OVERT

OBSERVABLE VERIFICATION EVIDENCE FOR RUNTIME TRUST

The Case for Verifiable Evidence

Purpose and scope · normative references · terms · architecture overview

DATE	June 2026
PUBLISHED BY	GLACIS Technologies, Inc.
REPRODUCES	Part 1: Foundations (Sections 1–4)
COMPLETE EDITION	overt.is
CONTACT	overt-review@glacis.io

OFFPRINT NOTICE

This fascicle reproduces Part 1: Foundations (Sections 1–4) of OVERT Version 1.1 without modification. Section numbering follows the Complete Edition, which is the sole authoritative text for conformance purposes. Conformance claims cite OVERT 1.1, never an individual fascicle. The Complete Edition and all fascicles are published at overt.is.

Contents of this Volume

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, implementers SHOULD NOT represent OVERT conformance as covering training, data lifecycle, or platform infrastructure security. Conformance statements that could be misread as covering these surfaces are non-conformant with the spirit of this standard.

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 October 1, 2025 (180 days after issuance). 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; OVERT conformance does not determine compliance with any framework crosswalked 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. Only cryptographic commitments cross trust boundaries.
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; only cryptographic commitments cross trust boundaries.

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 violation rate for [policy] did not exceed [bound]% during [epoch]," derived from cryptographically verifiable random sampling.

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. 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 registered profile per deployment.

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 involves 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

Deployment Architecture	Maximum AAL	Rationale
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

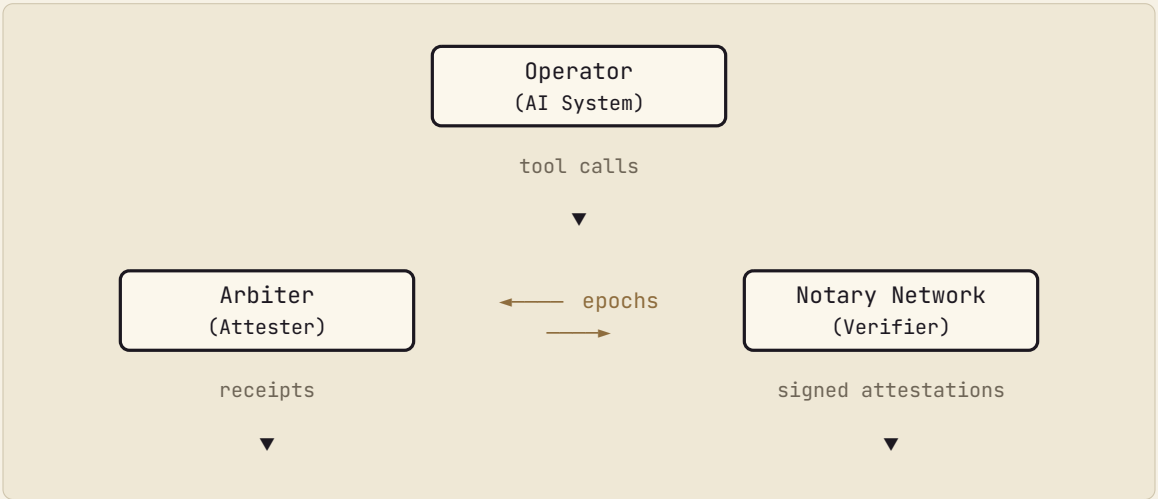
AAL-4 (cryptographic attestation with independent trust root) SHALL require that the notary service be operated by an entity structurally independent of the AI system operator — an Independent Attestation Provider (IAP) per Section 3.14. A single independent notary satