate identity, mutual authentication status) at each epoch	Transport attestation in NETATT extension	AAL-4
MCP-1.3	Verify and attest the managed server's published governance metadata — including data-handling commitments, geographic jurisdiction, and sub-processor disclosures	Governance metadata receipt in transparency log	AAL-3

ID	Control	Attestation Artifact	Level
	— at deployment time and upon detected change		
MCP-1.4	Attest per-call routing: for each tool call routed to a managed MCP server, the receipt SHALL identify the server instance that executed the call	Server instance identifier in per-call receipt	AAL-4

> NOTE. *MCP-1.3 is AAL-3 rather than AAL-4 because the governance metadata originates from the vendor's own disclosures. OVERT can attest that the metadata was retrieved, verified against a published schema, and hash-committed, but cannot independently verify the vendor's operational claims. Relying parties should treat MCP-1.3 evidence as vendor-asserted, hash-sealed metadata — not as independently verified operational posture.*

MCP-2: CUSTOM MCP SERVER RUNTIME ATTESTATION

Requirement: When an agentic system invokes tools through a custom (operator-hosted) MCP server, the conformant implementation SHALL attest the server's runtime identity, network isolation, and per-call authorization posture.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
MCP-2.1	Include the custom MCP server binary identity (binary hash, configuration hash) in the co-epoch binding. Binary identity verification SHALL use the same mechanism as arbiter binary identity (ATT-2.2)	Server binary identity in co-epoch record	AAL-4
MCP-2.2	Attest that the custom MCP server operates within the same network isolation boundary as the arbiter, or attest the cross-boundary transport security state if it does not	Network topology attestation in NETATT	AAL-4
MCP-2.3	Enforce per-call authorization at the MCP server boundary: each tool invocation SHALL be evaluated against the deployment policy before execution, with the authorization decision attested in the per-call receipt	Authorization decision in per-call receipt	AAL-4
MCP-2.4	Detect and attest configuration changes to the custom MCP server within an epoch. Unau-	Configuration change detection receipt	AAL-4

ID	Control	Attestation Artifact	Level
	thorized configuration changes SHALL generate governance alerts with the same quality as topology change detection (MULTI-2.2)		

> NOTE. *MCP-2 applies operator-grade attestation to custom MCP servers because the operator controls the server lifecycle. This is stronger than MCP-1 (managed servers) because the operator can provide runtime identity evidence that a third-party vendor cannot. Implementations that co-locate the MCP server and arbiter in the same attested process may satisfy MCP-2.1 and MCP-2.2 implicitly through the arbiter's own co-epoch binding.*

MCP-3: EXTERNAL MCP CONNECTION ASSURANCE

Requirement: When an agentic system connects to an external (third-party-hosted) MCP server, the conformant implementation SHALL attest the connection governance posture and enforce scope constraints on the external server's capabilities.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
MCP-3.1	Maintain an explicit external MCP server allowlist in the deployment policy. Connections to servers not on the allowlist SHALL be denied with attested denial receipts	Allowlist hash in policy attestation; denial receipt for unauthorized connections	AAL-4
MCP-3.2	Attest the external server's identity (endpoint URI, TLS certificate fingerprint, mutual authentication status) at each connection establishment and at each co-epoch boundary	External server identity in connection receipt	AAL-4
MCP-3.3	Enforce capability scoping for external MCP servers: the set of tools and parameters available through an external server SHALL be constrained to a declared subset of the server's advertised capabilities	Capability scope restriction in per-call receipt	AAL-4
MCP-3.4	Apply output filtering (PRO-4) to all responses from external MCP servers before the response enters the agent context. The filtering decision SHALL be attested in the per-call receipt	External response filtering receipt	AAL-4

ID	Control	Attestation Artifact	Level
MCP-3.5	Record external MCP server connection lifecycle events (connect, disconnect, error, timeout) in the tamper-evident audit trail (TOOL-5) with the same attestation quality as tool-call events	Connection lifecycle entries in audit trail	AAL-4

> NOTE. *MCP-3 treats external MCP servers as untrusted by default. The allowlist (MCP-3.1) plus capability scoping (MCP-3.3) plus output filtering (MCP-3.4) create a defense-in-depth posture. Even if the external server is compromised, the attested scope constraints and filtering limit blast radius. MCP-3 does not and cannot attest the external server's internal security posture — that remains outside OVERT scope. What MCP-3 does attest is the* connection governance *applied at the operator's boundary.*

12. Multi-Agent System Controls

MULTI-1: Inter-Agent Trust Boundaries

Requirement: In multi-agent systems, trust boundaries between agents SHALL be enforced and attested. Agents SHALL NOT inherit the trust level of peer agents.

ID	Control	Attestation Artifact	Level
MULTI-1.1	Enforce distinct policy evaluation for each agent in a multi-agent system; peer agent requests SHALL be evaluated against the same policy as external requests	Per-agent attestation receipts	AAL-4
MULTI-1.2	Attest the agent identity (binary hash, configuration) for each agent in the system independently	Per-agent co-epoch binding	AAL-4
MULTI-1.3	Monitor for inter-agent trust exploitation patterns (agents relaying requests to bypass restrictions). [See Annex C: Design Rationale for research basis on multi-agent trust exploitation vulnerabilities]	Anomaly detection attestation	AAL-4

MULTI-2: Agent Composition Attestation

Requirement: The composition and configuration of multi-agent systems SHALL be attested.

ID	Control	Attestation Artifact	Level
MULTI-2.1	Document and attest the agent topology: which agents exist, their roles, their communication paths, and their capability scopes	Agent topology attestation	AAL-4
MULTI-2.2	Detect and attest changes to agent topology within an epoch	Topology change detection	AAL-4

13. Capability-Based Access Control

Architectural reference: This section adapts capability-based access control principles for the attestation layer.

CAP-1: Data Provenance Tracking

Requirement: AI systems processing sensitive data SHALL track the provenance of values flowing through tool calls and enforce access policies based on provenance metadata.

ID	Control	Attestation Artifact	Level
CAP-1.1	Tag data values with provenance metadata indicating source (user, tool, AI-generated)	Provenance tracking in system design	AAL-3
CAP-1.2	Propagate provenance through transformations: if value C derives from values A and B, C inherits the provenance of both	Provenance propagation logic	AAL-3
CAP-1.3	Enforce policies based on provenance: e.g., data from untrusted sources SHALL NOT flow to sensitive tools without explicit authorization	Provenance-based policy enforcement receipts	AAL-4

CAP-2: Architectural Separation

Requirement: AI systems making autonomous decisions SHALL architecturally separate trusted planning from untrusted data processing.

ID	Control	Attestation Artifact	Level
CAP-2.1	Planning components (which determine what actions to take) SHALL NOT directly process untrusted external data except through an attested mediation layer declared in deployment policy	Architectural documentation and validation; for Level 3 Agentic: machine-generated enforcement telemetry demonstrating mediation layer interposition; for Level 4 Agentic: independently verifiable evidence of mediation layer interposition as defined in the registered Protocol Profile	AAL-2; AAL-3 for Level 3 Agentic; AAL-4 for Level 4 Agentic
CAP-2.2	Data processing components handling untrusted input SHALL NOT have direct tool-calling capabilities	Capability restriction documentation; for Level 3 Agentic: machine-generated telemetry demonstrating capability restriction enforcement; for Level 4 Agentic: independently verifiable evidence of capability restriction as defined in the registered Protocol Profile	AAL-2; AAL-3 for Level 3 Agentic; AAL-4 for Level 4 Agentic
CAP-2.3	Data flowing from untrusted processing to trusted planning SHALL pass through structured schema validation that constrains the output space	Schema validation implementation	AAL-3

> NOTE. *At Level 1 and Level 2, CAP-2.1 and CAP-2.2 are AAL-2 documentation and process controls; conformance claims based on CAP-2 at those levels reflect documentation-grade evidence. At Level 3 Agentic, CAP-2.1 and CAP-2.2 require AAL-3 (machine-generated enforcement telemetry demonstrating that the architectural separation is actively enforced, not merely documented). At Level 4 Agentic, CAP-2.1 and CAP-2.2 require AAL-4 (independently verifiable evidence of architectural separation, as defined in the registered Protocol Profile — for example, hardware-attested process isolation, independently observed network segmentation, or equivalent mechanisms that do not rely solely on operator-controlled telemetry). This progressive elevation reflects the principle that evidence-grade claims about architectural separation require evidence-grade proof, not operator-controlled telemetry.*

14. Agent Disclosure and Transparency

DISC-1: Agent Transparency Documentation

Requirement: Organizations deploying agentic AI systems SHALL publish transparency documentation describing agent capabilities, constraints, and attestation status.

ID	Control	Attestation Artifact	Level
DISC-1.1	Publish agent capability documentation: which tools are available, what actions the agent can take, what constraints are enforced	Agent capability document	AAL-1
DISC-1.2	Publish AI Bill of Materials (CycloneDX AI BOM or SPDX 3.0) documenting model, components, and dependencies	AIBOM in machine-readable format	AAL-2
DISC-1.3	Publish attestation summary: coverage ratio, S3P safety signals, override frequency, and gap accounting — all derived from the attestation stream with no content exposure	Attestation summary in OSCAL format	AAL-4

15. Human-in-the-Loop Attestation

Human-in-the-loop interactions within AI workflows SHALL receive the same attestation quality as automated enforcement decisions. [See Annex C: Design Rationale for analysis of the verification gap in human-AI governance interactions.]

HITL-1: Consent Attestation

Requirement: When an AI system requires human consent before interaction (recording, data processing, autonomous actions affecting the individual), the consent event SHALL be attested at AAL-4 with identity binding, timestamp, and scope.

> HUMAN IDENTITY IN RECEIPTS. *Throughout the HITL controls (and TOOL-4.2), the authenticated identity of a consenting, reviewing, or correcting party SHALL be bound into the receipt as a keyed commitment resolvable by the operator, never as plaintext. The transparency log provides public*

> *verifiability of* that *a HITL event occurred and was bound to an identity, without disclosing* whose*; the operator resolves the commitment to a natural identity only under appropriate authority.*

ID	Control	Attestation Artifact	Level
HITL-1.1	Define which AI interactions require prior human consent (recording, PHI processing, autonomous actions affecting the individual) and publish consent-required policy to transparency log	Consent-required policy in attestation configuration	AAL-4
HITL-1.2	Attest consent event with: authenticated identity of consenting party, timestamp, scope of consent (what was consented to), and method of consent (verbal, written, digital signature)	Consent receipt with identity binding	AAL-4
HITL-1.3	Gate AI interaction on consent receipt: the system SHALL NOT proceed with consent-required interactions without a valid consent attestation	Consent gate enforcement receipt (permit/deny)	AAL-4
HITL-1.4	Attest consent withdrawal with timestamp and scope; system SHALL cease consent-gated operations upon withdrawal attestation	Withdrawal receipt with enforcement confirmation	AAL-4

[See Annex C: Design Rationale for regulatory context on consent attestation requirements.]

HITL-2: Human Review Attestation

Requirement: When AI outputs are routed for human review (escalation, quality assurance, regulatory requirement), the review event, reviewer identity, and decision SHALL be attested at AAL-4.

ID	Control	Attestation Artifact	Level
HITL-2.1	Define which AI outputs require human review before delivery or action (clinical recommendations, financial decisions, content moderation, high-severity classifications) and publish review-required policy	Review-required policy in attestation configuration	AAL-4
HITL-2.2	Attest review event with: reviewer authenticated identity, timestamp, review decision (approve / reject / modify), and reference	Review receipt with identity binding	AAL-4

ID	Control	Attestation Artifact	Level
	to the AI output under review (by digest, not content)		
HITL-2.3	Gate output delivery or action on review receipt for review-required outputs: the AI output SHALL NOT be delivered or acted upon without a valid review attestation	Review gate enforcement receipt	AAL-4
HITL-2.4	Track and attest review latency: elapsed time from flagging to review completion, per epoch	Review latency in epoch metrics	AAL-4

HITL-3: Human Correction and Override Attestation

Requirement: When a human modifies, corrects, or overrides an AI output or recommendation (non-emergency), the intervention SHALL be attested at AAL-4.

ID	Control	Attestation Artifact	Level
HITL-3.1	Attest human corrections to AI outputs with: corrector authenticated identity, timestamp, correction type (edit, rejection, substitution), and reference to original AI output (by digest)	Correction receipt with identity binding	AAL-4
HITL-3.2	Attest non-emergency human overrides of AI recommendations with: identity, timestamp, reason category, and reference to the overridden recommendation (by digest)	Override receipt (non-emergency)	AAL-4
HITL-3.3	Aggregate correction and override rates per policy per epoch; surface as a risk signal	Correction rate in epoch metrics	AAL-4

Operational note: Elevated correction rates may indicate model degradation, domain shift, policy misalignment, or reviewer disagreement with system outputs. Sustained low correction rates are not independently sufficient to establish output quality and should be interpreted together with review quality, drift, and coverage signals.

HITL-4: Policy and Configuration Approval Attestation

Requirement: Human approvals of governance policy changes and system configuration changes SHALL be attested at AAL-4 with separation of duties enforcement.

ID	Control	Attestation Artifact	Level
HITL-4.1	Attest policy change approvals with: approver authenticated identity, timestamp,	Policy approval receipt in transparency log	AAL-4

ID	Control	Attestation Artifact	Level
	policy version transition (old hash -> new hash), and change justification category		
HITL-4.2	Attest system configuration change approvals with: approver identity, timestamp, configuration delta (by hash), and approval authority reference	Configuration change approval receipt	AAL-4
HITL-4.3	Enforce and attest separation of duties: the individual requesting a policy or configuration change SHALL NOT be the sole approver; attest both requesting and approving identities	Dual-identity approval receipt	AAL-4

15.5 Session-Scoped Attestation

Many AI interactions are organized around sessions — bounded periods of engagement between humans and AI systems (patient encounters, clinical workflows, therapy sessions, advisory engagements, educational tutoring sessions). Session boundaries carry governance significance: consent may be scoped to a session, regulatory retention may be session-delimited, and aggregate session metrics are relevant to coverage and risk assessment. This section defines attestation requirements for session lifecycle events.

SESS-1: SESSION OPEN ATTESTATION

Requirement: When a session-based AI interaction begins, a session_open receipt SHALL be generated attesting the session initiation.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
SESS-1.1	Generate a session_open receipt at session initiation containing: session_id (unique identifier), participant identities (authenticated per HITL identity binding requirements), session type (classification per operator's session taxonomy), and timestamp	Session open receipt with co-epoch binding	AAL-4
SESS-1.2	Include consent references in the session_open receipt linking to applicable HITL-1 consent attestations. Where consent was obtained prior to the session (pre-session consent), the session_open receipt	Session open receipt with consent attestation linkage	AAL-4

ID	Control	Attestation Artifact	Level
	SHALL reference the consent receipt attestation_id. Where consent is obtained during the session (in-session consent), the consent receipt SHALL reference the session_id		
SESS-1.3	Publish session type taxonomy to the transparency log as a machine-readable artifact. Session types SHALL be declared in the operator's governance policy and SHALL map to applicable consent requirements, retention policies, and regulatory classifications	Session type taxonomy in transparency log	AAL-4

SESS-2: SESSION CLOSE ATTESTATION

Requirement: When a session ends, a session_close receipt SHALL be generated attesting the session conclusion and summary disposition.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
SESS-2.1	Generate a session_close receipt at session termination containing: session_id (matching the session_open receipt), disposition (completed, abandoned, transferred, error, timeout, terminated), session duration, total action count within the session (tool calls, reviews, approvals, and other attested events), and timestamp	Session close receipt with co-epoch binding	AAL-4
SESS-2.2	The session_close receipt SHALL reference the session_open receipt by attestation_id, forming a verifiable session boundary pair	Session close receipt with session_open attestation_id reference	AAL-4
SESS-2.3	For sessions ending with disposition "transferred," the session_close receipt SHALL include the identity of the receiving entity (human or system) and a reference to any successor session_open receipt if available	Transfer disposition receipt with successor reference	AAL-4

SESS-3: SESSION CONSENT BINDING

Requirement: Consent attestations (HITL-1) SHALL be linkable to session scope.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
SESS-3.1	Consent granted for a specific session type SHALL cover actions within sessions of that type. The consent scope field in HITL-1 receipts SHALL support session-type-scoped consent declarations	Consent receipt with session type scope	AAL-4
SESS-3.2	When consent is withdrawn mid-session (per HITL-1.4), the session SHALL either terminate (generating a session_close receipt with disposition "abandoned") or continue with reduced scope as defined in the operator's consent withdrawal policy. The consent withdrawal receipt SHALL reference the session_id	Consent withdrawal receipt with session_id reference	AAL-4

SESS-4: SESSION-AGGREGATE SIGNALS

Requirement: Per-session summary data SHALL be reportable in epoch metrics.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
SESS-4.1	Report session-aggregate signals per epoch including at minimum: session count, average session duration, action density (average actions per session), and consent coverage rate (percentage of sessions with valid consent attestation at session open)	Session aggregate signals in epoch metrics	AAL-4
SESS-4.2	Session-aggregate signals SHALL be classified as operational signals (Annex D, Section D.2) and SHALL satisfy the signal properties defined in Section 4.6	Session signals in risk signal framework	AAL-4

SESS-5: SESSION CONTEXT DESTRUCTION ATTESTATION

Requirement: When session context is destroyed (as required by policy, regulation, or operator data lifecycle management), the destruction event SHALL be attested.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
SESS-5.1	Generate a session context destruction receipt when session-scoped data (conversation history, intermediate results, session state) is destroyed. The receipt SHALL include: session_id, destruction timestamp, destruction reason (policy-mandated, regulatory-required, retention-expired, operator-initiated), and a cryptographic commitment to the data being destroyed (hash of the session content, not the content itself)	Session context destruction receipt	AAL-4
SESS-5.2	Session context destruction receipts SHALL be retained in the transparency log for the operator's full retention period, even after the session context itself is destroyed. The destruction receipt attests that the committed context existed and that an attested destruction event was executed for it; it does not by itself prove physical or cryptographic erasure unless the registered Protocol Profile binds a key-destruction attestation. Its absence from the log after a destruction event is a conformance deviation	Destruction receipt in transparency log with retention	AAL-4

> NOTE. *Session-scoped attestation is applicable at Level 2 and above for systems with session-based interactions. Systems that process only stateless, independent requests without session boundaries are not required to implement this section. The determination of whether a system has "session-based interactions" is made by the operator based on the system's architecture and use context.*

15.6 Agent State and Prompt Governance

Agentic AI systems that persist state across sessions (conversation memory, retrieval-augmented context, tool-call history) or operate under registered prompt artifacts (system prompts, instruction templates, chain-of-thought scaffolding) introduce governance surfaces not covered by session-scoped attestation alone. This subsection defines evidence requirements for the integrity, lineage, and governance of those surfaces.

STATE-1: DURABLE AGENT STATE SEALING

Requirement: Agentic systems that persist state across session boundaries SHALL seal and attest durable state transitions so that the provenance and integrity of reused state are independently verifiable.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
STATE-1.1	At session close, compute a cryptographic commitment (hash) over the durable state snapshot that will be available to the next session. Publish the commitment to the transparency log with session-binding metadata (session_id, epoch, agent_class)	State commitment receipt in transparency log	AAL-4
STATE-1.2	At session open, verify that the loaded durable state matches the commitment published at the prior session close. Verification failure SHALL generate a state-integrity governance alert and SHALL prevent the session from proceeding until the operator resolves the discrepancy or explicitly overrides with attested justification	State verification receipt or state-integrity alert	AAL-4
STATE-1.3	Maintain a hash-chained lineage of state transitions: each state commitment SHALL reference the prior state commitment hash, enabling DAG reconstruction of the full state history for a given agent or agent class	State lineage chain in transparency log	AAL-4
STATE-1.4	Attest state mutation provenance: for each mutation to durable state within a session (memory write, context update, retrieval injection), record the source (user input, tool output, AI-generated, system-injected) and the policy evaluation result that authorized the mutation	State mutation provenance in per-action receipt	AAL-4
STATE-1.5	Enforce state access scoping: durable state SHALL be retrievable only by agent classes and sessions authorized by the deployment policy. Unauthorized state access attempts SHALL be denied and attested	State access authorization receipt or denial receipt	AAL-4

> NOTE. *STATE-1 does not prescribe the storage mechanism for durable state. It prescribes what must be attested about state transitions. Implementations may use vector stores, relational databases, key-value stores, or file systems — the attestation requirements are storage-agnostic. The hash-chained lineage (STATE-1.3) enables an auditor to reconstruct which state version was available to which session without accessing the state content itself.*

STATE-2: PROMPT ARTIFACT REGISTRATION AND BINDING

Requirement: Organizations deploying agentic AI systems SHALL register prompt artifacts in a governance-controlled registry and bind each agent execution to the specific prompt artifact version that governed it.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
STATE-2.1	Register all prompt artifacts (system prompts, instruction templates, chain-of-thought scaffolds, few-shot exemplars) in a versioned, hash-committed registry published to the transparency log	Prompt artifact registration receipt with content hash and version	AAL-4
STATE-2.2	At session initialization, bind the active prompt artifact version to the session attestation. The prompt artifact hash SHALL appear in the session_open receipt (§15.5)	Prompt artifact hash in session_open receipt	AAL-4
STATE-2.3	Detect and attest prompt artifact changes within a session. Mid-session prompt modification SHALL generate a governance alert and a new prompt-binding receipt	Prompt change detection receipt	AAL-4
STATE-2.4	Enforce prompt-to-action traceability: for each attested action (tool call, output generation, escalation), the receipt SHALL reference the prompt artifact version that was active when the action was authorized	Prompt artifact reference in per-action receipt	AAL-4
STATE-2.5	Require that prompt artifact registration and version changes be approved by a Qualified Risk Officer (per GOV-3.5) or equivalent governance authority declared in the deployment policy. Approval SHALL be attested with identity binding	Prompt change approval receipt with identity binding	AAL-4

> NOTE. *STATE-2 does not require that prompt content be disclosed in receipts or the transparency log — only the hash and version. This preserves the non-egress property: a verifier can confirm that a specific prompt version governed an execution without accessing the prompt text. Organizations that choose to disclose prompt content may do so; the standard does not require it.*

15.7 Delegated Identity Chain Attestation

In federated deployments, the principal authorizing an agent action may not be the directly authenticated user. The action may be authorized through a chain of delegated identities: a user authenticates to an IdP, the IdP issues a token, the token is exchanged for a scoped credential, the credential is used by an orchestrator that delegates to a sub-agent. Each link in that chain is a trust decision. Conformant implementations SHALL attest the full delegation chain so that relying parties can verify who authorized what, through which intermediaries, under which constraints.

IDENT-1: FEDERATED IDENTITY AND TOKEN PROVENANCE

Requirement: Agentic systems operating under federated or delegated identity SHALL attest the full identity delegation chain from the originating principal to the executing agent.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
IDENT-1.1	Record the originating principal identity (user, service account, or workload identity) in the attestation receipt for every governed action. The identity SHALL include the identity provider, the authentication method, and the authentication timestamp	Originating principal identity in per-action receipt	AAL-4
IDENT-1.2	Record the delegation chain: for each token exchange, credential delegation, or authority transfer between the originating principal and the executing agent, attest the delegating entity, the receiving entity, the scope constraints applied at delegation, and the delegation timestamp	Delegation chain in per-action receipt	AAL-4
IDENT-1.3	Verify scope narrowing at each delegation step: each delegation SHALL narrow or preserve (never widen) the capability scope of the prior step. Scope widening	Scope verification at each delegation step	AAL-4

ID	Control	Attestation Artifact	Level
	SHALL generate a governance alert and a denial receipt		
IDENT-1.4	Attest token lifetime and revocation status: for each token or credential in the delegation chain, record the issued-at time, expiration time, and (where available) revocation-check result at the time of action authorization	Token lifecycle metadata in per-action receipt	AAL-4
IDENT-1.5	For multi-agent delegation (agent A delegates to agent B), bind the delegating agent's attestation ID (parent_attestation_id per DRIFT-1.5) to the delegation chain, enabling unified identity-and-execution DAG reconstruction	Agent delegation linkage in per-action receipt	AAL-4

> NOTE. *IDENT-1 does not prescribe the identity provider, token format, or federation protocol. It prescribes what must be attested about the delegation chain. Implementations using OIDC, SAML, SPIFFE, or proprietary federation protocols all satisfy IDENT-1 provided they produce the required attestation artifacts. IDENT-1.3 (scope narrowing) is the critical security property: it ensures that delegation cannot silently escalate privileges.*

16. Behavioral Drift Governance

These controls address emergent behavioral changes in agentic AI systems that occur within authorized operational bounds — situations where every individual control passes but the system's aggregate behavior drifts, cascades, or produces ungovernable complexity. Behavioral drift governance is distinct from policy violation detection (covered by PROTECT and MEASURE domains): policy violation detection identifies individual actions that breach a rule, while behavioral drift governance detects statistically significant changes in authorized behavior patterns that may indicate systemic risk.

These controls are mandatory for any system classified as "Automation" capability under IDE-1.2 that deploys two or more interacting agents or any single agent with tool-calling capabilities.

DRIFT-1: Baseline Intent Declaration

Requirement: Agentic AI systems SHALL publish and maintain a baseline intent declaration specifying the permitted agent topology, behavioral bounds per agent class, permitted spawn relationships, model bindings, and human oversight requirements. The declaration SHALL be versioned, hash-chained, and published to the transparency log.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
DRIFT-1.1	Publish baseline intent declaration in machine-readable format to transparency log with cryptographic timestamp	Baseline intent declaration receipt in transparency log	AAL-4
DRIFT-1.2	Declare behavioral bounds per agent class including: permitted output distribution characteristics, permitted tool selection distributions, permitted spawn topologies, and human oversight checkpoint requirements	Behavioral bounds specification in baseline intent declaration	AAL-4
DRIFT-1.3	Version-link baseline intent declarations in the transparency log (each version references the hash of the prior version)	Hash-chained version linkage in transparency log	AAL-4
DRIFT-1.4	Require that baseline intent declaration changes be approved by a Qualified Risk Officer (per GOV-3.5) with attested separation of duties	Dual-identity approval receipt for baseline change	AAL-4
DRIFT-1.5	Publish parent-child attestation linkage requirements: every agent action receipt SHALL reference the spawning agent's attestation ID (parent_attestation_id), enabling DAG reconstruction	Parent-child attestation linkage in per-call receipts	AAL-4

DRIFT-2: Behavioral Drift Detection

Requirement: Conformant agentic systems SHALL employ sequential statistical methods to detect behavioral drift per agent class, using evaluation instruments that produce temporally stable, version-consistent measurement features. Drift detection SHALL operate on dimensions specified in the baseline intent declaration.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
DRIFT-2.1	Implement sequential statistical analysis (method specified in registered Protocol Profile) for detecting distribution shifts in agent behavior per agent class	Drift detection configuration in baseline intent declaration	AAL-4
DRIFT-2.2	Evaluation instruments used for drift measurement SHALL demonstrate score stability across instrument versions and cross-deployment comparability. Version stability requirements are specified in the registered Protocol Profile	Evaluation instrument version attestation	AAL-4
DRIFT-2.3	Drift detection SHALL operate per-dimension (output risk, tool selection, semantic characteristics) with independent statistical tracking per dimension	Per-dimension drift statistics in epoch metrics	AAL-4
DRIFT-2.4	Attest drift detection signals with the same co-epoch binding as enforcement receipts. Drift signals SHALL include: agent class, dimension, statistical test result, confidence level, and epoch	Drift signal receipt with co-epoch binding	AAL-4
DRIFT-2.5	Implement graduated response to drift signals: log, alert, escalate, block. Each escalation level SHALL be independently attested. The escalation ladder and thresholds SHALL be declared in the baseline intent declaration	Graduated response receipt per escalation level	AAL-4
DRIFT-2.6	Support adaptive sampling escalation triggered by drift signals — sampling rate SHALL increase when drift statistics approach declared thresholds. Escalation triggers and bounds SHALL be declared in the baseline intent declaration and attested when activated	Sampling escalation receipt with trigger evidence	AAL-4

> NOTE: *The standard requires drift detection capability and specifies what must be measured and attested. The specific statistical method (CUSUM, EWMA, or other sequential analysis), feature extraction architecture, and evaluation instrument design are specified in the registered Protocol Profile.*

DRIFT-3: Graph Topology Governance

Requirement: Conformant multi-agent systems SHALL compute and attest graph complexity metrics for each agentic execution. When graph complexity exceeds thresholds declared in the baseline intent declaration, the system SHALL generate governance alerts with attested evidence.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
DRIFT-3.1	Compute graph complexity metrics per execution including at minimum: total agent count, edge count, maximum depth, and branching factor	Graph complexity metrics in execution receipt	AAL-4
DRIFT-3.2	Evaluate graph complexity against thresholds declared in the baseline intent declaration	Threshold evaluation result in execution receipt	AAL-4
DRIFT-3.3	Generate attested governance alerts when graph complexity exceeds declared thresholds, including: execution DAG summary, complexity metrics, baseline threshold, and epoch binding	Graph complexity governance alert receipt	AAL-4
DRIFT-3.4	Attest spawn authorization decisions in sub-epoch time. The mechanism for real-time spawn authorization is specified in the registered Protocol Profile. Unauthorized spawn attempts SHALL generate denial receipts with the same attestation quality as tool-call denials (TOOL-1.3)	Spawn authorization receipt or spawn denial receipt	AAL-4

> NOTE: *DRIFT-3.4 requires real-time spawn authorization but does not prescribe the enforcement mechanism. Protocol Profile implementations may use probabilistic data structures, allowlist lookups, or other mechanisms capable of meeting the latency requirement.*

DRIFT-4: Causal Drift Attribution

Requirement: In multi-agent systems, when behavioral drift is detected in a downstream agent, conformant Level 4 Agentic systems SHALL evaluate upstream agents for correlated drift using parent-child attestation linkages. Attribution findings SHALL be attested.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
DRIFT-4.1	When drift is detected in a downstream agent (per DRIFT-2), evaluate upstream agents in the attestation DAG for correlated statistical changes in the same or adjacent epochs	Upstream correlation analysis in attribution receipt	AAL-4
DRIFT-4.2	Attest attribution findings including: downstream agent class, upstream agent class, correlation evidence, attestation DAG path, and epoch range	Causal attribution receipt	AAL-4
DRIFT-4.3	When causal attribution identifies an upstream root cause, propagate the graduated response (DRIFT-2.5) to the upstream agent class	Propagated graduated response receipt	AAL-4
DRIFT-4.4	Conformant implementations SHALL employ a multi-factor attribution methodology that considers, at minimum: propagated upstream drift, local downstream drift, exogenous environmental change, combined causes, and indeterminate attribution. The attribution methodology SHALL produce attribution confidence scores quantifying the strength of evidence for each attribution factor. The specific attribution formula (e.g., PathScore) is specified in the registered Protocol Profile	Attribution methodology receipt with per-factor confidence scores	AAL-4
DRIFT-4.5	Attribution results SHALL be classified using the following taxonomy: **PROPAGATED_UPSTREAM** (drift caused by upstream agent change), **LOCAL_DOWNSTREAM** (drift caused by local agent change), **EXOGENOUS** (drift caused by external environmental change), **COMBINED** (multiple contributing factors identified), **INDETERMINATE** (insufficient evidence for classification). The classification SHALL be included in the attribution receipt. Where the classification is COMBINED, the receipt SHALL enumerate the contributing factors and their respective confidence scores. Where the classification is INDE-	Attribution classification receipt with taxonomy code and supporting evidence	AAL-4

ID	Control	Attestation Artifact	Level
	TERMINATE, the receipt SHALL state the reason (insufficient data, conflicting evidence, or ambiguous correlation)		

> NOTE: *DRIFT-4 is required for Level 4 Agentic conformance because downstream drift without upstream attribution materially limits containment and post-incident reconstruction in multi-agent systems. Simpler deployments that do not claim Level 4 Agentic conformance may still omit DRIFT-4.*

DRIFT-5: Human Oversight Quality Assessment

Requirement: Conformant systems SHALL track and attest human review quality indicators including review duration, modification rate, and consistency between review decisions and risk signals. Sustained degradation in review quality indicators SHALL trigger governance escalation.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
DRIFT-5.1	Track per-reviewer and per-agent-class review quality indicators: review duration (time from presentation to decision), modification rate (proportion of reviews resulting in edits, rejections, or substitutions), and risk-signal consistency (agreement between review decisions and system risk classifications)	Review quality indicators in epoch metrics	AAL-4
DRIFT-5.2	Attest review quality indicators per epoch with the same co-epoch binding as other governance signals	Review quality attestation receipt with co-epoch binding	AAL-4
DRIFT-5.3	Define review quality degradation thresholds in the baseline intent declaration. When review quality indicators degrade below declared thresholds (e.g., review duration dropping, modification rate declining while risk signals remain elevated), generate attested governance alerts	Review quality degradation alert receipt	AAL-4
DRIFT-5.4	Review quality indicators SHALL be reported as risk signals (see Annex D and	Review quality in risk signal payload	AAL-4

ID	Control	Attestation Artifact	Level
	the registered Protocol Profile for signal specifications)		

> NOTE: *DRIFT-5 strengthens existing HITL-2 (Human Review Attestation) and TOOL-4.4 (approval velocity enforcement) by adding substantive quality assessment beyond mechanical timing checks. It directly supports EU AI Act Article 14's requirement that humans "properly understand the relevant capacities and limitations" of the system they oversee.*

16.1 Evaluator Compatibility Framework

Behavioral drift detection (DRIFT-2) depends on evaluation instruments that produce structured measurement features — governance feature vectors — which are compared across time to detect distributional shifts. When evaluator versions change (new models, updated rubrics, different scoring dimensions), the resulting feature vectors may not be comparable to those produced by the prior version. Silent reuse of detector state (baselines, thresholds, statistical accumulators) across incompatible evaluator versions produces spurious drift signals or, worse, masks genuine drift. This section defines the framework for evaluator compatibility, versioning, and state management.

EVAL-1: GOVERNANCE EVALUATORS AND STRUCTURED VERDICTS

Requirement: Governance evaluators — components that produce structured verdicts and governance feature vectors within the operator trust boundary — SHALL produce outputs conforming to a closed schema with declared dimensions.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
EVAL-1.1	Evaluator outputs SHALL conform to a closed schema (no undeclared fields) with a fixed, declared set of dimensions. Each dimension SHALL have a defined name, data type, value range, and semantic description	Evaluator schema artifact in transparency log	AAL-4
EVAL-1.2	Each evaluator version SHALL publish a semantic-ordering manifest: a machine-readable declaration specifying the meaning, ordering, and interpretation of each dimension in the governance feature vector.	Semantic-ordering manifest with transparency log inclusion proof	AAL-4

ID	Control	Attestation Artifact	Level
	The manifest SHALL be versioned, hash-chained, and published to the transparency log		
EVAL-1.3	Evaluator outputs SHALL include the evaluator version identifier and semantic-ordering manifest hash in every structured verdict, enabling downstream consumers to verify which evaluator produced which verdict	Evaluator version and manifest hash in verdict payload	AAL-4

EVAL-2: COMPATIBILITY DOMAINS AND DETECTOR-STATE PARTITIONING

Requirement: Evaluator versions SHALL be organized into compatibility domains within which feature vectors are longitudinally comparable. When an evaluator version change breaks compatibility, detector state SHALL be partitioned.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
EVAL-2.1	Declare compatibility domains: a compatibility domain is a set of evaluator versions whose feature vectors are longitudinally comparable (same schema, same dimensions, same semantic ordering, compatible value ranges). The active compatibility domain SHALL be published to the transparency log	Compatibility domain declaration in transparency log	AAL-4
EVAL-2.2	When a new evaluator version breaks compatibility (different schema, different dimensions, different semantic ordering, or materially different calibration), detector state SHALL be partitioned: the system SHALL establish a new compatibility domain with a fresh baseline, fresh statistical accumulators, and fresh drift thresholds. Silent reuse of detector state across incompatible evaluator versions is non-conformant	Detector state partition receipt with old and new domain identifiers	AAL-4
EVAL-2.3	Cross-domain drift comparison SHALL NOT be performed. Drift signals from one compatibility domain SHALL NOT be com-	Domain isolation attestation in drift signal receipts	AAL-4

ID	Control	Attestation Artifact	Level
	pared to or aggregated with drift signals from a different compatibility domain. Each domain maintains independent statistical history		

EVAL-3: COMPATIBILITY ASSESSMENT WORKFLOW

Requirement: Before a candidate evaluator version is activated, the system SHALL execute a compatibility assessment comparing the candidate to the active evaluator.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
EVAL-3.1	The compatibility assessment SHALL evaluate, at minimum: (a) schema conformance — the candidate produces the same set of fields as the active evaluator; (b) dimensionality — the candidate produces the same number of dimensions with the same names; (c) semantic-ordering-manifest equality — the candidate's manifest matches the active evaluator's manifest; (d) missingness behavior — the candidate handles missing or null inputs identically to the active evaluator; (e) continuity metrics — the candidate's score distributions on a held-out calibration set are within declared continuity bounds of the active evaluator's distributions; (f) calibration stability — the candidate's score-to-outcome calibration on the held-out set is within declared bounds	Compatibility assessment receipt with per-criterion results	AAL-4
EVAL-3.2	If the compatibility assessment determines that the candidate is compatible, the candidate MAY be activated within the existing compatibility domain. If the assessment determines incompatibility on any criterion, the candidate SHALL be activated in a new compatibility domain (per EVAL-2.2)	Compatibility determination receipt (compatible / incompatible) with criterion-level detail	AAL-4
EVAL-3.3	The compatibility assessment results SHALL be published to the transparency	Pre-activation compatibility assessment in transparency log	AAL-4

ID	Control	Attestation Artifact	Level
	log before the candidate evaluator is activated in production. Activation without a published compatibility assessment is nonconformant		

EVAL-4: EVALUATOR VERSION ATTESTATION

Requirement: The active evaluator version identifier and artifact hash SHALL be attested per-epoch.

Attestation Assurance Level: AAL-4

ID	Control	Attestation Artifact	Level
EVAL-4.1	The active evaluator version identifier, artifact hash (cryptographic digest of the evaluator binary or model artifact), and compatibility domain identifier SHALL be included in the epoch summary attestation	Evaluator version binding in epoch attestation	AAL-4
EVAL-4.2	Evaluator version changes within an epoch SHALL trigger an immediate compatibility assessment (EVAL-3) and SHALL be attested as a configuration change event (per Section 18.6)	Mid-epoch evaluator change receipt	AAL-4

> NOTE. *The Evaluator Compatibility Framework extends DRIFT-2.2 (evaluation instrument version stability) into a complete lifecycle framework. DRIFT-2.2 requires that evaluation instruments demonstrate score stability across versions; this section specifies how to verify, attest, and manage that stability through structured compatibility domains, assessments, and state partitioning.*

All evaluator compatibility controls in this section are normative at AAL-4 for Level 3 and Level 4 Agentic conformance claims. Systems not claiming Agentic scope are not required to implement this section.

PART 4: ATTESTATION ARCHITECTURE REQUIREMENTS

Part 4 defines the evidence trust plane required for credible AI security and governance claims. It specifies the minimum architectural properties needed for trustworthy detection, investigation, audit, and defensible response: non-egress attestation, temporal binding to runtime state, statistically reproducible measurement, third-party auditability, and preservation of records in a form suitable for later verification. These sections do not establish that a deployment is secure. They establish the conditions under which claimed control execution and observed events can be checked.

17. Non-Egress Attestation Architecture

> PLAINLY — *This is the move that makes adoption safe: to be governed, your data does not move. The system fingerprints each request and response and signs the fingerprint, never the content; a verifier confirms what happened without seeing the prompt, the record, or the output. Privacy and proof, usually in tension, are here the same mechanism.*

Requirement: The attestation protocol SHALL NOT require transmission of protected content outside the operator environment. Conformant receipt-service interfaces SHALL accept only cryptographic commitments and profile-defined metadata.

17.1 All AI request/response payloads SHALL be canonicalized using deterministic encoding as specified in the registered Protocol Profile. Deterministic encoding means that two conformant encoders encoding the same logical data produce identical byte sequences. The canonicalization method SHALL be version-pinned and identified by a cryptographic hash of the encoder.

17.1.1 Numeric values in attestation envelopes SHALL be represented in a lossless format as specified by the registered Protocol Profile. The encoding SHALL ensure that numeric values are not subject to platform-dependent rounding or representation variance.

17.2 Request commitments crossing the operator's trust boundary SHALL be computed using a keyed cryptographic function with tenant-scoped keys held exclusively in the operator's key management system. Raw content digests SHALL NOT egress.

> INFORMATIVE NOTE: *The keyed commitment construction prevents rainbow table reversal of low-entropy content (e.g., PII, SSNs) by any party with ledger access.*

17.3 Attestation artifacts (prompts, responses, policy evaluations) SHALL be stored in content-addressable storage within the operator's environment, indexed by attestation commitment, and subject to the operator's data retention policies (see Section 21).

17.4 Attestation egress SHALL be constrained to a single receipt service endpoint over mutually authenticated TLS with certificate pinning. The receipt service schema SHALL be closed (reject unknown fields) and SHALL accept only cryptographic commitments — never content. Disclosure of protected content to an authorized verifier under Section 20.4 and Annex G.1 is a separate, exceptional channel exercised under the requesting party's lawful authority or contractual agreement (Section 20.4) within the operator's environment; it is not attestation egress and does not relax this section (Annex G.1.2(d)).

17.5 The non-egress architecture SHOULD be designed to prevent the transmission of Protected Health Information (PHI) or other regulated content, supporting minimized compliance footprints. The applicability of data processing agreements or Business Associate Agreements remains a question of applicable law and regulatory interpretation.

17.6 Implementations SHALL conform to the non-egress specifications in the deployment's declared Protocol Profile — a registered profile, or at Level 2 a validly self-declared profile under Section 22.6.3 — including envelope schemas, commitment derivations, and receipt service constraints. The requirement to comply with the declared Protocol Profile is normative. The description of Protocol Profile 1.0 in Annex B is informative — it serves as a reference implementation specification. Protocol Profile 1.0 becomes normatively binding on implementations that register compliance with it. Conformance claims SHALL identify a dated, versioned profile specification.

18. Temporal Binding and Configuration Integrity

> PLAINLY — *An epoch is a short, fixed interval — about five minutes. Each receipt is sealed to its epoch and to the exact enforcement software and network state in force at the time. Nothing can be re-dated; nothing can be re-attributed to a configuration that was not running. The clock is part of the evidence.*

Requirement: Every attestation receipt SHALL be cryptographically bound to the system's binary identity, network isolation state, and runtime configuration during a bounded time interval, enabling retroactive proof of what configuration was running at any attested moment.

18.1 The attestation system SHALL establish bounded time intervals (epochs) during which system configuration is attested as stable. Epoch duration SHALL be configurable (recommended: 300 seconds). Specific epoch constraints are defined in the registered Protocol Profile.

18.2 System binary identity SHALL be derived by the notary through a measurement pipeline that is (a) not controlled by the attester, (b) rooted in a hardware or cryptographic trust anchor, and (c) reproducible by an independent auditor given the measurement policy. Client-supplied identity claims are insufficient for AAL-4 conformance. The notary SHALL maintain an authoritative mapping from epoch to binary identity.

Mode by level. Notary-derived binary identity is required from Level 3, where the required architecture includes a notary (Section 22.1, Section 22.2). At Levels 1–2, whose required architecture does not include an independent notary, binary identity MAY instead be derived through an **operator-attested measurement pipeline** that still satisfies (b) and (c) and whose measurement source is outside the measured workload's control (e.g., cloud-provider attestation documents, hypervisor- or orchestrator-attested measurements per the note below). Temporal binding so derived is operator-attested evidence — AAL-2 under the Section 22.1 grading rule — and SHALL be disclosed as such; in this mode the operator maintains the authoritative epoch-to-binary-identity mapping.

> NOTE: *Acceptable measurement pipelines include, but are not limited to: AWS Nitro Enclave attestation documents, Intel SGX/TDX DCAP quotes, AMD SEV-SNP attestation reports, TPM 2.0 PCR-based attestation, and hypervisor-attested or orchestrator-attested measurements. Software-based measurements from hypervisors, orchestrators, or container runtimes satisfy these properties where hardware TEE attestation is unavailable, provided the measurement source is outside the measured workload's control — and, for the notary-derived mode of 18.2, outside the attester's administrative*

> *control. Reproducible builds and binary transparency logs provide supplementary software provenance evidence but do not by themselves satisfy runtime measurement requirements. The registered Protocol Profile specifies which mechanism(s) a conformant implementation uses.*

Notary signature constructions (whether threshold signatures, multi-signatures, or other schemes achieving the t-of-n trust property) SHALL be as defined in the registered Protocol Profile. After January 1, 2031, conformant implementations SHALL use hybrid classical + post-quantum constructions, or pure post-quantum constructions, as specified in the registered Protocol Profile. Pure classical signature schemes become non-conformant after that date.

> INFORMATIVE NOTE: *The t-of-n trust property — that no single notary can forge or suppress attestations — can be realized through threshold signature schemes (a single aggregated signature, e.g., BLS) or multi-signature schemes (independent per-notary signatures verified against a t-of-n policy). Threshold schemes produce compact proofs but require distributed key generation and have limited post-quantum options. Multi-signature schemes use standard per-node signing algorithms, offer straightforward post-quantum migration via FIPS 204 (ML-DSA) or FIPS 205 (SLH-DSA), and provide transparency about which notaries participated. The Protocol Profile specifies the construction; the standard requires only the trust property.*

18.3 Network isolation state SHALL be attested at each epoch. The network isolation state hash SHALL cover, at minimum: (a) the effective egress policy, (b) the identity of the enforcement component, and (c) the TLS certificate pin set. Operators MAY include additional deployment-specific inputs such as network enforcement rules, runtime environment variables affecting AI behavior, and policy controller state. The minimum input set is specified in the registered Protocol Profile.

> NOTE — SCOPE OF NETWORK ISOLATION ATTESTATION: *The inputs attested under 18.3 are* declarative policy state *— the network policies, egress rules, and enforcement configuration that govern what the AI system is permitted to reach. Ephemeral infrastructure state (e.g., pod IP assignments, container instance identifiers, rotating service credentials) is not within scope unless the operator's measurement policy (18.2) explicitly includes it. The operator defines which specific inputs compose the network isolation state hash; the standard requires that the chosen inputs are sufficient to detect policy-level changes across epoch boundaries.*

18.4 Every attestation receipt SHALL be bound to: (a) the current epoch, (b) the notary-derived binary identity, and (c) the network isolation state hash. Receipts lacking any of these bindings SHALL be rejected.

18.5 Stale-epoch submissions (referencing a past epoch) SHALL be rejected with a deterministic error code. Bounded clock skew tolerance as defined in the registered Protocol Profile (recommended: <=2 seconds) is permitted. Late submissions flagged DELAYED_NOTARY or RECONSTRUCTED (Section 21.5) reference the epoch current at submission time and carry the covered event interval as metadata; they are therefore not stale-epoch submissions, and SHALL be accepted within the profile-defined maximum late-submission window.

18.6 Configuration drift — changes to binary identity, network state, or runtime configuration — SHALL be cryptographically detectable by comparing attestation bindings across epoch boundaries.

18.7 Implementations SHALL conform to the co-epoch binding and network attestation specifications in the registered Protocol Profile.

18.8 Within-Epoch Measurement Requirements

Cross-epoch binary identity verification (18.1–18.7) detects drift between measurement points but does not preclude a just-in-time substitution attack: an adversary may present a compliant binary at epoch-boundary measurement, execute a non-compliant binary during the epoch, and restore the compliant binary before the next measurement. To close this gap, conformant implementations SHALL ensure binary identity continuity within epochs.

18.8.1 Conformant implementations SHALL satisfy at least one of the following within-epoch measurement strategies:

(a) **Continuous remeasurement.** The implementation performs binary identity verification at a regular cadence within each epoch. The minimum within-epoch measurement frequency SHALL be specified in the registered Protocol Profile.

(b) **Per-receipt liveness proofs.** Each attestation receipt (or receipt batch, where batching is profile-defined) includes a fresh binary identity measurement binding the receipt to the binary state at the time of receipt generation. The liveness proof SHALL include a timestamp and a nonce or monotonic counter to prevent replay.

(c) **Event-driven remeasurement.** The implementation triggers an immediate binary identity verification upon any detected configuration change event, process restart, library reload, container image change, or equivalent mutation to the execution environment.

Implementations that rely solely on epoch-boundary measurement without any within-epoch strategy are NOT conformant with AAL-3 or AAL-4.

18.8.2 At AAL-4, binary identity verification SHALL occur at minimum once per receipt batch (where batch size is defined by the Protocol Profile) or upon any detected configuration change event, whichever is more frequent. The measurement result SHALL be cryptographically bound to the receipt or receipt batch it covers.

18.8.3 Any gap in within-epoch measurement exceeding the profile-defined maximum interval SHALL cause the affected receipts to be marked with the status `MEASUREMENT_GAP` and SHALL be disclosed in the epoch summary.

18.8.4 Implementations MAY use hardware-rooted continuous attestation mechanisms (e.g., runtime TCB measurement via TPM, Intel TXT, or ARM TrustZone) to satisfy within-epoch measurement requirements with higher assurance. Hardware-rooted continuous attestation meeting or exceeding the Protocol Profile minimum frequency SHALL be considered sufficient without additional software-layer remeasurement.

18.8.5 The registered Protocol Profile SHALL declare the within-epoch measurement strategy, minimum measurement frequency, maximum batch size for per-receipt proofs, and the set of configuration change events that trigger event-driven remeasurement.

19. Statistical Safety Measurement

Requirement: Safety monitoring SHALL produce quantified statistical statements with exact confidence intervals, derived from cryptographically unbiased sampling that is auditor-reproducible without content access.

19.1 Sampling for AI system monitoring SHALL be deterministic and auditor-reproducible. The operator SHALL NOT be able to selectively monitor favorable interactions — an auditor SHALL be able to verify that sampling was fair and comprehensive.

19.2 Sampling decisions SHALL be cryptographically unpredictable during the observation period and verifiable after it. A secret value (nonce) SHALL drive sampling decisions during each epoch, then be published after epoch close for independent reconstruction. This standard defines a single normative auditor-reproducible sampling and measurement method: the Statistical Safety Signal Protocol (S3P). Alternative sampling constructions SHALL be specified in a registered Protocol Profile and demonstrated to preserve completeness verification and auditor reconstruction.

> **WHAT S3P BOUNDS.** *S3P attests the* evaluator-judged *violation rate over a verifiably fair sample: an auditor can independently confirm that sampling was unbiased and complete, but the violation count itself is the verdict of the operator's evaluation instrument. The integrity of those verdicts is established separately — by version-binding the evaluation instrument (Section 16.1), signing the policy and baseline that define a violation (GOV-3.5), and retaining evidence commitments so that a verdict can be reproduced by re-running the version-attested evaluator against the retrieved evidence under Section 20.4 and Annex G.1, under appropriate authority. S3P proves the sample was honest; verdict reproducibility proves the judgments were.*

> **ROADMAP (INFORMATIVE).** *S3P's commit-then-reveal-and-recompute construction is the appropriate present-day method. Because it is commitment-first, it is compatible with a future succinct-proof upgrade: a registered Protocol Profile MAY later allow an operator to prove that the violation rate over the committed set, under a version-attested evaluator, is at or below a stated bound, without revealing the nonce-selected sample at all. The commitment architecture defined here is forward-compatible with that succinct-proof path.*

19.3 Safety claims SHALL carry exact confidence intervals computed using conservative statistical methods that avoid normal-approximation assumptions (exact binomial intervals under the S3P Bernoulli sampling model with binary evaluator verdicts). An upper-bound safety claim ("the violation rate did not exceed p at confidence 1 − α") SHALL be computed as a one-sided upper bound, and every S3P attestation SHALL declare its bound form (`one-sided-upper` or `two-sided`) so that bounds and sample sizes are interpreted under the correct form (Annex B.7, Annex B.8). Unquantified safety assertions are not attestation artifacts. [See Annex B: Protocol Profile Reference Summary for formula specifications.]

19.4 Per-epoch safety attestations SHALL include at minimum: total requests, sampled count, violation count, sampling rate, observed violation rate, confidence level, bound form, confidence interval bounds, denominator class (Section 4.6), and sampling methodology identifier.

19.5 An auditor SHALL be able to verify safety claims by: (a) obtaining published epoch secrets, (b) recomputing sampling decisions for all requests, (c) verifying sample-set membership, and (d) recomputing confidence intervals — all without accessing protected content.

19.6 Implementations SHALL conform to the statistical safety signal specifications in the registered Protocol Profile.

19.7 Signal Volume Prerequisites

19.7.1 Clopper–Pearson exact binomial confidence intervals require a minimum number of sampled events before the resulting upper bound on violation rate constitutes a credible statistical claim. Implementations SHALL NOT report a violation-rate bound tighter than the sample size supports. The following table specifies minimum sample sizes for common **one-sided upper confidence bound** and confidence-level combinations with zero observed violations. The bounds reported are one-sided upper bounds on the violation rate, consistent with §19.3's upper-bound claim form:

Target Upper Bound	Confidence Level	Min. Sampled Events (zero violations)
10% (0.1)	95%	29
5% (0.05)	95%	59
1% (0.01)	95%	299
0.5% (0.005)	95%	598
0.1% (0.001)	95%	2,995
1% (0.01)	99%	459
0.1% (0.001)	99%	4,603

> NOTE: *These values assume zero observed violations and are* ONE-SIDED UPPER CONFIDENCE BOUNDS *on the violation rate, computed at confidence* `1 - alpha` *(one-sided). For zero observed violations the minimum sample size is the smallest* `n` *satisfying* `(1 - p_bound)^n ≤ alpha`*. This is deliberately the one-sided form because the safety claim is an* upper bound *on the violation rate; it differs from the two-sided Clopper–Pearson interval defined in Annex B, which places* `alpha/2` *in each tail and therefore yields larger minimum sample sizes for the same nominal confidence (e.g., 368 rather than 299 at 95% / 1%). Any observed violation invalidates a zero-violation bound; the Clopper–Pearson interval then widens per standard exact binomial computation. Annex B.8 specifies both computations; the attestation's* `bound_form` *field declares which form its bounds carry, so an implementer provisioning from this table reports* `one-sided-upper` *rather than computing the two-sided form against a one-sided sample size.*

19.7.2 S3P attestations generated from epochs or aggregation windows where the number of sampled events falls below the minimum required for the claimed bound SHALL report the status code `ERR_INSUFFICIENT_SAMPLE` as defined in the registered Protocol Profile. Signal consumers SHALL NOT extrapolate a violation-rate bound from an epoch or aggregation window carrying `ERR_INSUFFICIENT_SAMPLE` status.

19.7.3 Deployments where the expected sampled-event volume within a single epoch is insufficient to meet the minimum sample size for the target bound SHALL use longer aggregation windows. Conformant approaches include:

(a) **Extended aggregation windows.** The implementation MAY aggregate sampled events over longer periods (e.g., daily, weekly). The aggregation period SHALL be explicitly disclosed in every S3P attestation produced under this mode.

(b) **Rolling windows.** The implementation MAY use a rolling window of the most recent n_min sampled events, provided the window boundary timestamps are included in the attestation.

Implementations SHALL NOT silently default to epoch-level bounds when volume is insufficient.

19.7.4 The coverage ratio reported in S3P attestations SHALL identify its denominator class (Section 4.6) and SHALL satisfy the denominator requirements of Section 4.7.4, including the Level 4 Independently-Attested requirement and its feasibility note. The denominator class carried in the attestation (`denominator_class`, MEA-2.6) SHALL match the class disclosed in the conformance claim.

20. Third-Party Auditability

Requirement: The attestation system SHALL enable third-party verification of AI governance claims without requiring trust in the operator or access to protected content.

20.1 All attestation receipts SHALL be recorded in an append-only transparency log conformant with RFC 6962 (Certificate Transparency) that provides: (a) inclusion proofs (receipt exists in log), (b) consistency proofs (log was not modified between time points), and (c) split-view detection (operator cannot show different logs to different auditors).

20.2 Machine-readable attestation packs SHALL be expressible in standard compliance formats (e.g., OSCAL Assessment Results) for interoperability with existing audit workflows.

20.3 Auditor verification SHALL be possible at three levels: (a) sampling integrity verification (using published ledgers and epoch secrets), (b) configuration integrity verification (using co-epoch bindings), and (c) content verification (accessing verifiable records in operator's local storage under legal authority; the retrieval interface is defined in Annex G.1 and is normative for all Level 4 claims).

20.4 Routine verification (levels a and b) SHALL operate entirely on cryptographic artifacts without content access. Content verification (level c) SHALL be the exception, used only under legal authority

or contractual agreement, with cryptographic proof that accessed attestation artifacts are genuine and contemporaneous.

20.5 Implementations SHALL conform to the auditor verification procedures in the registered Protocol Profile.

21. Legal Preservation and Production

Requirement: The attestation architecture SHALL define retention, preservation, export, and chain-of-custody requirements sufficient to support regulatory examination and litigation discovery, without compromising non-egress guarantees or cryptographic verifiability.

21.1 Retention Requirements

Operators SHALL define and publish a retention schedule for each attestation artifact class (receipts, attestation packs, S3P signals, ControlActions), mapped to applicable legal, regulatory, and contractual requirements. The retention schedule SHALL be attested in the transparency log. Operators are responsible for determining the retention periods appropriate to their jurisdictions and regulatory environment.

21.2 Legal Hold

Upon receipt of a litigation hold notice, preservation demand, or regulatory investigation notice, the operator SHALL:

(a) Suspend automated deletion of all attestation artifacts within scope.

(b) Generate a point-in-time attestation pack with transparency log inclusion proofs for all in-scope receipts.

(c) Attest the legal hold activation with timestamp and scope definition.

21.3 Immutable Export

Operators SHALL be capable of producing an immutable export package containing:

(a) All attestation receipts for a defined time period and scope.

(b) Transparency log inclusion and consistency proofs.

(c) Notary signatures and epoch data.

(d) S3P attestations and ControlActions.

(e) Custodian certification (identity of export operator, timestamp, scope declaration, hash of export package).

The export package SHALL be independently verifiable as to integrity, signature validity, and transparency-log consistency by any party with access to the transparency log and published epoch data. Production of operator-local artifacts (e.g., full CAS records) may require operator cooperation or lawful process.

21.4 Chain of Custody

Each attestation artifact SHALL include sufficient metadata to establish chain of custody: creation timestamp, creator identity (notary or arbiter), epoch binding, and transparency log position.

21.5 Receipt Temporality Classification

The receipt `flags` field SHALL be a fixed-width unsigned integer encoding attestation temporality. Two late-attestation classes are defined; they differ materially in evidentiary weight and SHALL NOT be conflated:

- `0x00` = **contemporaneous:** the attestation was generated and attested within the same epoch as the governed event.
- **Bit 0 (`0x01`) = DELAYED_NOTARY:** the attestation envelope was generated locally during the governed event — Phase 1 enforcement and the Phase 2 provisional receipt executed contemporaneously — but the notary counter-signature was obtained in a subsequent epoch (for example, transient notary unavailability spanning an epoch boundary).
- **Bit 1 (`0x02`) = RECONSTRUCTED:** the receipt was generated retroactively from locally logged records after a fail-open period, per RES-5.2. No contemporaneous Phase 2 receipt exists for the governed event; the chain of custody begins at the local fail-open log.

Remaining bits are reserved and SHALL be set to zero. The field width is specified in the registered Protocol Profile.

Receipts carrying either temporality bit SHALL NOT be counted as contemporaneous attestation coverage for conformance, risk-signal reporting (Annex D), or litigation reporting purposes, SHALL be distinguishable from contemporaneous receipts in all export packages, signal computations, and audit reports, and SHALL be reported as separate classes — a contemporaneous local record upgraded late is not the same evidence as a record reconstructed after the fact.

Submission semantics. A late receipt (either class) is submitted to the notary in the epoch current at submission time; its envelope SHALL carry the covered event interval (governed-event

epoch and timestamp) as metadata. The stale-epoch rejection of Section 18.5 and ATT-2.4 applies to the submission envelope's epoch reference, not to the covered event time. The registered Protocol Profile SHALL define the maximum late-submission window.

> NOTE — SCOPE OF DELAYED_NOTARY CLASSIFICATION: *DELAYED_NOTARY applies only when notary co-signing is delayed across an epoch boundary — that is, the governed event occurred in epoch* N *but the notary signature was obtained in epoch* N+1 *or later. Transient network latency* within *an epoch does not trigger reclassification. Under the three-phase attestation model (ATT-3), Phase 1 (policy enforcement) and Phase 2 (provisional receipt generation) execute locally and synchronously; only Phase 3 (notary co-signature upgrade) is asynchronous. Because the recommended epoch duration (18.1) is 300 seconds, routine network micro-outages are absorbed without reclassification.*

21.6 Redaction Procedures

When attestation packs must be produced with certain content redacted (e.g., to protect third-party PHI in multi-tenant environments), redaction SHALL preserve the cryptographic verifiability of unredacted portions. Redacted fields SHALL be replaced with their cryptographic commitments, enabling a verifier to confirm that the redacted content was present without accessing it. Redacted fields SHALL be salted with an operator-held secret prior to commitment generation. The salt SHALL be unique per field per attestation to prevent dictionary inversion of low-entropy content.

[See Annex C: Design Rationale for legal admissibility considerations and the role of attestation architecture in litigation readiness.]

PART 5: CONFORMANCE

> *A claim is worth what can be checked. This part fixes the levels of proof, the parties permitted to verify them, and what a conformance statement must put on the record — so that "we run OVERT" means something an outsider can test.*

22. Conformance

22.1 Overview

OVERT conformance is expressed as a composite claim combining a **maturity level** (1–4) and a **scope designator** (Core, Agentic, or Agentic-Extended). That claim describes the depth of control-execution evidence an implementation produces within its declared scope; it does not constitute a general representation that the system is secure, compliant, or safe.

OVERT conformance requires AAL-4 attestation for all controls designated as AAL-4 in this standard. Controls designated AAL-1, AAL-2, or AAL-3 require the specified level. Conformance is assessed per-control, not globally.

An AAL designation on a control specifies the assurance grade that control SHALL achieve **when the deployment architecture required by the claimed maturity level supports that grade**, per the architecture-to-maximum-AAL mapping in Section 4.1.1. At Levels 3 and 4, whose required architecture includes an independent notary, controls designated AAL-4 SHALL produce AAL-4 artifacts. At Levels 1 and 2, whose required architecture does not include an independent notary, a control designated AAL-4 SHALL be satisfied at the highest assurance level the level's required architecture supports — for example, AAL-2 operator-generated evidence for a Level 2 deployment with no independent attestation infrastructure, or AAL-3 where an operator-controlled notary is deployed. This grading does not relax the control-family requirements of the conformance matrix (Section 22.5); it determines the assurance grade at which a required control is evidenced for the claimed level. A conformance claim SHALL NOT represent that any control achieved AAL-4 unless the deployment architecture supporting the claim meets the AAL-4 requirements of Section 4.1.1.

The maturity level determines which governance domains, attestation architecture requirements, and response or preservation capabilities are in scope for verification. The scope designator determines whether the claim is limited to non-agentic operation or extends to agentic execution paths such as tool use, inter-agent coordination, delegated capability use, disclosure, drift governance, MCP trust governance, durable state governance, prompt registration, and delegated identity attestation.

Every conformance claim at any level SHALL include a human-readable scope summary identifying the systems, interfaces, and traffic classes covered, and a human-readable exclusions summary identifying what is not covered. Level 1 and Level 2 claims SHALL include the scope summary and exclusions summary. Level 3 and Level 4 claims SHALL additionally identify the mediation scope statement hash, the declared coverage percentage of the mediation scope relative to its denominator, the denominator class used for coverage and measurement claims (`Independently-Attested`, `Operator-Infrastructure`, or `Operator-Declared`; Section 4.6), and the total exposure-window duration (periods of unattested operation) during the claim period. A Level 3 or Level 4 claim whose deployment carries optimistic residue SHALL also disclose the percentage of in-scope actions executed as optimistic residue (ATT-3.5(d)) during the claimed period, including both the claim-period average and the worst single-epoch value.

A conformant implementation SHALL state its conformance using the grammar defined in Section 22.4 and SHALL satisfy every normative requirement associated with its claimed level and scope.

22.2 Maturity Levels

OVERT defines four cumulative maturity levels. Each level subsumes all requirements of the preceding level.

Level	Name	Governance Domains (Part 2)	Attestation Architecture (Part 4)	Summary
1	Foundation	GOVERN (Section 5): GOV-1 – GOV-5; IDENTIFY (Section 6): IDE-1, IDE-2	None	Documented governance basis and system characterization. Buyers and auditors may verify that policies, inventories, and impact assessments were documented. Runtime enforcement, continuous monitoring, and incident-grade evidence are outside Level 1.

Level	Name	Governance Domains (Part 2)	Attestation Architecture (Part 4)	Summary
2	Enforcement	Level 1 + PROTECT (Section 7): PRO-1 – PRO-5; HITL (Section 15): HITL-1, HITL-4	Section 17 (Non-Egress Architecture), Section 18 (Temporal Binding)	Adds attested boundary enforcement, non-egress architecture, temporal binding, and defined human approval paths. Buyers and auditors may verify that declared execution controls operated at the runtime boundary for in-scope actions.
3	Measurement	Level 2 + ATTEST (Section 8): ATT-1 – ATT-4; MEASURE (Section 9): MEA-1 – MEA-4; HITL (Section 15): HITL-2, HITL-3	Level 2 + Section 19 (Statistical Measurement), Section 20 (Auditability)	Adds independently useful telemetry, statistical measurement, transparency-log auditability, and full human-review evidence. Auditors and defenders may verify sampling integrity, coverage disclosures, measurement outputs, and attested review events for in-scope operations.
4	Evidence-Grade	Level 3 + RESPOND (Section 10): RES-1 – RES-5; ATTEST (Section 8): ATT-5	Level 3 + Section 21 (Legal Preservation)	Adds attested response actions, IAP governance, and preservation or export controls for later investigation. This is the highest OVERT evidence and preservation tier. It does not establish overall security, regulatory compliance, insurer endorsement, or entitlement to insurance coverage.

> NOTE. *HITL controls (Section 15) appear in Part 3 but are required at Core scope because human oversight obligations arise for both agentic and non-agentic systems at Levels 2–4. HITL is architec-*

> *turally situated in Part 3 for editorial coherence with the agentic control family, not because it is agentic-only.*

22.3 Scope Designators

The scope designator controls whether an implementation must additionally satisfy the agentic-specific control families in Part 3 (Sections 11–16, with HITL excluded from the scope gate). HITL (Section 15) is required by the maturity level regardless of scope.

Scope	Sections Required	Description
Core	Part 2 (Sections 5–10) per level + Section 15 (HITL) per level + Part 4 (Sections 17–21) per level	Systems that do not autonomously invoke external tools, coordinate with other agents, or operate under delegated authority.
Agentic	Core + TOOL (Section 11, excluding Section 11.5) + MULTI (Section 12) + CAP (Section 13) + DISC (Section 14) + DRIFT (Section 16), each per level	Systems that autonomously invoke external tools, participate in multi-agent orchestration, operate under delegated capability grants, or exhibit potential for goal drift. Does not use MCP or equivalent tool-hosting protocols for external tool invocation.
Agentic-Extended	Agentic + MCP (Section 11.5) + STATE (Section 15.6) + IDENT (Section 15.7), each per level	Agentic systems that additionally invoke tools through MCP servers (managed, custom, or external), persist durable agent state across session boundaries, register prompt artifacts, or operate under federated/delegated identity chains.

The controls required at each level are:

Level	Agentic Controls (in addition to Core)	Agentic-Extended Controls (in addition to Agentic)
1	None (policy and inventory only)	None (policy and inventory only)
2	TOOL-1 – TOOL-5, DRIFT-1, DRIFT-3.4	MCP-1 (if managed MCP), MCP-3.1 (if external MCP), STATE-2.1
3	Level 2 + MULTI-1 – MULTI-2, CAP-1 – CAP-2, DISC-1, DRIFT-2, DRIFT-3, DRIFT-5	Level 2 Extended + MCP-1 – MCP-3, STATE-1, STATE-2, IDENT-1.1 – IDENT-1.3
4	Level 3 + DRIFT-4	Level 3 Extended + IDENT-1.4, IDENT-1.5

An implementation deploying agentic capabilities SHALL claim Agentic scope. Claiming Core scope for a system that autonomously invokes external tools or coordinates with other agents is non-conformant regardless of maturity level.

An implementation deploying agentic capabilities that use MCP servers, persist durable agent state, register prompt artifacts, or operate under federated identity SHALL claim Agentic-Extended scope. Claiming Agentic scope (without the Extended qualifier) for a system that uses MCP servers or persists durable agent state is non-conformant. Where only a subset of the Agentic-Extended control families applies, the conformance statement Exclusions field SHALL declare the omitted family and the architectural justification.

> NOTE. *At Level 1, the Agentic scope designator indicates only that the system deploys agentic capabilities and that the operator has satisfied the Level 1 documentation requirements. It does not indicate that agentic-specific enforcement, monitoring, or drift controls are in place.*

> NOTE. *At Level 3 Agentic, CAP-2.1 and CAP-2.2 are elevated to AAL-3 (machine-generated enforcement telemetry). Claims about architectural separation based on CAP-2 at Level 3 therefore reflect operator-controlled telemetry-grade evidence, not cryptographically independent proof.* LEVEL 3 AGENTIC CONFORMANCE STATEMENTS SHALL EXPLICITLY STATE THAT CAP-2 EVIDENCE IS AAL-3 (OPERATOR-CONTROLLED TELEMETRY) IN THE CONFORMANCE CLAIM ITSELF, NOT ONLY IN SUPPORTING DOCUMENTATION. *At Level 4 Agentic, CAP-2.1 and CAP-2.2 require AAL-4 (independently verifiable evidence as defined in the registered Protocol Profile). Level 4 Agentic claims about architectural separation therefore require evidence beyond operator-controlled telemetry. Conformance claims at Level 4 Agentic that cannot satisfy AAL-4 for CAP-2 SHALL NOT assert evidence-grade architectural separation.*

22.4 Conformance Statement Grammar

A conformance claim SHALL take one of the following forms:

```
For Level 1 and Level 2:
OVERT Level <N> <Scope> — <Standard-Version>, <Profile-Version>, Scope Summary:
<Scope-Summary>, Exclusions: <Exclusions-Summary>, [ABD: <ABD-Hash>,] <Date>

For Level 3 and Level 4:
OVERT Level <N> <Scope> — <Standard-Version>, <Profile-Version>, Scope Summary:
<Scope-Summary>, Exclusions: <Exclusions-Summary>, Scope: <Coverage-Percent> of
<Denominator-Description>, Denominator: <Independently-Attested|Operator-
```

```
Infrastructure|Operator-Declared>, Scope Statement: <Scope-Hash>, Exposure Window:
<Exposure-Duration>, [Optimistic: <Optimistic-Average>/<Optimistic-Worst-Epoch> of
in-scope actions,] [Model Identity: <Included|Excluded|Not-Supported>,] [MCS:
<component=status; ...>,] [IAP Topology: <Single-IAP|Multi-IAP (t,n)>,] [Arbiter
Isolation: Software-Only,] [ABD: <ABD-Hash>,] <Date>
```

Where:

- `<N>` is the maturity level (1, 2, 3, or 4).
- `<Scope>` is `Core`, `Agentic`, or `Agentic-Extended`.
- `<Standard-Version>` is the OVERT standard version (e.g., `v1.1.0`).
- `<Profile-Version>` is the declared protocol profile version (e.g., `Profile v1.0`). For Level 1, where no protocol profile is operationally required, this field SHALL read `No Profile` or reference the intended target profile. For a Level 1 or Level 2 claim under a validly self-declared profile (Section 22.6.3), this field carries the `SD-`-prefixed profile identifier and version.
- `<Scope-Summary>` is a human-readable summary enumerating the system identifiers, interfaces, and traffic classes covered by the claim. The scope summary SHALL enumerate specific system identifiers (not generic descriptions), the interfaces through which attested traffic flows, and the traffic classes within scope. Generic or free-text-only scope summaries are non-conformant.
- `<Exclusions-Summary>` SHALL take one of three forms: (1) `None (full coverage verified)` — all identified in-scope traffic and interfaces are attested; (2) `Not assessed: <list>` — identified systems or interfaces that have not yet been evaluated for conformance, enumerated by identifier; (3) a specific exclusion list with per-item justification stating why each excluded item is outside the claim scope. Free-text exclusion summaries without per-item justification are non-conformant for Level 3 and Level 4 claims.
- `<Coverage-Percent>` is the declared mediation-scope coverage percentage for the claim.
- `<Denominator-Description>` is a human-readable description of the denominator used for the coverage claim (e.g., `inbound API traffic`).
- `<Independently-Attested|Operator-Infrastructure|Operator-Declared>` states the denominator class per Section 4.6: **Independently-Attested** (measurement source outside the operator's unilateral administrative control, or cryptographically co-attested by an independent party — counterparty counts, provider co-attestation, IAP-observed ingress); **Operator-Infrastructure** (operator-administered telemetry such as load balancer request counts or API gateway metrics, not externally co-attested); or **Operator-Declared** (asserted by the operator without an infrastructure measurement source).

- `<Scope-Hash>` is the mediation scope statement hash identifying the published scope artifact.
- `<Exposure-Duration>` is the total duration of unattested operation (exposure windows) during the claim period, expressed as hours and as a percentage of the claim period. If zero, this field SHALL read `0h (0%)`.
- `<Optimistic-Average>` is the claim-period average percentage of in-scope actions executed as optimistic residue (ATT-3.5(d)).
- `<Optimistic-Worst-Epoch>` is the worst single-epoch optimistic-residue percentage during the claim period.
- `IAP Topology: <Single-IAP|Multi-IAP (t,n)>` is mandatory for ALL Level 4 claims, and optional for Level 3 claims (included where an independent notary or IAP is part of the deployment). Multi-IAP deployments SHALL state the threshold parameters `(t, n)` and satisfy the entity-diversity requirements of ATT-5.3. A Single-IAP deployment declares attestation independence without multi-party resilience (Section 4.1.1) and SHALL NOT enable cross-tenant revocation (RES-4.2).
- `Model Identity: <Included|Excluded|Not-Supported>` is mandatory for Level 4 claims and RECOMMENDED for Level 3 claims. It states whether model/runtime identity is bound into co-epoch attestation (Section 3.3); `Excluded` and `Not-Supported` claims convey that OVERT conformance does not attest the model weights, provider runtime, or inference runtime (Section 22.10).
- `MCS: <component=status; ...>` is the Measured Component Set — mandatory for Level 4 claims, RECOMMENDED for Level 3. Each evidence-relevant component (`arbiter`, `policy`, `evaluator`, `classifier`, `scanner`, `model`, `mcp`, `state`, `prompts`, `identity`) is assigned `M` (measured), `OA` (operator-attested), `X` (excluded), or `NA` (not applicable), as defined in Section 22.10; an unlisted canonical component is treated as `X`, and the `model` entry SHALL agree with the Model Identity field.
- `Arbiter Isolation: Software-Only` is included when Section 4.7.3(f) requires disclosure that the AAL-4 arbiter is not running in a hardware-attested TEE.
- `ABD: <ABD-Hash>` is mandatory for Agentic-Extended claims and identifies the published Attestation Boundary Declaration defined in Section 22.10.
- `<Date>` is the ISO 8601 date on which the conformance assessment was completed.

Examples:

```
OVERT Level 2 Core — v1.1.0, Profile v1.0, Scope Summary: sys-cda-001 clinical
documentation API (FHIR R4 interface, HL7v2 ADT feed), Exclusions: Not assessed:
batch-analytics-002 (scheduled for Q3 assessment), 2026-03-15
OVERT Level 3 Agentic — v1.1.0, Profile v1.0, Scope Summary: sys-agent-010 patient-
facing agentic workflows (API gateway gw-prod-01, FHIR interface, voice endpoint),
Exclusions: None (full coverage verified), Scope: 85% of inbound API traffic,
```

```
Denominator: Independently-Attested, Scope Statement: sha256:<scope-hash>, Exposure
Window: 0h (0%), IAP Topology: Multi-IAP (2,3), 2026-02-28
OVERT Level 4 Agentic-Extended — v1.1.0, Profile v1.0, Scope Summary: sys-agent-010
and sys-cds-020 production agentic workflows (API gateway gw-prod-01, internal RPC
mesh, FHIR R4 interface), Exclusions: None (full coverage verified), Scope: 100% of
declared in-scope actions, Denominator: Independently-Attested, Scope Statement:
sha256:<scope-hash>, Exposure Window: 2h (0.03%), Optimistic: 8%/22% of in-scope
actions, Model Identity: Excluded, MCS: arbiter=M; policy=M; evaluator=M;
classifier=OA; scanner=M; model=X; mcp=M; state=M; prompts=M; identity=M, IAP
Topology: Single-IAP, ABD: sha256:<abd-hash>, 2026-03-15
OVERT Level 1 Core — v1.1.0, No Profile, Scope Summary: sys-ambient-005 ambient
clinical documentation system (voice capture endpoint, EHR integration interface),
Exclusions: Not assessed: sys-transcribe-006 non-AI transcription workflows,
2026-01-10
```

> NOTE. *Conformance claims are point-in-time assertions. A conformance claim does not represent ongoing conformance unless accompanied by continuous attestation evidence at Level 3 or above. Implementations SHOULD include the standard version and profile version in all conformance documentation.*

22.5 Conformance Matrix

The following matrix maps the primary control-family requirements for each Level–Scope combination. This matrix is non-exhaustive: conformance additionally requires satisfaction of the normative overlays in Sections 4.1 (AAL mapping), 4.5 (threat model mitigations), 4.6 (risk signal properties and verifiability classification), 4.7 (security considerations including IAP compromise response, log monitor diversity, arbiter hardening, mediation scope attestability, and anomaly triage), 4.8 (cross-boundary attestation for cross-boundary workflows), 22.1 (scope and exclusions disclosure), 22.6 (protocol profile registration), 22.7 (IAP qualification), and 22.8 (qualified assessor requirements). All applicable normative overlays SHALL be satisfied for the claimed level. A cell marked **R** indicates the requirement is mandatory (SHALL). A cell marked **S** indicates the requirement is recommended (SHOULD). A cell marked **—** indicates the requirement does not apply. All requirements are cumulative: Level N includes all requirements from Levels 1 through N−1.

Section	Control Family	L1 Core	L1 Agentic	L2 Core	L2 Agentic	L3 Core	L3 Agentic	L4 Core	L4 Agentic

Part 2

Section	Control Family	L1 Core	L1 Agentic	L2 Core	L2 Agentic	L3 Core	L3 Agentic	L4 Core	L4 Agentic
§5	GOVERN (GOV-1 – GOV-5)	R	R	R	R	R	R	R	R
§6	IDENTIFY (IDE-1, IDE-2)	R	R	R	R	R	R	R	R
§7	PROTECT (PRO-1 – PRO-5)	—	—	R	R	R	R	R	R
§8	ATTEST (ATT-1 – ATT-4)	—	—	—	—	R	R	R	R
§8	ATTEST (ATT-5)	—	—	—	—	—	—	R	R
§9	MEASURE (MEA-1 – MEA-4)	—	—	—	—	R	R	R	R
§10	RESPOND (RES-1 – RES-5)	—	—	—	—	—	—	R	R
Part 3									
§11	TOOL (TOOL-1 – TOOL-5)	—	—	—	R	—	R	—	R
§12	MULTI (MULTI-1	—	—	—	—	—	R	—	R

Section	Control Family	L1 Core	L1 Agentic	L2 Core	L2 Agentic	L3 Core	L3 Agentic	L4 Core	L4 Agentic
	– MULTI-2)								
§13	CAP (CAP-1 – CAP-2)	—	—	—	—	—	R	—	R
§14	DISC (DISC-1)	—	—	—	—	—	R	—	R
§15	HITL (HITL-1, HITL-4)	—	—	R	R	R	R	R	R
§15	HITL (HITL-2, HITL-3)	—	—	—	—	R	R	R	R
§15.5	SESS (SESS-1 – SESS-5)	—	—	R	R	R	R	R	R
§16	DRIFT (DRIFT-1, DRIFT-3.4)	—	—	—	R	—	R	—	R
§16	DRIFT (DRIFT-2, DRIFT-3, DRIFT-5)	—	—	—	—	—	R	—	R
§16	DRIFT (DRIFT-4)	—	—	—	—	—	—	—	R
§16.1	EVAL (EVAL-1 – EVAL-4)	—	—	—	—	—	R	—	R
Part 4									
§17	Non-Egress Architecture	—	—	R	R	R	R	R	R

Section	Control Family	L1 Core	L1 Agentic	L2 Core	L2 Agentic	L3 Core	L3 Agentic	L4 Core	L4 Agentic
§18	Temporal Binding	—	—	R	R	R	R	R	R
§19	Statistical Measurement	—	—	—	—	R	R	R	R
§20	Auditability	—	—	—	—	R	R	R	R
§21	Legal Preservation	—	—	—	—	—	—	R	R
Part 1									
§4.8	Cross-Boundary Attestation	—	—	—	—	R	R	R	R

> NOTE. *§15.5 (SESS) applies at Level 2+ for systems with session-based interactions; systems without session-based interactions are exempt. §16.1 (EVAL) applies at Level 3+ Agentic. §4.8 (Cross-Boundary Attestation) applies at Level 3+ for cross-boundary workflows; single-boundary deployments are exempt. §18 (Temporal Binding) at Level 2 operates in the operator-attested mode of Section 18.2; notary-derived binary identity applies from Level 3. §22.8 (Qualified Assessor) applies as: Level 3 SHOULD, Level 4 SHALL.*

AGENTIC-EXTENDED OVERLAY

Agentic-Extended claims add the following control-family overlays on top of the base Agentic matrix:

Control	Level 1	Level 2	Level 3 Agentic-Extended	Level 4 Agentic-Extended
MCP-1.1	—	Required (if managed MCP)	Required	Required
MCP-1.2	—	Required (if managed MCP)	Required	Required
MCP-1.3	—	Required (if managed MCP)	Required	Required
MCP-1.4	—	Required (if managed MCP)	Required	Required
MCP-2.1	—	—	Required (if custom MCP)	Required
MCP-2.2	—	—	Required (if custom MCP)	Required
MCP-2.3	—	—	Required (if custom MCP)	Required
MCP-2.4	—	—	Required (if custom MCP)	Required
MCP-3.1	—	Required (if external MCP)	Required	Required
MCP-3.2	—	—	Required (if external MCP)	Required
MCP-3.3	—	—	Required (if external MCP)	Required
MCP-3.4	—	—	Required (if external MCP)	Required
MCP-3.5	—	—	Required (if external MCP)	Required
STATE-1.1	—	—	Required	Required
STATE-1.2	—	—	Required	Required
STATE-1.3	—	—	Required	Required
STATE-1.4	—	—	Required	Required
STATE-1.5	—	—	Required	Required
STATE-2.1	—	Required	Required	Required
STATE-2.2	—	—	Required	Required
STATE-2.3	—	—	Required	Required
STATE-2.4	—	—	Required	Required
STATE-2.5	—	—	Required	Required
IDENT-1.1	—	—	Required	Required
IDENT-1.2	—	—	Required	Required

Control	Level 1	Level 2	Level 3 Agentic-Extended	Level 4 Agentic-Extended
IDENT-1.3	—	—	Required	Required
IDENT-1.4	—	—	—	Required
IDENT-1.5	—	—	—	Required

22.6 Protocol Profile Registry Governance

A protocol profile defines the specific cryptographic primitives, serialization formats, and transport bindings that satisfy the normative profile-dependent clauses throughout the standard. Profile-dependent clauses appear in Part 2 (Sections 5–10) for schema definitions, governance artifact formats, and measurement output structures, and in Part 4 (Sections 17–21) for cryptographic algorithms, envelope structures, hash functions, signature schemes, key hierarchies, and evidence serialization.

An implementation claiming OVERT Level 2 or above SHALL reference a registered protocol profile. The profile SHALL cover all profile-dependent normative clauses applicable to the claimed level — not solely those in Part 4.

22.6.1 REGISTRY PUBLICATION

The Protocol Profile Registry SHALL be published at a stable URL with complete version history. Each registry entry SHALL include the profile identifier, version, submission date, registration date, and a persistent link to the full profile specification.

22.6.2 SUBMISSION AND REGISTRATION

Any party MAY submit a protocol profile for registration. The registry maintainer SHALL accept or reject submissions within 90 calendar days of receipt. A submission SHALL satisfy:

1. **Normative coverage.** The profile SHALL specify concrete cryptographic constructions, envelope schemas, key derivation methods, and receipt formats satisfying every normative SHALL requirement applicable to the claimed scope.
2. **Test vectors.** The profile SHALL include published test vectors for every cryptographic operation, including cross-boundary reference vectors covering the `parent_attestation_id` derivation of Section 4.8.2.
3. **Deterministic verification.** Given identical inputs and the profile specification, any two independent implementations SHALL produce identical cryptographic outputs.
4. **Public specification.** The profile specification SHALL be publicly available for inspection.
5. **Patent disclosure.** The profile submission SHALL disclose any known patent claims that may be essential to implementation.

6. **Conformance test suite.** The submission SHALL include or reference a publicly available conformance test suite sufficient to verify implementation correctness.

Evaluation is mechanical: profiles meeting all criteria SHALL be registered. The registry maintainer SHALL NOT reject profiles on grounds other than failure to meet the criteria above.

22.6.3 SELF-DECLARATION UPON REGISTRY NON-RESPONSE

If the registry maintainer fails to issue a written acknowledgment within 14 calendar days, or fails to render a decision within 90 calendar days of receipt of a complete submission, the submitter MAY publish the profile as a self-declared profile. A self-declared profile SHALL include a prominent notice stating that registry registration was attempted but no response was received, and SHALL use the profile identifier prefix `SD-` to distinguish it from registry-registered profiles. A self-declared profile is valid for Level 1 and Level 2 conformance claims. A self-declared profile SHALL NOT be used for Level 3 or Level 4 conformance claims. Level 3 and Level 4 claims require a registry-registered profile because the evidence-grade and measurement-grade claims at those levels depend on third-party-reviewed cryptographic constructions, test vectors, and conformance test suites that self-declaration cannot provide.

Reading rule. Throughout this standard, references to "the registered Protocol Profile" as the source of profile-defined specifics are read, for a Level 1 or Level 2 deployment operating under a validly self-declared profile, as references to the deployment's declared (`SD-`) profile. The self-declared profile SHALL define every profile-delegated item required by the sections applicable at the claimed level.

22.6.4 REGISTRY GOVERNANCE POLICY

The registry governance policy SHALL be published alongside the registry and SHALL specify criteria for acceptance and rejection, appeals process, update and deprecation procedures, registry maintainer identity and contact information, and succession conditions. Changes to the governance policy SHALL be published with at least 60 calendar days advance notice.

22.6.5 REGISTRY CONTINUITY

If the registry maintainer ceases operations for more than 180 consecutive calendar days, any organization MAY establish a successor registry provided it incorporates all entries from the prior registry, publishes a governance policy meeting the requirements of Section 22.6.4, and provides at least 90 calendar days public notice before accepting new submissions.

22.7 Independent Attestation Provider (IAP) Qualification

An entity operating as an Independent Attestation Provider (IAP) per Section 3.14 SHALL satisfy the following requirements:

Structural independence:

1. The IAP SHALL NOT hold equity in, be a subsidiary of, or share common management with the AI system operator whose attestations it validates.
2. The IAP SHALL maintain contractual independence: the operator SHALL NOT have unilateral authority to suppress, modify, or delay attestation artifacts.
3. The IAP SHALL disclose any material business relationships with operators whose attestations it validates.
4. The IAP SHALL disclose its beneficial ownership structure to operators upon request.

Operational requirements:

5. The IAP SHALL publish uptime and availability metrics for its notary infrastructure.
6. The IAP SHALL provide auditor access to epoch nonces, digest publication ledgers, and transparency log entries as required by Section 20.
7. The IAP SHALL maintain key management practices consistent with the security requirements of the registered Protocol Profile.
8. The IAP SHALL demonstrate operational capability for all Part 4 sections required by the highest level it services.

Transparency and accountability:

9. The IAP SHALL publish its operational policies, including key management practices, geographic distribution of notary infrastructure, and incident response procedures.
10. The IAP SHALL disclose any security incidents affecting attestation integrity within 72 hours of detection (see Section 4.7.1).
11. The IAP SHALL publish a transparency report at least annually, covering receipt volume, verification failure rates, and any compromise or coercion events to the extent permitted by law.

Portability and resilience:

12. Operators SHALL be able to transition between IAPs without loss of historical attestation data. The outgoing IAP SHALL provide transparency log entries and published epoch data for the transition period.
13. The IAP SHALL support portability escrow: upon operator request, the IAP SHALL export all configuration artifacts, epoch data, and transparency log entries necessary for a replacement

IAP to assume attestation services. The export format SHALL be documented and publicly specified.

14. For Level 4 claims, the IAP SHALL cooperate with the operator's annual migration rehearsal (Section 4.7.1(g)) by providing a test environment or equivalent mechanism sufficient to validate the portability escrow.

Any entity meeting these requirements MAY operate as an IAP. This standard does not restrict IAP operation to any specific commercial entity.

22.8 Qualified OVERT Assessor Program

This section defines the requirements for third-party assessors who evaluate OVERT conformance on behalf of operators, relying parties, or regulatory bodies.

22.8.1 PURPOSE

A Qualified OVERT Assessor is a third-party entity that independently evaluates an operator's OVERT conformance claim against the normative requirements of this standard. The Qualified Assessor program establishes minimum competence, independence, and procedural requirements for assessment activities, ensuring that conformance claims at higher maturity levels are subject to rigorous independent evaluation.

22.8.2 ASSESSOR REQUIREMENTS

An entity seeking qualification as an OVERT Assessor SHALL satisfy the following requirements:

Independence:

(a) The assessor SHALL be structurally independent of the assessed organization. The same structural independence requirements applicable to IAPs (Section 22.7, items 1–4) apply to assessors, adapted for assessment: the assessor SHALL NOT hold equity in, be a subsidiary of, or share common management with the organization whose conformance it assesses. The assessor SHALL maintain contractual independence and disclose any material business relationships with assessed organizations.

(b) The assessor SHALL not have provided implementation consulting, system design, or protocol profile development services to the assessed organization for the system under assessment within the 24 months preceding the assessment. This does not preclude prior training or educational engagements.

Competence:

(c) The assessor SHALL demonstrate competence in: (i) OVERT standard interpretation — documented understanding of all normative requirements across Parts 1–5, including version-specific

changes; (ii) cryptographic verification procedures — ability to independently verify receipt signatures, co-epoch bindings, transparency log inclusion and consistency proofs, and S3P attestation recomputation; (iii) attestation infrastructure assessment — ability to evaluate deployment topology, arbiter isolation, notary governance, and mediation scope completeness; (iv) governance framework evaluation — documented understanding of the external governance frameworks against which the operator's deployment is mapped (e.g., NIST AI RMF, ISO/IEC 42001) sufficient to evaluate OVERT conformance claims in context.

(d) The assessor SHALL maintain a documented assessment methodology aligned with this standard, including checklists, evidence collection procedures, and report templates.

Professional standards:

(e) The assessor SHALL maintain professional liability coverage adequate to the scope of assessments performed.

(f) The assessor's assessment staff SHALL complete annual continuing education in AI governance and attestation, with documented training records. A minimum of 16 hours of continuing education per year is RECOMMENDED.

22.8.3 ASSESSMENT PROCEDURES

Qualified Assessors SHALL conduct assessments using the following procedures:

(a) **Scope validation.** Verify that the claimed conformance scope (systems, interfaces, traffic classes) matches the actual deployment. The assessor SHALL independently confirm that the systems identified in the conformance statement are the systems under attestation, and that the mediation scope statement accurately describes the attested traffic.

(b) **Control verification.** Test each claimed control against normative requirements at the claimed level. For AAL-4 controls, the assessor SHALL independently verify at least one representative attestation artifact per control family (receipt signature, co-epoch binding, transparency log proof). For AAL-1 through AAL-3 controls, the assessor SHALL review the documented evidence.

(c) **Evidence review.** Verify attestation artifacts against transparency log entries. The assessor SHALL independently retrieve receipts from the transparency log, verify inclusion proofs, and confirm that the artifacts presented by the operator match the log entries.

(d) **Signal validation.** Independently recompute risk signals from published data for at least one representative epoch. The assessor SHALL verify that the operator's reported coverage ratio, violation rate bounds, and gap accounting are consistent with the published epoch data and transparency log entries. Recomputation establishes arithmetic consistency over published inputs; it does not by itself establish denominator independence. The assessor SHALL determine the denominator class

(Section 4.6) and whether the denominator was independently verified, and SHALL state both in the assessment report.

(e) **Optimistic-residue eligibility review (Level 4).** For Level 4 claims, review the operator's action-class taxonomy and capability-issuance policy (ATT-3.5(f)): whether side-effecting actions are correctly classified, and whether capability artifact scope, lifetime, and issuance practice are consistent with the mediation scope statement.

(f) **Report generation.** Produce a standardized assessment report per Section 22.8.4.

22.8.4 ASSESSMENT REPORTS

Assessment reports SHALL include:

(a) **Assessed organization and system identification.** Legal entity name, system identifiers, deployment environment description, and attestation infrastructure description (IAP identity, protocol profile, deployment topology).

(b) **Claimed conformance level and scope.** The operator's conformance statement (per Section 22.4 grammar) as claimed.

(c) **Assessment date range and methodology version.** The start and end dates of the assessment period, the OVERT standard version assessed against, and the assessor's methodology version.

(d) **Per-control findings.** For each control applicable to the claimed level and scope: conformant, non-conformant, or not applicable. Non-conformant findings SHALL include a description of the deficiency and the normative requirement not satisfied.

(e) **Signal verification statement.** For Level 3 and Level 4 claims: the recomputed risk signals, the denominator class (Section 4.6), and an explicit statement of whether the denominator was independently verified (yes/no, with the verification source).

(f) **Identified deficiencies and recommended remediation.** A summary of all non-conformant findings with specific remediation recommendations and, where applicable, a recommended timeline for remediation.

(g) **Assessor certification and signature.** The lead assessor's identity, the assessor organization's identity, the assessor's qualification status (registry identifier per Section 22.8.5), and a signed certification that the assessment was conducted in accordance with this standard and the assessor's documented methodology.

22.8.5 ASSESSOR REGISTRY

GLACIS Technologies or successor registry maintainer SHALL maintain a public registry of Qualified OVERT Assessors. The registry SHALL include: assessor organization identity, qualification date, qualification scope (which maturity levels and scope designators the assessor is qualified to assess), and annual requalification status. Registry governance SHALL follow the same continuity provisions as the Protocol Profile Registry (Section 22.6.5).

22.8.6 LEVEL REQUIREMENTS FOR ASSESSMENT

(a) Level 1 and Level 2 conformance MAY be self-assessed by the operator.

(b) Level 3 conformance SHOULD use a Qualified OVERT Assessor. Self-assessment at Level 3 SHALL be disclosed in the conformance statement.

(c) Level 4 conformance SHALL use a Qualified OVERT Assessor. Level 4 conformance claims not supported by a Qualified Assessor's assessment report are non-conformant.

(d) A managed deployment, hosted reference implementation, or implementation-vendor attestation service MAY improve deployment assurance and evidence readiness, but SHALL NOT be represented as equivalent to Qualified Assessor certification unless the assessment is performed by an entity satisfying the independence requirements of this section.

22.9 Envelope Format vs. Conformance Claims ("OVERT-Compatible")

Systems MAY produce attestation artifacts using the OVERT envelope format (defined in Section 17 and detailed in Annex B) without making a conformance claim. Such a system MAY describe itself as **OVERT-Compatible**: a designation indicating structural compatibility with OVERT tooling and verification procedures, offered as a zero-commitment on-ramp. It does not constitute an assertion that the system satisfies the normative requirements of any Attestation Assurance Level, and SHALL NOT be presented as a conformance claim.

An OVERT conformance claim requires satisfaction of all normative requirements at the claimed AAL level, assessment by a qualified assessor (where required by Section 22.8.6), and disclosure per Section 22.3.

22.10 Attestation Boundary Declaration

Conformant implementations SHALL publish an **Attestation Boundary Declaration** (ABD) specifying which system surfaces are within the attested runtime boundary and which are outside it. The ABD SHALL be published to the transparency log and referenced in the conformance statement.

The ABD SHALL address, at minimum:

Surface Category	In-Scope Indicator	Out-of-Scope Indicator
MCP servers (managed)	Server identity, transport, governance metadata attested per MCP-1	Vendor internal operations, hosting lifecycle
MCP servers (custom)	Binary identity, network isolation, authorization attested per MCP-2	Deployment automation, patching lifecycle
MCP servers (external)	Connection governance, capability scoping, output filtering attested per MCP-3	External server internal security posture
Agent durable state	State sealing, lineage, mutation provenance attested per STATE-1	Storage infrastructure security, backup/recovery
Prompt artifacts	Registration, binding, change governance attested per STATE-2	Prompt engineering methodology, content quality
Identity delegation	Delegation chain, scope narrowing, token lifecycle attested per IDENT-1	IdP implementation, credential storage
Model and inference runtime	Model-identity binding where a measurement path exists (Section 3.3 note)	Model weights, training pipeline, provider runtime internals
Evaluators and local classifiers	Evaluator version identifier and artifact hash attested per EVAL-4; classifier identity where measured (Section 3.23)	Evaluator training data, rubric quality, upstream model provenance
Scanners	Scanner binary identity and configuration where measured (Section 3.22)	Detection efficacy, signature freshness
Unmanaged clients	(Not attestable — outside runtime boundary)	Client-side code, credential storage, UI integrity
Secret storage	(Not attestable — outside runtime boundary)	Secret rotation, vault implementation, access control

Where a surface is partially in scope (e.g., the arbiter attests transport security to an external MCP server but not the server's internal posture), the ABD SHALL state the boundary precisely.

The ABD SHALL additionally state: (a) the **attestation topology** — `Single-IAP` or `Multi-IAP (t,n)` (Section 22.4) with the operating entities' independence relationships (ATT-5.1, ATT-5.3); and (b) the **model-identity binding status** — `Included`, `Excluded`, or `Not-Supported` — so that relying parties cannot misread arbiter-scoped attestation as model-runtime attestation (Section 3.3).

Measured Component Set (MCS). Every Level 4 conformance claim SHALL carry, and Level 3 claims SHOULD carry, a Measured Component Set field in the conformance statement (Section 22.4) assigning each evidence-relevant component exactly one status:

- `M` **(measured)** — component identity is cryptographically bound into epoch attestation (e.g., arbiter per ATT-2.2; evaluator per EVAL-4);
- `OA` **(operator-attested)** — declared and operator-signed but not independently measured;
- `X` **(excluded)** — outside the attested boundary, with justification in the ABD or the Exclusions field;
- `NA` **(not applicable)** — the component class does not exist in the deployment.

The canonical component keys are: `arbiter`, `policy` (signed policy artifacts, GOV-3.5), `evaluator` (Section 16.1/EVAL-4), `classifier` (Section 3.23), `scanner` (Section 3.22), `model` (model/runtime identity — SHALL agree with the Model Identity field), `mcp` (MCP-1/2/3), `state` (STATE-1), `prompts` (STATE-2), and `identity` (IDENT-1). A canonical component absent from a claimed MCS SHALL be treated as `X`. The MCS exists to close a specific misreading: a measured arbiter presented as if it covered unmeasured evaluators, classifiers, or scanners — the components that produce the verdicts and detections the attestation stream reports.

22.11 Standard Versioning and Errata Policy

This standard is versioned `MAJOR.MINOR.PATCH`.

- **PATCH** releases contain editorial corrections and technical corrigenda only. A technical corrigendum resolves an internal contradiction or defect in the published text and SHALL NOT introduce a new requirement, remove an existing requirement, or change any conformance level definition. A conformance claim citing version `X.Y.Z` remains valid against any later `X.Y` patch release.
- **MINOR** releases MAY add normative content (new controls, annexes, transport bindings, or extension points) but SHALL be additive: an implementation conformant to version `X.Y` remains conformant to version `X.Y+1` without modification. New obligations bind only conformance claims that cite the new minor version or later.
- **MAJOR** releases MAY remove or alter existing normative requirements, renumber sections, or change envelope schemas.

Control identifiers (e.g., ATT-3.5, GOV-5.6) are stable handles: they SHALL NOT be renumbered or reassigned within a MAJOR version. Section numbers MAY change only in MAJOR releases; cross-references in conformance tooling SHOULD therefore bind to control identifiers rather than section numbers.

Errata are published at overt.is alongside each release. The changelog in the front matter of each release enumerates the changes in that release and their classification under this policy.

Annex A: Glossary (Informative)

The following terms and acronyms are used throughout this standard. Where a term has a specific OVERT definition that differs from common usage, the OVERT-specific definition is provided.

Term ID	Term	Definition
A.1	AAL	**Attestation Assurance Level.** One of four tiers (AAL-1 through AAL-4) describing the cryptographic verifiability and independence of governance attestation artifacts. AAL-1: Policy Documentation (self-asserted). AAL-2: Process Records (self-attested, auditor must trust operator). AAL-3: Automated Monitoring (machine-generated, operator-controlled). AAL-4: Cryptographic Attestation (third-party verifiable, zero content access required). See Section 4.1.
A.2	Arbiter	The enforcement sidecar that intercepts AI system actions (tool calls, API requests, data access) and evaluates them against policy before permitting execution. Implemented as an enforcement module; the specific runtime technology is specified by the applicable Protocol Profile. The arbiter generates attestation envelopes for every intercepted action.
A.3	OVERT	**Observable Verification Evidence for Runtime Trust.** This standard.
A.4	Attestation Artifact	A cryptographically signed record produced by the OVERT attestation infrastructure demonstrating that a specific governance control executed at a specific time under a specific configuration. Includes envelopes, receipts, S3P attestations, and ControlActions.
A.5	Attestation Pack	A bundled collection of attestation artifacts sufficient to demonstrate conformance for a defined scope and time period. Includes receipts, transparency log proofs, epoch data, S3P attestations, and ControlActions.
A.6	BLS	**Boneh-Lynn-Shacham.** A pairing-based signature scheme permitting efficient aggregation of multiple signers' signatures into a single compact signature. Used for notary threshold signatures in Protocol Profile 1.0. Not post-quantum resistant; the standard requires hybrid classical + post-quantum constructions, or pure post-quantum constructions, after January 1, 2031. Alternative notary signature constructions (e.g., multi-signature with Ed25519 or ML-DSA) may be specified by other Protocol Profiles.
A.7	CAS	**Content-Addressable Storage.** Local storage within the operator's environment where attestation evidence (prompts, responses, evaluations) is stored indexed by cryptographic commitment. Content never leaves the operator's

Term ID	Term	Definition
		boundary on the attestation path; disclosure to an authorized verifier occurs only under the exception procedures of Section 20.4 and Annex G.1.
A.8	CBOR	**Concise Binary Object Representation.** Binary data serialization format (RFC 8949). Protocol Profile 1.0 uses CBOR deterministic encoding per Section 4.2 of RFC 8949 for canonical byte-level representation. The standard requires deterministic encoding as a property (Section 17.1); the specific format — CBOR, JSON via JCS (RFC 8785), or other deterministic encoding — is specified by the applicable Protocol Profile.
A.9	CI (Confidence Interval)	A statistical interval computed using the Clopper-Pearson exact method providing bounds on an evaluator-judged violation rate with guaranteed coverage probability. Carries a declared bound form (`one-sided-upper` for upper-bound safety claims; `two-sided` for interval reporting); see Annex B.8. Used in S3P attestations.
A.10	Co-epoch Binding	The cryptographic binding of an attestation receipt to the binary identity, network isolation state, and configuration of the system under attestation at the time the attestation was produced. Ensures that attestations cannot be replayed across different system configurations.
A.11	ControlAction	A cryptographically attested record of a governance response to a detected violation. Includes action type, timestamp, scope, and co-epoch binding.
A.12	DPL	**Digest Publication Ledger.** A per-epoch publication of request commitments (never raw digests) enabling auditor verification of sampling fairness without content access.
A.13	Epoch	A bounded time interval (configurable; recommended default: 300 seconds) during which attestation parameters remain constant. Epoch boundaries trigger nonce publication, key rotation, and S3P computation.
A.14	HKDF	**HMAC-based Key Derivation Function.** Key derivation per RFC 5869. Protocol Profile 1.0 uses HKDF for deriving tenant_pepper, storage_key, sampling_key, and epoch_secret from root secrets within the split-knowledge key hierarchy. The standard requires key derivation as specified by the applicable Protocol Profile.
A.15	HMAC	**Hash-based Message Authentication Code.** Keyed hash function per RFC 2104. Protocol Profile 1.0 uses HMAC for request commitments, evidence commitments, PRF tags, and S3P sampling tags with domain separation prefixes. The standard requires keyed commitment functions as specified by the applicable Protocol Profile.
A.16	IAP	**Independent Attestation Provider.** An entity structurally independent of the AI system operator that operates notary infrastructure, validates attesta-

Term ID	Term	Definition
		tions, and publishes transparency log entries. An IAP does not access protected content. Multiple IAPs may operate under different governance models.
A.17	Base Envelope	The baseline attestation envelope emitted for every AI request. Contains 9 fields including blinded identifier, request commitment, encoder binary identity, and metadata. Closed schema — no additional fields permitted. See Annex B for architecture; full field-level schema is in the Protocol Profile.
A.18	Extended Envelope	The extended attestation envelope emitted for sampled requests. Contains 10 fields including the full PRF tag for auditor recomputation and policy evaluation scores. Closed schema. See Annex B for architecture; full field-level schema is in the Protocol Profile.
A.19	NETATT	**Network Attestation.** A per-epoch attestation of the system's network isolation state, covering at minimum the effective egress policy, the enforcement component identity, and the TLS certificate pin set; operators may include additional deployment-specific inputs such as network policy controller identity, eBPF state, CNI configuration, and environment variables affecting AI behavior (Section 18.3). Bound to all receipts issued during the epoch.
A.20	OSCAL	**Open Security Controls Assessment Language.** NIST-developed machine-readable format for security control documentation. OVERT attestation packs are expressible as OSCAL Assessment Results.
A.21	Late-Attested Receipt	An attestation receipt carrying either temporality flag of Section 21.5: **DELAYED_NOTARY** (envelope generated during the governed event; notary counter-signature landed in a subsequent epoch) or **RECONSTRUCTED** (receipt generated retroactively from local fail-open records per RES-5.2). Both classes are excluded from contemporaneous attestation coverage for conformance, risk-signal reporting, and litigation reporting purposes — the normative exclusions are stated in RES-5.2 and Section 21.5 — and are reported separately because their evidentiary weight differs. Distinguishable from contemporaneous receipts in all export packages and signal computations (flags field, Annex B.6).
A.22	PRF	**Pseudorandom Function.** A deterministic function (HMAC-SHA256 in Protocol Profile 1.0) used to determine whether a given request falls within the attestation sample. Operates on request commitments, not raw content.
A.23	Protocol Profile	A registered implementation specification defining cryptographic constructions, envelope schemas, key derivation methods, and receipt formats that implement this standard. Multiple profiles may coexist. Conformance requires exactly one declared profile per deployment; registration requirements are level-dependent (Section 22.6 — Level 1 may state No Profile, Levels 1–2 may use a validly self-declared profile, Levels 3–4 require a registered profile). See Annex B for Protocol Profile 1.0 summary.

Term ID	Term	Definition
A.24	RATS	**Remote ATtestation procedureS.** IETF architecture for remote attestation (RFC 9334). OVERT attestation architecture is complementary to RATS, with roles mapping to Attester (arbiter), Verifier (notary), and Relying Party (auditor/insurer).
A.25	Receipt	A cryptographically signed record issued by the notary service proving that a specific attestation envelope was received, validated, and recorded during a specific epoch. The base receipt contains 9 fields including attestation_hash, epoch binding, binary identity, network state, flags (temporality: contemporaneous, DELAYED_NOTARY, RECONSTRUCTED), and transparency log proofs; for cross-boundary workflows the registered Protocol Profile defines an extended receipt type additionally carrying the Section 4.8 parent-reference fields. Each receipt type is a closed schema. See Annex B for architecture; full field-level schemas are in the Protocol Profile.
A.26	S3P	**Statistical Safety Signal Protocol.** The normative auditor-reproducible sampling and measurement method defined in Section 9 (MEASURE). Uses commit-then-reveal epoch nonces, keyed-function-based sampling (HMAC in Protocol Profile 1.0), and Clopper-Pearson exact confidence intervals to produce statistically rigorous, evaluator-judged safety signals.
A.27	Severity Class	A classification of governance violations by severity, defined in GOV-3.2. Maps to risk-signal computation and response escalation requirements.
A.28	Split-Knowledge Key Hierarchy	The key management architecture in which content-binding keys (operator-managed, e.g., tenant_pepper) and sampling/identity keys (platform-managed, e.g., sampling_key, epoch_secret) are held by different parties. Ensures that routine audit is a zero-content-knowledge operation.
A.29	SPKI	**Subject Public Key Info.** DER-encoded public key information used for TLS certificate pinning in NETATT.
A.30	STH	**Signed Tree Head.** A signed commitment to the current state of the transparency log Merkle tree, enabling split-view detection.
A.31	SUT	**System Under Test.** The AI system being governed. The attestation system treats the SUT as untrusted — self-reports from the SUT are insufficient for AAL-4 conformance. This designation is specific to the OVERT attestation relationship and is distinct from the NIST SP 800-207 Zero Trust Architecture for network security.
A.32	TEVV	**Test, Evaluation, Verification, and Validation.** The systematic process of evaluating AI system performance, safety, and governance compliance. OVERT provides the attestation infrastructure for TEVV activities.
A.33	Transparency Log	An append-only, cryptographically verifiable log (per RFC 6962) in which attestation receipts and S3P attestations are recorded. Supports inclusion proofs

Term ID	Term	Definition
		(proving a receipt is in the log), consistency proofs (proving the log has not been tampered with), and signed tree heads for split-view detection.
A.34	Notary Network	One or more notary nodes operated under a published governance model (ATT-5.1) that validate attestations on behalf of relying parties. Where multiple nodes are deployed, t-of-n agreement is required before a valid receipt can be issued and no single node can unilaterally issue or suppress a receipt. A single structurally independent notary satisfies the AAL-4 independence requirement but not the multi-party resilience property — which is not required for AAL-4 (Section 4.1.1); single-IAP deployments disclose this limitation per Section 4.7.1 and the conformance statement grammar (Section 22.4). Multi-entity sets provide both independence and resilience; a multi-node set operated by a single entity remains `Single-IAP` (ATT-5.1(a)). The signature or verification construction achieving the t-of-n property (where applicable) is specified in the registered Protocol Profile.
A.35	Scanner	A runtime monitoring sidecar or component that inspects inputs, outputs, and intermediate states of an AI system to detect policy violations, security threats, or behavioral drift. It works in conjunction with the arbiter.
A.36	Local Classifier	A local evaluation component that runs classification or inference models to categorize inputs, outputs, or agent behaviors for policy decision-making.
A.37	Proof of Possession (PoP)	A challenge–response operation by which an operator demonstrates, without egressing protected content, that it retains the evidence payload bound to a given `evidence_commitment` at the time of challenge. Defined in Annex G, Section G.1.3.
A.38	OVERT Discovery Document	A machine-readable document published under the `/.well-known/` URI namespace (RFC 8615) that enumerates the endpoints from which an auditor retrieves the cryptographic artifacts required for independent verification. Defined in Annex G, Section G.3.

Annex B: Protocol Profile Reference Summary

> INFORMATIVE — *Protocol Profile 1.0 is the initial registered profile authored by GLACIS Technologies. Additional profiles may be submitted by any party meeting the registration requirements defined in Section 22.6. This annex summarizes Protocol Profile 1.0 for reference. The authoritative specification is the Protocol Profile document itself.*

B.1 Cryptographic Primitives

Protocol Profile 1.0 specifies the following cryptographic constructions:

Primitive	Usage	Specification
SHA-256	All digests, binary hashes, commitments	FIPS 180-4
HMAC-SHA256	PRF tags, request/evidence commitments, bearer tokens, S3P sampling	RFC 2104
HKDF-SHA256	Key derivation for both operator and platform key hierarchies	RFC 5869
Ed25519	Arbiter signatures, notary signatures, controller signatures	RFC 8032
Notary signature (BLS threshold in Profile 1.0)	Notary network t-of-n verification	draft-irtf-cfrg-bls-signature (construction selected by Protocol Profile 1.0; the core standard requires only the t-of-n trust property and permits alternative constructions including multi-signature schemes)
Deterministic encoding (CBOR in Profile 1.0, JCS for JSON profiles)	Canonical encoding of attestation structures	RFC 8949, Section 4.2 (CBOR); RFC 8785 (JCS)

Primitive	Usage	Specification
Merkle trees	Transparency log inclusion and consistency proofs	RFC 6962
Clopper-Pearson	Exact binomial confidence intervals for S3P	Standard statistical method

> NON-NORMATIVE NOTE ON POST-QUANTUM CRYPTOGRAPHY: *A post-quantum migration path for the signature layer using ML-DSA-65 (FIPS 204) is under development. This is a forward-looking implementation note only and does not alter the normative requirements or conformance level definitions of this version.*

Post-quantum migration path: Protocol Profile 1.0 uses BLS threshold signatures and Ed25519 single-signer signatures, both of which rely on the computational Diffie-Hellman problem and are vulnerable to quantum attack via Shor's algorithm. Section 18 requires that after January 1, 2031, conformant implementations use hybrid classical + post-quantum constructions or pure post-quantum constructions. For single-signer operations, the recommended migration is ML-DSA (FIPS 204) + Ed25519. For notary network operations, Protocol Profiles using multi-signature constructions can migrate each notary independently to ML-DSA; profiles using threshold signatures require threshold-compatible post-quantum schemes. Pure classical signature schemes become non-conformant after that date; this is OVERT's own profile-binding cutoff, informed by the NIST IR 8547 (draft) transition timeline, which deprecates quantum-vulnerable classical signatures after 2030 and disallows them after 2035. Protocol Profile 1.0's classical constructions (BLS threshold, Ed25519) are accordingly valid only until that cutoff; the hybrid or post-quantum replacements are specified by a revised or successor profile version registered under Section 22.6 before the cutoff, together with a published transition test plan (Section 2.2 note).

IETF RATS alignment and EAT forward reference: OVERT attestation envelopes are structurally aligned with the IETF RATS architecture (RFC 9334). The Entity Attestation Token (EAT, RFC 9711) defines a CBOR/JSON token format for attester-generated claims about entity identity and state. Future Protocol Profile revisions should evaluate expressing OVERT attestation envelopes as EAT profiles, enabling interoperability with the broader RATS ecosystem. In particular, an Agentic AI EAT Capability Attestation profile — encoding arbiter binary identity, co-epoch binding, and capability-scoped policy claims as EAT claims — would position OVERT attestation artifacts for direct consumption by EAT-aware verifiers and relying parties within the IETF trust model. This alignment is a design goal for Protocol Profile 2.0 and does not affect Protocol Profile 1.0 conformance.

SCITT alignment (informative). OVERT transparency logs implement the same pattern the IETF SCITT (Supply Chain Integrity, Transparency and Trust) architecture generalizes: an append-only, independently auditable log of signed statements. A registered Protocol Profile MAY operate an OVERT transparency log as a SCITT transparency service, registering receipts as signed statements and exposing SCITT receipts as inclusion evidence; this is targeted for Protocol Profile 2.0 alongside the EAT alignment above. Relatedly, the build-pipeline-compromise threat (Section 4.5) is addressed in the supply-chain ecosystem by in-toto and SLSA provenance attestations, which a deployment MAY bind to the arbiter binary identity verified under ATT-2.2.

B.2 Domain Separation and Key Architecture

Protocol Profile 1.0 uses versioned domain separation prefixes on all HMAC operations to prevent cross-protocol attacks. Each HMAC operation — request commitment, evidence commitment, sampling PRF, epoch bearer token, and S3P sampling — uses a distinct prefix with a version suffix enabling future protocol evolution.

The split-knowledge key hierarchy ensures that content-binding keys (operator-managed) and sampling/identity keys (platform-managed) are held by different parties. This separation prevents any single party from both reversing content AND verifying sampling fairness. The specific prefix strings, salt values, and key derivation parameters are specified in the Protocol Profile.

B.3 Canonicalization

Protocol Profile 1.0 canonicalizes all OVERT messages per RFC 8949 Section 4.2 (Deterministic Encoding). Two conformant encoders encoding the same logical data must produce identical byte sequences, which is the prerequisite for all hash-based verification. The standard requires deterministic canonicalization as a property (Section 17.1); the specific encoding format is specified by the applicable Protocol Profile. Protocol Profiles using JSON are expected to specify JCS (RFC 8785) for deterministic canonicalization; Protocol Profiles using CBOR are expected to specify RFC 8949 Section 4.2.

Protocol Profile 1.0 prohibits IEEE-754 floating-point numbers in attestation envelopes, requiring scaled integers for deterministic cross-platform hashing. The S3P schema uses decimal strings for rates and bounds. Timestamps use uint64 nanoseconds since Unix epoch. Indefinite-length encodings, NaN, and +/-Inf are rejected. Protocol Profiles using JSON encodings are expected to specify

equivalent numeric safety requirements (e.g., string-encoded decimals, integer-only numeric fields, or explicit precision annotations) to satisfy the numeric losslessness property of Section 17.1.1.

B.4 Commitment Architecture

Protocol Profile 1.0 defines a layered commitment architecture:

- **Request commitments** are computed by HMAC over the content digest using an operator-managed key derived from the operator's root secret via HKDF. The content digest stays local; only the HMAC commitment crosses the trust boundary.
- **Evidence commitments** follow the same pattern for policy evaluation evidence.
- **PRF sampling tags** are computed using a platform-managed key, operating on the request commitment (not the raw content digest). This ensures auditors can verify sampling fairness without holding content-reversing keys.

The specific HMAC constructions, HKDF derivation parameters, and key hierarchy are specified in the Protocol Profile.

Critical constraint: Operator-managed content-binding keys never leave the operator's environment. Deep audits requiring content verification are conducted on-premises under operator control and legal authority.

B.5 Key Hierarchy

The split-knowledge key hierarchy has two branches:

Operator-managed keys (content binding): Derived from an HSM-backed root secret. Includes keys for content commitment and local storage encryption. These keys never cross the operator's trust boundary.

Platform-managed keys (identity and sampling): Derived from an HSM-backed root secret managed by the notary network operator. Includes keys for sampling, epoch management, and notary signing. In Protocol Profile 1.0, notary signing keys are BLS threshold shares distributed across notary nodes; other Protocol Profiles may use per-notary signing keys with independent key management.

Forward secrecy: Epoch-scoped keys are deleted after the subsequent epoch begins. Compromise of a current epoch secret does not reveal past epoch secrets.

Recovery: Shamir k-of-n (recommended: 3-of-5) across geographic regions or HSMs for operator root secrets. Platform key recovery is specified in the registered Protocol Profile (Protocol Profile 1.0

uses BLS threshold key shares across notary nodes; multi-signature profiles use standard per-node key backup procedures).

The specific HKDF derivation paths, salt values, and key tree structure are specified in the Protocol Profile.

B.6 Attestation Envelope Architecture

Protocol Profile 1.0 defines three closed-schema structures:

Base Envelope (all requests — 9 fields): Emitted for every in-scope AI action. Contains a blinded identifier, request commitment, encoder binary identity, non-content metadata, monotonic counter, nanosecond timestamp, key identifier, arbiter instance identifier, and signature. No additional fields are permitted (closed schema).

Extended Envelope (sampled requests — 10 fields): Emitted for requests selected by the sampling PRF. Contains a reference to the matching Base Envelope, request and evidence commitments, the full PRF tag for auditor recomputation, policy evaluation scores, monotonic counter, timestamp, key and arbiter identifiers, and signature. Closed schema.

Receipt (9 fields, issued by notary service): Contains the attestation hash (cryptographic digest of the submitted envelope), validated epoch, notary-derived binary hash, network state hash, monotonic counter, issuance timestamp, flags (temporality: contemporaneous, DELAYED_NOTARY, RECONSTRUCTED — Section 21.5), notary signature (single-signer, multi-signature, or threshold, as specified in the Protocol Profile), and transparency log proofs (inclusion proof, consistency proof, signed tree heads). Closed schema. A receipt's `attestation_id`, as referenced in Section 4.8 and Annex G, is its attestation-hash field under the name reflecting its role as a cross-boundary reference key. For cross-boundary workflows, the Protocol Profile defines an **extended receipt type** that additionally carries the Section 4.8 parent-reference fields (`parent_attestation_id`, `parent_reference_status`); each receipt type is itself a closed schema.

The `flags` field distinguishes contemporaneous receipts (`0x00`) from the two late-attestation classes of Section 21.5: bit 0 (`0x01`, `DELAYED_NOTARY`) marks a receipt whose envelope was generated during the governed event but notary-attested in a subsequent epoch; bit 1 (`0x02`, `RECONSTRUCTED`) marks a receipt generated retroactively from local fail-open records after attestation infrastructure resumed (RES-5.2). Auditors and risk-signal computations filter on this field; neither late class counts toward contemporaneous coverage, and the two classes are reported separately because their evidentiary weight differs (a contemporaneous local receipt upgraded late versus a record reconstructed after the fact).

The complete field-by-field schemas with types, constraints, and signature scopes are specified in the Protocol Profile.

B.7 S3P Attestation Schema

The S3P attestation schema is a 16-field closed structure capturing all data needed for auditor-reproducible verification of the evaluator-judged violation rate. Every field is necessary and sufficient for independent recomputation of statistical bounds.

The schema fields are: (1) epoch identifier; (2) violation type; (3) total request count; (4) sampled request count; (5) sampling rate (decimal string to avoid IEEE-754 variance); (6) violation count (`n_violations`); (7) observed violation rate (decimal string); (8) confidence level; (9) bound form (`bound_form` : `one-sided-upper` or `two-sided` ; Section 19.3); (10) lower confidence bound (decimal string; `0` where the bound form is one-sided-upper); (11) upper confidence bound (decimal string); (12) denominator class (`denominator_class` ; Section 4.6); (13) sampling threshold; (14) epoch nonce commitment; (15) status indicator; (16) notary signature.

The three status values are: `"OK"` (valid computation), `"ERR_INSUFFICIENT_SAMPLE"` (sample size below minimum), and `"ERR_NONCE_NOT_PUBLISHED"` (verification failure — epoch nonce was not published after epoch close).

The complete schema with field types and encoding rules is specified in the Protocol Profile.

B.8 Clopper-Pearson Confidence Interval Computation

The Clopper-Pearson method provides exact (not approximate) binomial confidence intervals with guaranteed coverage probability. It provides exact binomial interval coverage under the S3P sampling model and remains valid for small sample sizes without normal-approximation assumptions. The upper bound is conservative by construction.

Given `k` violations observed in `n` sampled requests at confidence level `1 - alpha` :

```
CI_lower = Beta_inv(alpha/2;     k,     n - k + 1)     for k > 0, else 0
CI_upper = Beta_inv(1 - alpha/2; k + 1, n - k)         for k < n, else 1
```

Where `Beta_inv(p; a, b)` denotes the p-th quantile of the Beta distribution with shape parameters `a` and `b` .

The two-sided interval above is the reporting form. For an upper-bound safety claim — "the violation rate did not exceed `p_bound` at confidence `1 - alpha`" — the governing computation is the **one-sided upper bound**, which places the full error probability in the upper tail:

```
CI_upper(one-sided) = Beta_inv(1 - alpha; k + 1, n - k)   for k < n, else 1
```

(equivalently, for `k = 0`: the smallest `n` satisfying `(1 - p_bound)^n ⩽ alpha`). The sample-size table in Section 19.7.1 corresponds to this one-sided form; an S3P attestation declares which form its bounds carry in the `bound_form` field (Annex B.7, Section 19.3). Computing an upper-bound claim with the two-sided `alpha/2` form requires the larger sample sizes noted in Section 19.7.1 (e.g., 368 rather than 299 at 95% / 1%).

Properties:

- Exact coverage: P(p_true in [CI_lower, CI_upper]) >= 1 - alpha for all p_true
- Conservative: The interval is wider than approximate methods (Wald, Wilson), never narrower
- Valid for all sample sizes, including small samples
- No normal-approximation assumptions; exactness holds under the S3P Bernoulli sampling model with binary evaluator verdicts

B.9 Receipt Service Architecture

The receipt service accepts a closed-schema request containing only a hash and an epoch identifier, and returns a signed receipt. The API schema enforces the non-egress architecture at the protocol level: the service is structurally incapable of receiving content because its schema does not contain fields for content. Unknown fields are rejected.

This constraint is architectural, not merely a validation rule. The receipt service API schema is specified in the Protocol Profile.

B.10 Informative Latency Targets

The following latency targets are informative recommendations for Protocol Profile 1.0. Specific latency requirements are deployment-dependent and are not normative requirements of the standard.

Phase	Operation	Informative Target
Phase 1 — Enforcement	Local policy evaluation	< 5 ms P50
Phase 1 — Enforcement	Distributed policy evaluation	< 25 ms P50
Phase 2 — Attestation	Receipt round-trip	< 50 ms P50
Phase 3 — Commitment	Transparency log inclusion	< 100 ms P95

Total overhead (enforcement + attestation): informative target < 50 ms P50, which is negligible relative to typical LLM inference latency (500-5000 ms).

B.11 Informative Default Parameters

The following default parameters are informative recommendations for Protocol Profile 1.0. Operators configure these values according to their deployment requirements.

Parameter	Informative Default	Notes
Epoch duration	300 seconds (5 minutes)	Configurable per deployment policy
Tool-call recursion depth	25	Configurable threshold defined in deployment policy
Clock skew tolerance	<= 2 seconds	Bounded skew tolerance; stale submissions rejected
Override review SLA	Within operator-defined SLA	Recommended: 24 hours
TEVV testing interval	Per operator's risk management policy	Not to exceed 12 months or as required by applicable regulation

B.12 Implementation Resources

Protocol Profile 1.0 includes CBOR diagnostic notation examples, reference test vectors for S3P computation, and auditor verification procedures. These materials enable implementers to validate their implementations against known-good results. Protocol Profiles using other encodings are expected to provide equivalent notation examples and test vectors in their respective formats.

Reference test vectors and implementation examples are available in the Protocol Profile document. Organizations implementing OVERT using Protocol Profile 1.0 should obtain the Protocol Profile from the OVERT Protocol Profile Registry.

Annex C: Design Rationale and Case Studies

> INFORMATIVE — *This annex provides design rationale and contextual analysis. It describes legal, operational, and institutional conditions relevant to the standard's development. It does not impose requirements on implementers or assert legal conclusions. The normative requirements of the standard are specified in Parts 1-5. It is structured as Design Decision, Rationale, and Supporting Analysis.*

C.1 Verification Gaps in High-Stakes AI Deployments

Design Decision: OVERT requires independent, third-party verifiable attestation (AAL-4) for governance controls in high-stakes deployments, rather than relying on self-attestation or contractual governance alone.

Rationale: Contractual governance has proven structurally insufficient as the sole enforcement mechanism for AI safety controls. When disputes arise between AI system providers and their customers over safety control execution, neither party can independently verify what controls actually ran if no attestation infrastructure exists.

Supporting Analysis: In early 2026, a series of disputes between major AI laboratories and government agencies demonstrated this proof gap with extraordinary clarity. In one instance, an AI company insisted on contractual red lines regarding prohibited uses, while the government customer demanded unrestricted operational access. Neither party could independently verify whether AI use complied with stated restrictions during operational deployment. The dispute was adjudicated through contract negotiations, leaked internal memoranda, public conference statements, and executive action — rather than through independent verification of actual system behavior.

Simultaneously, competing AI laboratories publicly accused each other of inadequate safety practices, with characterizations ranging from "safety theater" to "mendacious" claims about governance controls. These mutual accusations could not be independently adjudicated because no party had deployed infrastructure capable of producing verifiable records of what safety controls actually

executed on any given interaction. The disputes were resolved — or remain unresolved — through political, commercial, and reputational channels rather than through technical verification.

This pattern illustrates a structural problem: contractual governance produces assertions about intended behavior, not verifiable records of actual behavior. When the only evidence of safety control execution is the operator's own claims, disputes become contests of credibility rather than questions of fact. If verification technology becomes commercially deployable at scale, the continued reliance on unverifiable self-attestation may become relevant to courts, regulators, and insurers evaluating evidentiary and governance posture under applicable legal and policy frameworks.

OVERT addresses this gap by specifying how to produce tamper-evident, independently verifiable, temporally bound proof that AI governance controls executed — without exposing protected content.

C.2 The T.J. Hooper Principle and Potential Standard-of-Care Analysis

Design Decision: OVERT is designed as an open standard that can serve as one reference point in discussions of verifiable AI governance, recognizing that courts — not industries — ultimately determine the standard of care.

Rationale: The T.J. Hooper principle holds that an entire industry can be found negligent for failing to adopt available safety technology, regardless of industry custom.

Supporting Analysis: In The T.J. Hooper, 60 F.2d 737 (2d Cir. 1932), Judge Learned Hand held that tugboat operators were negligent for failing to carry radio receivers that would have warned of an approaching storm — even though no tugboat company used radios at the time. The court stated: "a whole calling may have unduly lagged in the adoption of new and available devices… Courts must in the end say what is required; there are precautions so imperative that even their universal disregard will not excuse their omission."

The principle has been applied consistently for nearly a century. The RAND Corporation's report on AI tort liability explicitly cited T.J. Hooper, noting that "courts can still find AI companies negligent even if they did follow industry custom" and that safety-conscious companies developing standards "could establish benchmarks for the whole industry in future litigation."

For AI governance, the implication is narrower. If cryptographic attestation technology becomes commercially deployable and operationally mature, failure to consider its adoption could become relevant to negligence analysis, depending on jurisdiction, commercial availability, deployment maturity, and the surrounding facts. An open standard may strengthen that analysis by documenting

an interoperable approach, but publication of a standard alone does not establish that the technology is available, required, or legally obligatory.

The English equivalent is Bolitho v. City and Hackney Health Authority [1998] AC 232, where Lord Browne-Wilkinson held that courts may reject professional custom as unreasonable if "the professional opinion is not capable of withstanding logical analysis." The Australian statutory calculus under Section 5B of the Civil Liability Act 2002 (NSW) reaches the same outcome through explicit consideration of the probability of harm, seriousness of harm, burden of precautions, and social utility. The German doctrine of Verkehrssicherungspflichten requires anyone who creates or controls a potential source of danger to take necessary precautions.

C.3 Adverse Inference Doctrine and the Duty to Create Records

Design Decision: OVERT Section 21 (Legal Preservation and Production) requires retention policies, legal hold procedures, immutable export capabilities, and chain-of-custody metadata — addressing the risk that operators could "define away bad evidence" through self-serving retention policies.

Rationale: The adverse inference doctrine permits factfinders to draw unfavorable conclusions when a party fails to preserve records it had the available technology to produce. For AI systems, where the "black box" nature makes governance documentation critical, the absence of attestation technology may be relevant to evidentiary analysis. Whether the failure to deploy attestation creates a substantive claim or an evidentiary disadvantage depends on jurisdiction, applicable duty, commercial availability of attestation technology, and the specific facts. This rationale identifies the doctrinal relevance; it does not assert a specific legal outcome.

Supporting Analysis: Under FRCP 37(e), if electronically stored information that should have been preserved is lost because a party failed to take reasonable steps to preserve it, the court may order measures no greater than necessary to cure the prejudice. Upon finding that a party acted with the intent to deprive another party of the information's use in the litigation, the court may presume that the lost information was unfavorable to the party, instruct the jury that it may or must presume the information was unfavorable, or dismiss the action or enter a default judgment.

In Zubulake v. UBS Warburg, 229 F.R.D. 422 (S.D.N.Y. 2004), failure to preserve digital evidence resulted in a $29.2 million verdict including $21.1 million in punitive damages. The court granted an adverse inference instruction: "if you find that UBS could have produced this evidence… you are permitted, but not required, to infer that the evidence would have been unfavorable to UBS."

Valcin v. Public Health Trust of Dade County, 473 So.2d 1297 (Fla. 3d DCA 1984), provides the closest doctrinal analogue. Where a hospital's file failed to contain an operative note, the court imposed a rebuttable presumption of negligence and shifted the burden of proof for records that should have been created pursuant to a duty. If industry standards (NIST AI RMF, ISO/IEC 42001) and available technology create a de facto duty to record AI safety control execution, the failure to deploy attestation technology triggers a parallel adverse presumption.

OVERT Section 21 directly mitigates this risk by requiring operators to define retention policies, implement legal hold procedures, and maintain export capabilities — ensuring that attestation artifacts are available when needed for legal proceedings, regulatory investigations, or insurance claims.

C.4 Consent Attestation and Healthcare AI

Design Decision: OVERT HITL-1 requires cryptographic attestation of patient consent in healthcare AI deployments, with consent receipts that are independently verifiable.

Rationale: AI systems that generate their own compliance records without actual human attestation create a novel and dangerous category of false documentation.

Supporting Analysis: Recent class-action litigation regarding an ambient clinical documentation system deployed without all-party consent illustrates this risk. The complaint alleged violations of the California Invasion of Privacy Act (CIPA) and the Confidentiality of Medical Information Act (CMIA).

The most significant allegation: the AI tool allegedly inserted false statements into patient charts claiming patients "were advised" and "consented" to recording when they had not. This represents AI systems generating their own false compliance documentation — a pattern that conventional audit methods (reviewing the documentation itself) cannot detect, since the documentation asserts the very compliance whose absence it conceals.

Estimates suggest 100,000+ patient encounters may have been affected. The legal theories — wiretapping, unauthorized third-party disclosure, false consent documentation, retention failures — represent patterns emerging across healthcare AI deployments.

OVERT consent attestation addresses this by requiring that consent events be cryptographically attested with independent verification: the consent receipt is signed by the notary network, not generated by the AI system itself. The receipt proves that a consent interaction occurred at a specific time, was recorded through a specified mechanism, and was attested by an independent party.

This approach also responds to the California Invasion of Privacy Act's requirement for "all party" consent to recording, as well as SB 53's incident reporting requirements effective January 1, 2026.

C.5 Multi-Agent Trust Exploitation

Design Decision: OVERT Sections 11-16 (Agentic AI Controls) require per-call attestation, capability-based access control, and multi-agent trust boundary enforcement.

Rationale: Multi-agent AI systems exhibit systematic vulnerability to trust exploitation through prompt injection, tool misuse, and cross-agent privilege escalation.

Supporting Analysis: In a 2025 evaluation of 17 state-of-the-art models, 82.4% executed malicious commands when requested by peer agents — even where they resisted the identical instruction delivered directly — demonstrating inter-agent trust exploitation as the dominant agentic attack surface, alongside prompt injection through shared context, capability escalation via delegated tool access, and information exfiltration through inter-agent communication channels (The Dark Side of LLMs: Agent-based Attacks for Complete Computer Takeover, arXiv:2507.06850, 2025). The CaMeL framework (Google DeepMind, 2025) proposed capability-based prompt injection defense through separation of privileged and quarantined execution contexts — a design pattern that OVERT formalizes through per-call attestation and capability-scoped access control.

The "policy-quality gap" is particularly acute in multi-agent systems: an attestation system that faithfully records and attests to the outputs of a compromised agent produces cryptographically valid records of invalid outputs. OVERT addresses this through capability-based access control (CAP-1, CAP-2) that constrains what each agent can do, combined with per-call attestation (TOOL-1 through TOOL-5) that records what each agent actually did. The combination enables forensic reconstruction of multi-agent interactions and detection of capability violations even when individual agent outputs are compromised.

C.6 Tiered Certification Analogy

Design Decision: OVERT is structured as a tiered standard (AAL-1 through AAL-4) with progressive requirements, permitting organizations to adopt attestation incrementally while establishing a clear ceiling (AAL-4) for the highest assurance tier.

Rationale: Tiered certification avoids all-or-nothing adoption barriers and creates a progressive path toward comprehensive governance. Building-certification systems provide a useful structural analogue.

Supporting Analysis: The relevant lesson from tiered certification systems is structural, not economic: progressive assurance levels can lower adoption friction, and separation between a standard-setter and an independent verifier can improve trust in published claims. For OVERT, the

relevant point is narrower: tiered adoption and separation between standard-setting and independent verification can accelerate uptake without changing the standard's underlying technical claims. External regulatory, contractual, or market incentives may influence adoption, but they do not alter what OVERT itself proves.

Cautionary lessons: Tiered systems can incentivize point gaming or over-interpretation of lower-tier certifications. For AI governance, this informed OVERT's emphasis on distinguishing documentation, operator-controlled telemetry, and independently verifiable attestation rather than treating all conformance levels as equivalent.

C.7 PCI-DSS Contractual Adoption Precedent

Design Decision: OVERT includes a signal architecture and independent-verification model that can be incorporated into contractual and oversight processes, paralleling the way PCI-DSS was operationalized through private agreements.

Rationale: Standards adoption often accelerates when contractual incentives align with verification requirements.

Supporting Analysis: PCI-DSS illustrates that private agreements can embed verification expectations into commercial relationships without waiting for legislation. The relevant point for OVERT is that procurement, platform, insurance, or sector-specific contracts may reference verifiable governance evidence and independent verification artifacts. OVERT is designed to be referenceable in those settings, but it does not prescribe a particular market structure, assessment industry, or contractual model.

C.8 FedRAMP and NIST SP 800-53 Adoption History

Design Decision: OVERT provides crosswalks to NIST SP 800-53 Rev 5 and FedRAMP (in the companion document `OVERT_v1.1_CROSSWALKS.md`) and supports OSCAL-formatted attestation packs.

Rationale: Federal adoption of security standards follows established patterns through NIST framework alignment, FedRAMP authorization, and OSCAL-based automation.

Supporting Analysis: Federal security standards often spread through crosswalks, machine-readable artifacts, and reuse in existing compliance workflows. The relevant point for OVERT is interoperability: by mapping to NIST/FedRAMP concepts and supporting OSCAL-compatible out-

puts, implementers can present attestation evidence in familiar oversight formats. These references explain integration paths, not adoption forecasts or official endorsement.

C.9 Insurance Market Interpretation

Design Decision: OVERT Section 4.6 (Risk Signal Architecture) and Annex D (Risk Signal Framework) are primary design pillars, not afterthoughts.

Rationale: Insurance market reactions illustrate that external risk bearers may seek more verifiable runtime evidence for AI systems. Those reactions are informative context; they do not determine the scope or legal effect of this standard.

Supporting Analysis: Insurance markets have begun issuing both exclusions and affirmative products addressing AI risk. Those developments are relevant here only as evidence that some external risk bearers are beginning to differentiate among AI governance postures; they do not imply insurer endorsement of OVERT, any required coverage position, or any specific underwriting outcome.

These developments show why independently verifiable runtime evidence may become relevant to external risk assessment. OVERT provides a signal and evidence architecture that can be evaluated in that context, but it does not determine coverage availability, pricing, or legal entitlement.

C.10 Non-Egress Architecture and Business Associate Agreement Exposure

> INFORMATIVE — *This section describes architectural properties relevant to regulatory analysis. It does not constitute legal advice. Organizations should obtain qualified legal counsel regarding data processing agreements and BAA requirements for their specific deployments.*

Design Decision: Section 17.5 states that the non-egress architecture "SHOULD be designed to prevent the transmission of Protected Health Information (PHI) or other regulated content" while explicitly hedging: "The applicability of data processing agreements or Business Associate Agreements remains a question of applicable law and regulatory interpretation."

Rationale: The architectural argument for reduced BAA exposure is strong but the legal conclusion is not yet settled. OVERT preserves the argument without overclaiming.

Supporting Analysis: Under HIPAA, a Business Associate is any person or entity that "creates, receives, maintains, or transmits" PHI on behalf of a covered entity (45 CFR §160.103). The OVERT non-egress architecture is specifically designed so that the attestation layer — including the receipt service, notary network, and transparency log — never receives PHI. Only cryptographic commitments (HMAC-SHA256 digests with tenant-scoped keys) cross the operator's trust boundary. The raw content remains in the operator's content-addressable storage, never leaving the covered entity's environment.

The architectural claim is: if the attestation provider never receives, creates, maintains, or transmits PHI — receiving only irreversible cryptographic commitments from which PHI cannot be reconstructed — the attestation provider may not meet the statutory definition of a Business Associate. This may reduce the factual basis for treating the attestation provider as a recipient of PHI, but BAA obligations remain a matter of applicable law and deployment-specific facts.

However, OCR has not issued guidance specifically addressing whether receipt of cryptographic commitments derived from PHI constitutes "receiving" PHI. The closest analogue is the de-identification safe harbor (45 CFR §164.514(b)), which permits disclosure of health information from which specified identifiers have been removed. HMAC commitments are arguably stronger than de-identification: they are computationally irreversible without the tenant-scoped key, which the attestation provider never possesses.

The hedge in Section 17.5 reflects the current state: the architectural argument is sound, the legal conclusion requires either OCR guidance or judicial interpretation, and the standard should not assert a legal conclusion that applicable law has not yet confirmed. Healthcare operators should consult qualified HIPAA counsel regarding their specific deployment architecture.

C.11 Emergent Behavior in Authorized Agentic Systems

Design Decision: OVERT Section 16 (Behavioral Drift Governance) introduces five control families (DRIFT-1 through DRIFT-5) addressing emergent behavioral changes in agentic AI systems that occur entirely within authorized operational bounds.

Rationale: Existing governance frameworks — including earlier per-action runtime control models — are designed to detect and prevent policy violations: individual actions that breach a defined rule. Agentic AI systems introduce a qualitatively different governance challenge: emergent behavior

where every individual control passes but the system's aggregate behavior drifts, cascades, or produces ungovernable complexity. This gap cannot be closed by tightening existing controls; it requires a new category of governance capability.

Supporting Analysis: Per-action attestation — the foundational model used by earlier runtime-governance designs and comparable frameworks — operates on a premise inherited from conventional access control: that governance is the sum of individual authorization decisions. This premise holds for request-response systems where each invocation is independent. It does not hold for agentic AI systems, where persistent agents accumulate state, spawn subordinate agents, and operate across extended time horizons. Six illustrative scenarios demonstrate the structural inadequacy of per-action governance for agentic deployments.

Spawn chain complexity. An orchestrator agent, operating within its declared capability set, spawns sub-agents to decompose a complex task. Each sub-agent, also operating within its declared capability set, spawns further sub-agents. Every individual spawn decision is authorized under the system's capability policy. The resulting execution graph, however, may comprise dozens or hundreds of leaf agents operating in parallel, each issuing tool calls, consuming resources, and producing outputs that feed into sibling and parent agents. The aggregate topology — the total number of active agents, the depth of the spawn hierarchy, the fan-out at each level — may far exceed what any human operator anticipated or any governance process was designed to oversee. Existing controls such as MULTI-2 attest the topology of agent hierarchies but do not evaluate whether the observed topology complexity exceeds the deployment's declared operational baseline.

Within-bounds behavioral drift. An agent produces outputs that individually conform to all applicable policy constraints across successive operational epochs. No single output is flagged, rejected, or escalated. Over time, however, the statistical distribution of those outputs shifts: risk scores trend higher or lower, topic coverage narrows, tool selection patterns change. The shift may be gradual enough that no individual epoch-to-epoch comparison triggers concern, yet the cumulative drift from the system's initial behavioral baseline is substantial. Measurement and evaluation controls such as MEA-2 assess whether individual outputs violate policy thresholds; they do not detect distributional shifts in the population of authorized outputs. Such drift may indicate model degradation, subtle prompt manipulation that biases rather than violates, or environmental changes that alter the agent's effective decision-making.

Cascading depth exploitation. Consider a three-level agent hierarchy where each level spawns three sub-agents. The resulting execution graph contains twenty-seven leaf agents. Each individual agent operates within its authorized bounds — its tool calls are permitted, its outputs conform to policy, its resource consumption falls within declared limits. But the combinatorial complexity of the full execution graph — the total volume of tool invocations, the interaction patterns between agents at different levels, the aggregate resource consumption, the effective attack surface — may be orders

of magnitude beyond the deployment's design assumptions. Existing recursion depth limits operate per-trace and do not evaluate the aggregate complexity of concurrent execution graphs sharing a common orchestrator.

Tool selection drift. An agent authorized to invoke multiple tools shifts its selection distribution over time. Where the agent previously selected one tool for approximately sixty percent of invocations and another for approximately forty percent, the ratio gradually inverts. Neither tool is prohibited; every individual invocation is authorized. The change in selection distribution, however, may indicate that the agent's underlying decision-making behavior has materially changed. Existing controls log individual tool invocations but do not track selection distributions across tools over time, and therefore cannot detect distributional shifts that leave every individual action compliant.

Propagated drift across agent hierarchies. When a parent agent drifts in the manner described above, its outputs — which serve as inputs to downstream agents — change in distribution. Child agents, whose models, policies, and configurations remain unchanged, alter their behavior in response to the changed input distribution. The behavioral drift propagates through the attestation DAG without any agent individually violating its policy. Existing controls evaluate each agent's behavior independently and do not correlate behavioral changes across parent-child attestation linkages, rendering propagated drift invisible to per-agent governance.

Human oversight quality degradation. Human reviewers responsible for overseeing AI outputs initially conduct substantive reviews: they spend adequate time, apply corrections at rates consistent with the system's risk signals, and demonstrate decision patterns that correlate with output characteristics. Over time — through automation bias, workload pressure, or miscalibrated trust — review duration decreases, modification rates decline, and the statistical correlation between risk signals and review decisions weakens. The review process continues to occur, and existing controls attest that it occurred, but the review ceases to be substantively meaningful. Approval velocity controls may cap the rate of approvals but do not assess whether the cognitive engagement underlying each approval is sufficient for the decision's risk level.

These six scenarios share a common structural feature: the governance gap lies not in any individual action but in the relationship between per-action compliance and system-level behavior. An attestation system that evaluates each action independently and finds no violation may nonetheless fail to detect that the system's aggregate behavior has materially changed — potentially in ways that alter its risk profile, undermine its fitness for purpose, or erode the effectiveness of human oversight. DRIFT-1 through DRIFT-5 close this gap by requiring that conformant systems declare their intended behavioral baseline (DRIFT-1), detect deviations from that baseline using sequential statistical methods (DRIFT-2), evaluate execution topology complexity against declared bounds (DRIFT-3), trace behavioral drift propagation across agent hierarchies (DRIFT-4), and assess the substantive quality of human oversight processes (DRIFT-5). The standard specifies what conformant systems

must detect and attest. The specific statistical methods, evaluation instruments, and enforcement mechanisms are specified in the registered Protocol Profile.

Annex D: Risk Signal Framework (Informative)

This annex describes the framework for OVERT risk signals. Signal definitions, mathematical formulas, derivation procedures, and minimum credibility thresholds are specified in the registered Protocol Profile or companion signal specification.

D.1 Signal Properties

All OVERT risk signals satisfy the properties specified in Section 4.6 (where the normative requirements are stated):

1. Content-free derivation
2. Verifiability classification
3. Temporal granularity
4. Statistical rigor
5. Scope binding

D.2 Signal Categories

OVERT risk signals are organized into three categories:

Category	Scope	Examples
Operational Signals	Attestation infrastructure health	Coverage ratios, exposure windows, response latency, retention integrity
Governance Risk Signals	Policy compliance indicators	Violation rate bounds, override frequency, consent coverage, review completion
Agentic Risk Signals	Agentic system behavioral indicators	Behavioral drift rate, graph complexity, spawn authorization, review quality

Agentic Risk Signals apply only to systems classified as "Automation" or "Agentic" under IDE-1.2 and are required for OVERT Agentic conformance.

D.3 Signal Derivation Requirements

Signal specifications in the registered Protocol Profile are required to include, for each signal (see Section 4.6):

- **Signal identifier** — unique, namespaced (e.g., OVERT-INS-NNN for insurance signals, or other prefixes as defined by companion specifications)
- **Definition** — precise natural-language description
- **Formula** — mathematical formula with defined numerator, denominator, and unit
- **Data type and unit** — including encoding requirements to avoid floating-point variance
- **Source artifacts** — which attestation artifacts are required for computation
- **Derivation procedure** — step-by-step auditor-reproducible computation method
- **Aggregation window** — temporal scope (epoch, daily, policy-period)
- **Missing-data handling** — behavior when source artifacts are unavailable
- **Minimum credibility threshold** — minimum sample size for statistically credible interpretation
- **Severity classification** — threshold-based severity levels

D.4 Design Rationale

Risk signals are a primary design goal of OVERT because the verification gap described in the Foreword affects defenders, auditors, regulators, procurement reviewers, and external risk assessors alike. Quantitative risk signals — independently verifiable where the denominator source supports it, operator-dependent where it does not (see Section 4.6) — enable:

- **Security operations:** Monitoring of coverage, overrides, exposure windows, and other runtime indicators within the attested scope
- **Audit and investigation:** Recomputable evidence for control-execution and anomaly analysis
- **Regulatory and oversight reporting:** Quantified posture reporting without content exposure
- **External risk analysis:** Structured inputs for insurance, procurement, or other third-party evaluations, subject to the verifiability classification of the underlying signals

Signal specifications are maintained in the Protocol Profile rather than this standard so that signal definitions can evolve with operational experience and measurement practice, without requiring standard revisions.

Annex E: Legal Admissibility Analysis (Informative)

> INFORMATIVE — *This annex does not constitute legal advice. Admissibility determinations are made by courts applying jurisdiction-specific rules. Organizations should consult qualified legal counsel regarding the admissibility of attestation artifacts in their jurisdictions.*

This annex analyzes how AAL-4 attestation artifacts produced by OVERT-conformant systems may relate to evidentiary rules governing the admissibility of electronic records. The discussion identifies how OVERT design features address foundational admissibility concepts — authenticity, integrity, chain of custody, and hearsay exceptions — without asserting that any specific attestation artifact will be admitted in any specific proceeding.

E.1 Federal Rules of Evidence 902(13): Certified Records of Regularly Conducted Activity (Electronic)

Rule: FRE 902(13), effective December 1, 2017, provides for self-authentication of "a record of a regularly conducted activity" in electronic form, when accompanied by a certification from a qualified person that the record: (A) was made at or near the time of the occurrence of the matters set forth by a person with knowledge, or from information transmitted by such a person; (B) was kept in the course of the regularly conducted activity; and (C) was made as a regular practice of that activity.

OVERT Mapping:

FRE 902(13) Requirement	OVERT Feature	Relevant Controls
"made at or near the time of the occurrence"	Attestation receipts include nanosecond-precision timestamps (wall_time_ns), co-epoch binding, and transparency log inclusion with signed tree heads providing independent temporal verification.	ATT-1, ATT-2, Section 18

FRE 902(13) Requirement	OVERT Feature	Relevant Controls
"by a person with knowledge, or from information transmitted by such a person"	OVERT records are machine-generated by the arbiter and notary network. Courts increasingly accept automated systems as sources of business records when the system's reliability is established. The notary network's independent derivation of binary identity and network state provides an additional reliability indicator.	ATT-2.2, ATT-3.3, ATT-5
"kept in the course of regularly conducted activity"	OVERT attestation is continuous and automatic — operating on every in-scope AI action as a regular practice, not created in anticipation of litigation. The transparency log provides a tamper-evident, append-only record.	ATT-4, Section 21.1
"made as a regular practice"	AAL-4 conformance requires continuous attestation for all in-scope actions. The DPL publishes request commitments per epoch as a regular operational practice.	Section 22 (Conformance)
Certification by qualified person	Section 21.3(e) (Custodian Certification) requires the export package to include custodian identity, timestamp, scope declaration, and hash of the export package. This certification can be prepared by the operator's designated custodian of records.	Section 21.3, 21.5

Analysis: OVERT attestation artifacts are designed to address the structural elements of FRE 902(13). Whether a specific court accepts these artifacts under FRE 902(13) will depend on the proponent's compliance with notice requirements (FRE 902(13) requires written notice and opportunity to inspect), the court's assessment of the underlying system's reliability, the proponent's ability to establish the "regular practice" and "person with knowledge" elements for machine-generated records, and the specific facts of the deployment. This analysis identifies design alignment; it does not predict admissibility outcomes.

E.2 Federal Rules of Evidence 902(14): Certified Data Copied from Electronic Device, Storage Medium, or File

Rule: FRE 902(14), also effective December 1, 2017, provides for self-authentication of data "copied from an electronic device, storage medium, or file" when accompanied by a certification from a qualified person that the process of digital identification used to verify the data is trustworthy, typically through cryptographic hash verification.

OVERT Mapping:

FRE 902(14) Requirement	OVERT Feature	Relevant Controls
Data "copied from" electronic device	OVERT immutable export packages (Section 21.3) are copied from the operator's content-addressable storage and the transparency log.	Section 21.3
Process of digital identification	SHA-256 hashing, HMAC commitments, Ed25519/BLS signatures, and Merkle tree inclusion proofs provide multiple layers of cryptographic verification.	ATT-1, Section 18, Annex B
Process "used to verify" is trustworthy	The entire OVERT verification chain is publicly specified, uses NIST-approved cryptographic primitives (SHA-256, HMAC-SHA256, HKDF), and is independently reproducible by any party.	Protocol Profile, Annex B
Certification by qualified person	Section 21.3(e) requires custodian certification with identity, timestamp, scope, and hash of the export package.	Section 21.3(e)

Analysis: FRE 902(14) was specifically designed to accommodate cryptographic hash verification of electronic data. OVERT attestation artifacts employ multiple layers of cryptographic integrity verification (content hashes, commitment chains, Merkle tree proofs, notary signatures) designed to address the requirements of 902(14) authentication. The publicly documented verification procedures enable opposing parties and courts to assess the trustworthiness of the digital identification process.

E.3 Federal Rules of Evidence 803(6): Business Records Exception to Hearsay

Rule: FRE 803(6) excludes from the hearsay rule a record of a regularly conducted activity if: (A) made at or near the time by someone with knowledge; (B) kept in the course of a regularly conducted business activity; (C) making the record was a regular practice; and (D) these conditions are shown by testimony of the custodian or another qualified witness, or by a certification under FRE 902(11), (12), or (13). The record may be excluded if "the source of information or the method or circumstances of preparation indicate a lack of trustworthiness."

OVERT Mapping:

Elements (A), (B), (C), and (D) map identically to the FRE 902(13) analysis above. The additional trustworthiness inquiry under FRE 803(6)(E) is addressed by OVERT's design properties:

Trustworthiness Factor	OVERT Feature
Source reliability	Attestation records are generated by cryptographically verified arbiters (binary identity derived by independent notaries, not self-reported) operating within verified network isolation (NETATT).
Preparation circumstances	Attestation is continuous, automatic, and not created in anticipation of litigation. The system operates identically regardless of whether litigation is pending.
Tamper evidence	Transparency log provides append-only storage with Merkle tree consistency proofs. Any modification is detectable through signed tree head comparison.
Independent verification	Any party can independently verify attestation integrity using publicly available verification procedures without operator cooperation.

Analysis: OVERT attestation artifacts are designed with properties relevant to the FRE 803(6) trustworthiness inquiry. Whether a specific court finds a particular deployment's attestation artifacts trustworthy under 803(6)(E) will depend on the deployment-specific facts, including system reliability, operational consistency, and the circumstances of record creation. This analysis identifies design alignment; it does not predict admissibility or trustworthiness determinations.

E.4 Federal Rules of Civil Procedure 37(e): Failure to Preserve ESI

Rule: FRCP 37(e) addresses the consequences of failing to preserve electronically stored information (ESI) that should have been preserved in anticipation or conduct of litigation. If ESI is lost because a party failed to take reasonable steps to preserve it, and it cannot be restored or replaced through additional discovery, the court may: (1) upon finding prejudice, order measures no greater than necessary to cure the prejudice; or (2) upon finding that the party acted with intent to deprive, presume the information was unfavorable, instruct the jury accordingly, or dismiss the action.

OVERT Mitigation:

OVERT Section 21 (Legal Preservation and Production) directly addresses FRCP 37(e) exposure:

FRCP 37(e) Element	OVERT Mitigation	Relevant Section
"reasonable steps to preserve"	Section 21.1 requires operators to define and publish retention policies meeting or exceeding the longer of regulatory requirements or applicable statutes of limitation. Section 21.2 requires legal hold procedures upon receipt of litigation hold notice or preservation demand.	Section 21.1, 21.2
"lost because a party failed"	The transparency log provides an independent, tamper-evident record of what attestation artifacts existed. Even if the operator's local copy is lost, the transparency log entries (hashes, inclusion proofs) remain, proving that the artifacts existed and establishing their content hashes.	ATT-4, ATT-4 (Section 8)
"cannot be restored or replaced"	Section 21.3 requires immutable export capabilities. The transparency log + notary signatures provide partial reconstruction capability even if local evidence is lost. Co-epoch binding and receipt hashes enable a court to determine the scope of loss.	Section 21.3, 21.4
"intent to deprive"	OVERT audit trails make intentional destruction detectable. If an operator deletes local evidence, the transparency log still contains the receipts — showing what was attested and when. The gap between transparency log entries and available local evidence is itself evidence of deletion.	ATT-4, ATT-4 (Section 8)

Analysis: OVERT does not eliminate FRCP 37(e) exposure — no technical system can prevent a party from destroying evidence if they are willing to accept the consequences. However, OVERT creates a structural environment where: (a) preservation obligations are documented in the operator's published retention policy; (b) legal hold procedures are defined and attestable; (c) the transparency log provides an independent record of what artifacts existed, making destruction detectable; and (d) the gap between what the log shows existed and what the operator can produce is itself a measurable, verifiable retention-integrity signal.

E.5 International Admissibility References

United Kingdom: Civil Evidence Act 1995

The Civil Evidence Act 1995 abolished the common law rule against hearsay in civil proceedings, making all relevant evidence admissible subject to weight. Section 9 provides that a document shown to form part of the records of a business may be received in evidence without further proof, with a certificate signed by an officer of the business sufficing to establish that the document forms part of those records; Section 8 separately permits a statement in a document to be proved by production of the document or an authenticated copy. OVERT attestation artifacts, accompanied by custodian certification (Section 21.3(e)), are designed to support business-records certification under Section 9 and proof by authenticated copy under Section 8. The weight given to such evidence remains at the court's discretion, informed by factors including the reliability of the computer system and the manner in which the data was processed.

The UK has not enacted standalone AI legislation as of March 2026, maintaining a "pro-innovation" regulatory approach with cross-sector principles applied through existing regulators. The Online Safety Act 2023, with a February 2026 amendment bringing standalone AI chatbots within scope, creates an expanding statutory backdrop. In the absence of AI-specific evidentiary rules, OVERT attestation artifacts would be assessed under general principles of electronic evidence admissibility.

European Union: eIDAS Regulation (Regulation 910/2014 and eIDAS 2.0)

The eIDAS Regulation provides a legal framework for electronic identification and trust services across EU member states. Under eIDAS:

- **Electronic signatures** (Article 25): An electronic signature shall not be denied legal effect solely on the grounds that it is in electronic form. Qualified electronic signatures have the equivalent legal effect of handwritten signatures.

- **Electronic seals** (Article 35): An electronic seal shall not be denied legal effect solely on the grounds that it is in electronic form. Qualified electronic seals enjoy a presumption of integrity of the data and correctness of the origin.
- **Electronic time stamps** (Article 41): An electronic time stamp shall not be denied legal effect solely on the grounds that it is in electronic form. Qualified electronic time stamps enjoy a presumption of accuracy.
- **Electronic documents** (Article 46): An electronic document shall not be denied legal effect solely on the grounds that it is in electronic form.

OVERT attestation artifacts — cryptographically signed, timestamped, and integrity-verified — are designed to satisfy the structural requirements for electronic evidence under eIDAS. Organizations operating under eIDAS may additionally seek qualified trust service provider (QTSP) status for their notary network operations, which would provide the legal presumptions associated with qualified electronic signatures, seals, and time stamps. The eIDAS 2.0 update extends the framework to include electronic ledgers, which may be relevant to OVERT transparency log operations.

For EU AI Act operationalization specifically, the Article 12 logging obligations are expected to be implemented through harmonized standards developed by CEN-CENELEC JTC 21; referencing OVERT within, or contributing it to, those deliverables is the highest-leverage path to regulatory recognition. Separately, W3C Verifiable Credentials is a natural encoding for both the conformance claim (Section 22.4) and the IDENT-1 delegation chain, and is worth tracking as agent-identity standardization (OAuth token exchange, GNAP, and emerging agent-identity efforts) matures. These are informative positioning notes, not conformance requirements.

The revised EU Product Liability Directive (Directive 2024/2853), with transposition deadline December 9, 2026, explicitly treats software and AI systems as products. Article 9 creates rebuttable presumptions of defectiveness where a defendant fails to comply with disclosure obligations. OVERT attestation artifacts are designed to support the operator's ability to respond to disclosure obligations with verifiable, tamper-evident records.

Annex F: Sample Citation Language (Informative)

This annex provides canonical citation forms for referencing OVERT conformance in legal, procurement, insurance, and regulatory contexts.

F.1 Canonical Conformance Citation Format

The standard citation form for an OVERT conformance claim follows the grammar defined in Section 22.4. All claims include a human-readable scope summary and exclusions summary. Level 3 and Level 4 claims additionally include coverage percentage, denominator class, scope hash, and exposure-window duration; Level 4 claims also carry the IAP topology, model-identity binding, and Measured Component Set fields (Section 22.4, Section 22.10).

Example (Level 3):

> EXAMPLE — *OVERT Level 3 Agentic — v1.1.0, Profile v1.0, Scope Summary: sys-agent-010 patient-facing agentic workflows (API gateway gw-prod-01, FHIR interface, voice endpoint), Exclusions: None (full coverage verified), Scope: 85% of inbound API traffic, Denominator: Independently-Attested, Scope Statement: sha256:[scope-hash], Exposure Window: 0h (0%), IAP Topology: Multi-IAP (2,3), 2026-06-15*

Example (Level 2):

> EXAMPLE — *OVERT Level 2 Core — v1.1.0, Profile v1.0, Scope Summary: sys-cda-001 clinical documentation API (FHIR R4 interface, HL7v2 ADT feed), Exclusions: Not assessed: batch-analytics-002 (scheduled for Q3 assessment), 2026-03-15*

F.2 Guidance for Referencing OVERT in External Documents

Organizations referencing OVERT conformance in procurement, insurance, regulatory, or legal contexts are advised to:

1. Use the canonical citation format from Section F.1, which includes scope summary, exclusions, denominator source, and exposure-window fields.
2. Not imply that OVERT conformance establishes legal compliance, regulatory approval, insurance coverage, or a judicially recognized standard of care.
3. Not imply that OVERT conformance covers systems, interfaces, or traffic classes outside the declared scope.
4. Consult qualified legal counsel when drafting contract, insurance, or regulatory language that references OVERT.
5. Clearly distinguish between independently verifiable signals and operator-dependent signals when making claims about evidence quality.
6. Use the Level 4 designation "Evidence-Grade" only as the name of the conformance level defined in Section 22.2, stated with its level number and declared scope — not as a free-standing descriptor of a product, system, or organization.

> NOTE — *Previous versions of this annex included sample legal, procurement, insurance, and regulatory paragraphs. Those samples were removed because copy-paste-ready advocacy language in a standard creates misrepresentation risk. Organizations should draft context-specific language with qualified counsel using the canonical citation format and the scope/exclusions/denominator disclosures required by Section 22.4.*

F.3 Disclaimer

> NOTE — *This annex is informative only and does not constitute legal advice. OVERT has not been judicially recognized as defining a standard of care. No insurer, regulator, or court has adopted OVERT as dispositive evidence. Citation forms should be adapted to the specific legal jurisdiction, regulatory framework, and contractual context in which they are used. Organizations should consult qualified legal counsel when referencing OVERT in any external document.*

Annex G: Supplementary Requirements (Normative — added in v1.1)

> STATUS. *This annex is* NORMATIVE. *It defines supplementary requirements added in version 1.1 that operationalize, and do not modify, the Part 1–22 normative core. Each subsection states the conformance level and scope at which its requirements apply. Where a subsection states that its requirements apply at AAL-4 for Level 4 Agentic-Extended claims, conformance to that subsection is required for the corresponding claim in addition to the conformance matrix of Section 22.5. Section G.4 is an informative reference that documents an artifact already mandated normatively by Section 10; it introduces no new obligation.*

G.1 Local CAS Evidence Retrieval and Retention Integrity

> STATUS. *This section defines normative requirements (SHALL/SHALL NOT/MAY per RFC 2119) added in version 1.1. It operationalizes, and does not replace, the content-verification provisions of Section 20.3(c) and Section 20.4, the immutable-export provisions of Section 21.3, and the retention-integrity Operational Signal defined in Annex D, Section D.2. All endpoint paths, wire encodings, and message schemas referenced below are specified in the registered Protocol Profile; this section specifies required behavior, not transport.*

G.1.1 Purpose and Scope

OVERT's non-egress architecture (Section 17) requires that only cryptographic commitments and profile-defined metadata cross the operator's trust boundary during routine attestation. Section 20.3(c) and Section 20.4 establish that content verification — auditor access to the protected payloads underlying attested interactions — is the exception, permitted only under legal authority or contractual agreement and accompanied by cryptographic proof that the accessed artifacts

are genuine and contemporaneous. This subsection defines the normative interface and integrity-assurance requirements by which a relying party exercises that exception against the operator's Local Content-Addressable Storage (CAS, Annex A.7), and by which an external risk bearer obtains continuous assurance that required evidence is being retained.

The requirements in this subsection are normative at AAL-4 for **all Level 4 conformance claims** — Core, Agentic, and Agentic-Extended: a deployment claiming the Evidence-Grade level SHALL be able to satisfy a lawful content-verification request through this interface (Section 20.3(c)). They are OPTIONAL for Level 1 through Level 3 claims; an operator at those levels that does not expose a content-verification interface SHALL declare the omission in the conformance statement Exclusions field with architectural justification (Section 22.4).

G.1.2 Evidence Retrieval Interface

An operator that exposes a content-verification interface SHALL provide a deterministic evidence-retrieval operation keyed on the `evidence_commitment` field of the Extended Envelope (Annex B.6). The operation accepts one or more `evidence_commitment` values identifying targeted receipts, a declared governance reason for the retrieval, and the verifiable identity of the requesting principal; it returns, for each located commitment, the original canonicalized payload (the prompt, response, and policy-evaluation scores as committed), together with an enumeration of commitments that could not be located.

The interface SHALL satisfy the following requirements:

(a) **Authorization.** The interface SHALL require out-of-band, mutually authenticated authorization. Retrieval requests SHALL NOT be served on the routine receipt-service egress path (Section 17.4); content verification is the exception path of Section 20.4, not part of routine attestation.

(b) **Governance-reason declaration.** Each request SHALL declare a governance reason drawn from a closed enumeration defined in the registered Protocol Profile (for example: incident response, regulatory audit, insurance claim, legal discovery). The declared reason SHALL be attested and SHALL be retained as part of the operator's audit record.

(c) **Verifiability of retrieved payloads.** The requesting party SHALL be able to independently recompute the `evidence_commitment` over each returned canonicalized payload, using the canonicalization method (Section 17.1) and the commitment construction specified in the registered Protocol Profile, and to confirm that the recomputed value equals the commitment recorded in the corresponding receipt. Because the `evidence_commitment` construction is keyed (Protocol Profile 1.0 derives it via HMAC under operator-held keys within the split-knowledge key hierarchy; see Annex A.28 and Annex B), the operator SHALL provide the requesting party a profile-defined **commitment-opening mechanism** sufficient to recompute and verify the commitment under the

request's authorization — for example, disclosure of a one-time, commitment-scoped opening key; a key-management-service (KMS) verification oracle; or verification within an attested verifier enclave — without disclosing long-term content-binding keys. A payload whose recomputed commitment does not match its recorded `evidence_commitment` SHALL be treated as a content-integrity failure and reported as such.

(d) **Non-egress preservation.** Retrieval under this subsection occurs within the operator's environment under the requesting party's lawful authority or contractual agreement (Section 20.4); it does not relax the non-egress property of Section 17. Protected content disclosed under this interface is disclosed to an authorized verifier under Section 20.4, not egressed to the attestation layer.

(e) **Closed schema.** The request and response message schemas SHALL be closed (unknown fields rejected; `additionalProperties: false`), consistent with the closed-schema requirement of Section 17.4. The concrete field set, encodings, the endpoint path, and the commitment construction are specified in the registered Protocol Profile.

> NOTE — RELATIONSHIP TO IMMUTABLE EXPORT. *The evidence-retrieval interface is a targeted, commitment-keyed lookup. It does not replace the immutable export package of Section 21.3, which remains the mechanism for producing a verifiable, signed, transparency-log-consistent corpus for litigation or regulatory examination. Production of operator-local artifacts under either mechanism may require operator cooperation or lawful process (Section 21.3).*

G.1.3 Proof of Possession

An external risk bearer or auditor may require continuous assurance that the operator is in fact retaining the evidence payloads required for future investigation, subrogation, or audit, without compelling the operator to egress those payloads. An operator that makes such an assurance SHALL implement a Proof of Possession (PoP) challenge–response operation with the following properties:

(a) The requesting party generates a fresh challenge value of at least 256 bits from a cryptographically secure random source and submits it together with a subset of `evidence_commitment` values to be challenged.

(b) For each challenged commitment, the operator's CAS retrieves the corresponding canonicalized payload and computes a possession value that cryptographically binds the challenge value to the canonical payload, using the construction specified in the registered Protocol Profile (a keyed or domain-separated hash over the challenge value and the canonical payload). The operator returns the possession values and a count of challenged commitments whose payloads could not be retrieved.

(c) The requesting party SHALL retain the challenge value. If the audit is later escalated under Section 20.4, the requesting party may obtain the full payload under that authority, recompute the possession value with the retained challenge value, and thereby verify that the operator held the unmodified payload at the time of the original challenge.

(d) The PoP operation SHALL be subject to the same out-of-band, mutually authenticated authorization required by Section G.1.2(a). The challenge construction, possession construction, encodings, and endpoint path are specified in the registered Protocol Profile.

> NOTE. *Proof of Possession provides retention assurance, not contemporaneity. A successful PoP demonstrates that the operator can produce a payload matching the recorded* `evidence_commitment` *at the time of challenge; contemporaneity of the original attestation continues to be established by the receipt's co-epoch binding (ATT-2, Section 8) and transparency-log inclusion (Section 20.1), not by PoP.*

> NOTE — OPERATOR PROTECTION. *A Proof of Possession challenge requires the operator to read and hash full payloads; a bulk challenge over very large or numerous payloads (for example, multimodal contexts spanning millions of tokens or large media objects) can impose significant disk-I/O and CPU load amounting to an authorized denial of service. PoP endpoints SHOULD support asynchronous processing, rate-limiting, batch-size limits, and/or a profile-defined sampling discipline (for example, Merkle-tree-based spot-checks over a payload's chunks) to bound operator cost while preserving the integrity guarantee.*

G.1.4 Retention Integrity Signal

The count of challenged commitments whose payloads could not be retrieved during a Proof of Possession operation (Section G.1.3(b)), together with the count of commitments reported as not located during evidence retrieval (Section G.1.2), SHALL feed the **retention integrity** Operational Signal enumerated in Annex D, Section D.2. A non-zero retention-failure count indicates that evidence required to be retained under the operator's retention schedule (Section 21.1) is not retrievable, which constitutes a gap in attestation continuity.

The retention integrity signal SHALL satisfy the risk-signal properties of Section 4.6 and the derivation requirements of Annex D, Section D.3; its identifier, formula (numerator, denominator, unit), source artifacts, aggregation window, missing-data handling, minimum credibility threshold, and severity classification are specified in the registered Protocol Profile or companion signal specifica-

tion. Retention-integrity failures SHALL be reportable to relying parties consistent with the operator's anomaly-triage obligations (Section 4.7) and SHALL NOT be silently suppressed.

> INFORMATIVE. *Downstream relying parties — including parametric risk bearers — may treat a non-zero retention-integrity signal as a governance condition with contractual consequences. Any such consequence is a matter of the relevant contract or policy and is outside the scope of this standard; this standard defines only the signal and its verifiable derivation.*

G.2 HTTP Transport Binding for Cross-Boundary Attestation

> STATUS. *This section is normative. It specifies the HTTP wire encoding for the cross-boundary attestation protocol defined normatively in Section 4.8, for HTTP/1.1 and HTTP/2 transports. The* `OVERT-Parent-Attestation-Id` *header is the wire encoding of the existing Section 4.8.2* `parent_attestation_id` *reference and adds no obligation beyond Section 4.8. The* `OVERT-Trace-Id` *correlation header (Section G.2.2) is a new requirement of this binding. Transports other than HTTP are bound by their own Protocol Profile specifications.*

G.2.1 Scope

This binding applies when a conformant arbiter (Section 3.2) participates in a cross-boundary workflow (Section 4.8) and the boundary is crossed over an HTTP/1.1 or HTTP/2 transport. It defines canonical request headers that carry the Section 4.8.2 `parent_attestation_id` reference and a correlation identifier across heterogeneous runtimes, gateways, and service meshes without modification of application code. Transports other than HTTP are specified by their own Protocol Profile bindings. This binding does not alter the Section 4.8.8 conformance condition: cross-boundary controls are normative at AAL-4 for Level 3 and Level 4 claims involving cross-boundary workflows, and are not required for workflows that do not cross trust boundaries.

G.2.2 OVERT Context Headers

An arbiter operating on the upstream (egress) path of a cross-boundary HTTP request SHALL inject the following headers. An arbiter operating on the downstream (ingress) path SHALL extract and validate them per Section G.2.4.

Header	Value	Definition
`OVERT-Trace-Id`	32- or 64-character lowercase hexadecimal string (a 128-bit W3C Trace Context identifier, or a 256-bit identifier)	Correlation identifier for the multi-boundary execution graph. It is a transport-layer correlation aid only; it is NOT an attestation artifact and SHALL NOT be relied upon as evidence of enforcement.
`OVERT-Parent-Attestation-Id`	64-character lowercase hexadecimal string	The SHA-256 hash of the upstream receipt's `attestation_id`, computed exactly as defined in Section 4.8.2 and derived from the Phase 2 provisional receipt per Section G.2.3(b) — the upstream log entry may not yet exist at injection time (ATT-3.3). This is the wire encoding of the `parent_attestation_id` field.

`OVERT-Parent-Attestation-Id` SHALL match `^[a-f0-9]{64}$` (a SHA-256 digest). `OVERT-Trace-Id` SHALL match `^[a-f0-9]{32}([a-f0-9]{32})?$`, accepting either a 32-character (128-bit) W3C Trace Context `trace-id` for interoperability with OpenTelemetry, or a 64-character (256-bit) identifier. Header names are case-insensitive per the HTTP specification; the forms above are the canonical spellings and follow RFC 6648 (no `X-` prefix for newly defined fields).

> NOTE. *Earlier drafts of this binding proposed a third header declaring an "Identity Assurance Level (IAL)". OVERT does not define IAL; the standard's assurance taxonomy is the Attestation Assurance Level (AAL-1 through AAL-4, Section 4.1), which classifies attestation artifacts, not transport-asserted identity. No identity-assurance header is defined by this binding. Where an arbiter needs to convey the assurance level of an upstream receipt, that level is established by verifying the referenced receipt itself (Section 4.8.6), not by trusting a transport header.*

G.2.3 Egress Injection (Upstream Path)

For each outbound tool invocation, model API call, or RPC that crosses a trust boundary through a conformant arbiter:

(a) **Trace correlation.** If the incoming request already carries an `OVERT-Trace-Id`, the arbiter SHALL preserve and propagate that value unchanged. If no `OVERT-Trace-Id` is present, the arbiter SHALL generate a new identifier using a cryptographically secure random source.

(b) **Parent binding.** The arbiter SHALL set `OVERT-Parent-Attestation-Id` to the SHA-256 hash of the `attestation_id` of its own provisional receipt for the originating action, computed as specified in Section 4.8.2. The provisional receipt is the Phase 2 artifact defined in Section 8 (control ATT-3.2). Because Phase 3 notary co-signature is asynchronous (Section 8), the injected reference SHALL be derived from the Phase 2 provisional `attestation_id`; if and when the upstream receipt's transparency-log status changes, the downstream `parent_reference_status` is resolved per Section 4.8.6 and Section 4.8.7, not by re-emitting the header.

G.2.4 Ingress Extraction and Validation (Downstream Path)

When a conformant arbiter intercepts an incoming HTTP request at a trust boundary:

(a) **Extraction.** The arbiter SHALL extract `OVERT-Trace-Id` and `OVERT-Parent-Attestation-Id` where present.

(b) **Validation and mapping.** The arbiter SHALL validate each present value against its pattern (`OVERT-Trace-Id` against `^[a-f0-9]{32}([a-f0-9]{32})?$`; `OVERT-Parent-Attestation-Id` against `^[a-f0-9]{64}$`). If `OVERT-Parent-Attestation-Id` is present and structurally valid, the arbiter SHALL record it as the `parent_attestation_id` of the receipt it generates for the downstream action, per Section 4.8.2.

(c) **Status assignment.** The arbiter SHALL set the downstream receipt's `parent_reference_status` (Section 4.8.7) according to the outcome of header extraction and the **generation-time** checks of Section 4.8.7. Transparency-log inclusion is not checked at intercept — an upstream Phase 2 provisional receipt's log entry may legitimately not yet exist (ATT-3.3); relying parties verify inclusion during DAG validation (Section 4.8.6):

- If `OVERT-Parent-Attestation-Id` is absent, `parent_reference_status` SHALL be `UNAVAILABLE`.
- If the header is present but fails structural validation, or a presented upstream receipt fails a generation-time check (signature or co-epoch binding), `parent_reference_status` SHALL be `INVALID`.
- If an upstream endpoint queried for the generation-time checks does not respond within the profile-defined timeout, `parent_reference_status` SHALL be `TIMEOUT`.
- Otherwise `parent_reference_status` SHALL be `VALID`.

(d) **Continuation.** When `parent_reference_status` is any value other than `VALID`, the downstream receipt SHALL still be generated; attestation of the downstream boundary's own controls

proceeds regardless of upstream availability (Section 4.8.7). The transaction MAY proceed where permitted by local policy. Relying parties SHALL treat a non-`VALID` link as a gap in cross-boundary verification, not as a failure of the downstream boundary's own attestation (Section 4.8.7).

G.2.5 Header Survival and Reporting

A gateway, sidecar, or proxy claiming conformance to this binding SHALL propagate the OVERT context headers without alteration to downstream hops it forwards. Where a downstream service or intermediary strips or fails to propagate the headers, the resulting incomplete link SHALL be recorded with `parent_reference_status = UNAVAILABLE` at the next conformant ingress point (Section G.2.4(c)) and SHALL be reported through the operator's gap-accounting reporting (ATT-3.4) and as an exposure window in risk-signal reporting where the missing link corresponds to an unattested interval (Annex D; signal definitions in the registered Protocol Profile). This binding adds no new conformance level and no new control identifier; conformance with it is conformance with Section 4.8 over an HTTP transport.

G.3 Automated Auditor Discovery and Well-Known Endpoint Protocol

> STATUS. *This section is normative. The endpoint paths, JSON wire schemas, and conformance test vectors referenced below are specified in the registered Protocol Profile; this annex defines only the discovery obligation and its binding to existing auditability requirements (Section 20) and measurement requirements (Section 9). The requirements of this section apply at Level 4.*

G.3.1 Purpose and Scope

Section 20 (Third-Party Auditability) requires that the attestation system enable third-party verification of governance claims without requiring trust in the operator. Section 20.5 delegates the auditor verification procedures to the registered Protocol Profile. This annex specifies a discovery mechanism that allows a relying party to locate those procedures and the associated verification artifacts automatically, without bespoke per-operator integration.

For Level 4 conformance claims, the discovery publisher — the qualified IAP (Section 22.7), or the operator self-hosting the discovery and artifact-retrieval endpoints for an IAP-backed deployment — SHALL publish a machine-readable discovery document under the `/.well-known/` URI namespace

defined by RFC 8615. (Operator-controlled notary infrastructure itself satisfies at most AAL-3 — Section 4.1.1, ATT-5.1(c); self-hosting in this annex refers to the discovery and retrieval surface, never the notary trust anchor.) The discovery document SHALL enumerate the endpoints from which an auditor can retrieve the cryptographic artifacts required by Sections 20.1, 20.3(a), and 20.3(b).

G.3.2 The OVERT Discovery Document

A conformant publisher SHALL expose an OVERT discovery document at the well-known location specified in the registered Protocol Profile (Protocol Profile 1.0 registers the path `/.well-known/overt-configuration`). The document SHALL be retrievable by an unauthenticated HTTP GET and SHALL be served over TLS.

The discovery document SHALL be a closed-schema JSON object. The Protocol Profile specifies the complete field-by-field schema, types, and constraints; at minimum it SHALL include:

- the authoritative issuer URI of the IAP or operator;
- the set of registered OVERT Protocol Profile identifiers the publisher supports;
- the base URI of the RFC 6962 transparency log required by Section 20.1;
- the URI template for per-epoch artifact retrieval (see Section G.3.3);
- the URI of the key set containing the notary network's public verification keys.

The discovery document MAY additionally reference a content-retrieval endpoint for verifiable records held in the operator's local storage (Section 20.3(c) content verification). Because content verification is the exception and operates only under legal authority or contractual agreement (Section 20.4), any such endpoint SHALL require out-of-band, mutually authenticated authorization and SHALL NOT be exposed for unauthenticated routine access. For Level 4 claims, the content-verification **capability** itself is mandatory (Annex G.1.1); what remains optional is its unauthenticated discovery publication — a Level 4 operator MAY omit the endpoint from the public discovery document and convey it out-of-band to authorized verifiers under Section 20.4.

> NOTE. *The optional content-retrieval endpoint referenced above corresponds to the Local CAS evidence-retrieval interface of Section G.1. Where a Level 1–3 operator does not expose that interface, the corresponding discovery field is omitted or marked reserved, with the exclusion declared per G.1.1.*

G.3.3 Per-Epoch Artifact Retrieval

Section 9 requires post-epoch publication of the Digest Publication Ledger (DPL) (MEA-1.4) and the S3P epoch nonce (MEA-2.5) so that auditors can reconstruct all sampling decisions for a closed

epoch. The per-epoch retrieval endpoint advertised in the discovery document SHALL return, for any closed epoch, the artifacts required for that reconstruction.

The per-epoch response SHALL be a closed-schema JSON object specified in the registered Protocol Profile. For each epoch it SHALL convey:

- the epoch identifier;
- the epoch status, drawn from a closed enumeration that distinguishes active epochs (for which the nonce has not yet been revealed) from closed epochs;
- the S3P epoch nonce, which SHALL be present for a closed epoch and SHALL be omitted while the epoch is active (consistent with the withhold-during-epoch requirement of MEA-2.1);
- the Digest Publication Ledger for the epoch, including its Merkle root and the complete, deterministically ordered set of request commitments processed during the epoch (per MEA-1.4);
- the notary signature over the epoch identifier, status, revealed nonce, and DPL root.

For an active (not yet closed) epoch, the publisher SHALL withhold the epoch nonce and the DPL contents that would permit premature reconstruction, consistent with MEA-2.1 and MEA-2.5. Reveal of these artifacts before epoch close is a conformance failure.

G.3.4 Auditor Verification Flow

The discovery document enables, but does not replace, the auditor verification procedures specified in Section 20 and the registered Protocol Profile. A relying party performing independent verification proceeds as follows:

1. Retrieve the OVERT discovery document from the publisher's declared attestation domain.
2. Retrieve the notary network's public verification keys from the key-set URI advertised in the discovery document.
3. Read receipts from the transparency log (Section 20.1) to obtain the epoch identifiers in scope.
4. Retrieve the revealed nonces and DPLs for the closed epochs from the per-epoch retrieval endpoint (Section G.3.3).
5. Recompute the sampling tags and S3P sampling boundaries using the construction specified in the registered Protocol Profile (MEA-1.2, MEA-2.2) and independently verify the coverage ratio (Section 4.6, item 1) and the statistical safety signals (Section 4.6, item 2; MEA-2.4).

Where a publisher advertises endpoints but a relying party finds a closed epoch for which the required artifacts (nonce, DPL, notary signature) are unavailable or fail verification, the relying party SHALL treat the affected epoch as an attestation gap event for the purposes of gap accounting (Section 4.6, item 3; ATT-3.4).

G.3.5 Conformance

A Level 4 publisher (the IAP, or an operator self-hosting the discovery and retrieval endpoints for an IAP-backed deployment) SHALL expose the discovery document and the per-epoch retrieval endpoint as defined in this annex and the registered Protocol Profile. The discovery document SHALL accurately reflect the publisher's live endpoints; a discovery document that advertises endpoints that do not serve the required artifacts is non-conformant. Discovery is a Level 4 obligation; Levels 1 through 3 MAY publish a discovery document but are not required to.

G.4 ControlAction Reference Schema (Informative)

> STATUS: INFORMATIVE. *This section documents the wire-level structure of the* `ControlAction` *artifact and its validation procedure. It introduces no new conformance obligations. The normative requirement to emit, gate, and bound* `ControlAction` *artifacts is established by Section 10, controls RES-1.2, RES-1.3, and RES-1.4; the cryptographic primitives are those already registered in Annex B.1 (Ed25519 for controller signatures, per RFC 8032). The authoritative, field-level specification is the Protocol Profile document itself.*

`ControlAction` is the attestation artifact emitted by the bounded control loop when verified violation metrics exceed a policy threshold (RES-1.2). Its glossary definition is A.11. The arbiter evaluates every `ControlAction` through the five cryptographic gates of RES-1.3 before applying the requested parameters, and enforces the parameter bounds of RES-1.4 (both obligations are normative in Section 10); this section does not restate those obligations, it only specifies the structure over which they operate. In this structure the requested governance response — the "action type" and "scope" of glossary A.11 — is expressed as the delta between `params_before` and `params_after`, and the temporal binding (the A.11 "timestamp" and co-epoch binding) is carried by `epoch` and the co-epoch receipt referenced by `proof_ref`.

G.4.1 Closed Schema

Protocol Profile 1.0 represents the `ControlAction` artifact as a closed-schema structure. No additional fields are permitted. The parameter set (`sampling_prob`, `queue_max`, `rate_limit`) is the set enumerated in RES-1.2; profiles defining additional bounded parameters extend the parameter objects and the corresponding RES-1.4 bounds in the registered Protocol Profile, not in the core standard.

```
{
  "type": "object",
  "properties": {
    "epoch": {
      "type": "integer",
      "description": "Epoch identifier in which the action is applied. Used by the
Gate 2 (epoch currency) check of RES-1.3."
    },
    "binary_hash": {
      "type": "string",
      "pattern": "^[a-f0-9]{64,128}$",
      "description": "Digest of the controller binary issuing the action (SHA-256 by
default; a larger digest, e.g. SHA-384/512, where the registered Protocol Profile
specifies one)."
    },
    "params_before": {
      "type": "object",
      "properties": {
        "sampling_prob": { "type": "string", "description": "Decimal string or scaled
integer per Annex B.3; IEEE-754 float not permitted in attested structures." },
        "queue_max": { "type": "integer" },
        "rate_limit": { "type": "integer" }
      },
      "description": "System parameters in effect prior to the action.",
      "additionalProperties": false
    },
    "params_after": {
      "type": "object",
      "properties": {
        "sampling_prob": { "type": "string", "description": "Decimal string or scaled
integer per Annex B.3; IEEE-754 float not permitted in attested structures." },
        "queue_max": { "type": "integer" },
        "rate_limit": { "type": "integer" }
      },
      "description": "Requested new parameters. Subject to the RES-1.4 bounds
check.",
      "additionalProperties": false
    },
    "proof_ref": {
      "type": "string",
      "pattern": "^[a-f0-9]{64,128}$",
      "description": "Digest of the epoch metrics bundle (RES-1.1) that triggered the
action (SHA-256 by default; a larger digest where the registered Protocol Profile
specifies one). Resolved by the Gate 4 co-epoch receipt check."
    },
    "signature": {
      "type": "string",
      "description": "Ed25519 signature (Annex B.1, RFC 8032) by the authorized
controller key over the canonical encoding of epoch, binary_hash, params_before,
params_after, and proof_ref."
    }
  },
  "required": ["epoch", "binary_hash", "params_before", "params_after", "proof_ref",
```

```
"signature"],
  "additionalProperties": false
}
```

Numeric and digest fields follow the registered Protocol Profile. Per the deterministic-encoding constraints of Annex B.3, IEEE-754 floats are not used in attested structures; `sampling_prob` is therefore represented as a scaled integer or decimal string, as reflected in the schema above. Digest fields (`binary_hash` , `proof_ref`) are SHA-256 (64 hexadecimal characters) by default; a profile specifying a larger digest (e.g., SHA-384/512 for alignment with a higher-security signature suite) widens them within the `^[a-f0-9]{64,128}$` constraint. The signature scope and canonical byte layout are specified in the Protocol Profile.

G.4.2 Five-Gate Validation (Reference)

The five gates below restate, for reference, the validation sequence already required by RES-1.3 (gates 1–5), whose gate 3 enforces the parameter bounds defined in RES-1.4. They are reproduced here only to map each gate onto the schema fields above; the normative obligation is in Section 10. If any gate fails, the arbiter rejects the action and retains `params_before` .

Gate	Check	RES-1 basis	Field(s)
1	Verify `signature` against the authorized controller public key.	RES-1.3(1)	`signature` , `binary_hash`
2	`epoch` matches the current active epoch; stale actions are rejected.	RES-1.3(2)	`epoch`
3	All `params_after` values fall within the statically defined bounds of RES-1.4 (e.g., `p_min ⩽ sampling_prob ⩽ p_max`), independent of signature validity.	RES-1.3(3), RES-1.4	`params_after`
4	A valid co-epoch receipt exists for the metrics bundle named by `proof_ref` .	RES-1.3(4)	`proof_ref`
5	A valid co-epoch Network Attestation (NETATT, A.19) exists for `epoch` .	RES-1.3(5)	`epoch`

The arbiter is the enforcement component that performs gates 1–5 (Section 3 / A.2).

G.4.3 Test Vectors

Section 22.6.2 requires that a registered Protocol Profile include published test vectors for every cryptographic operation. Protocol Profile 1.0 provides reference `ControlAction` instances together

with their canonical encodings and Ed25519 signatures, and a worked five-gate evaluation (one passing and one per-gate failing case), in the Protocol Profile document (cf. Annex B.12).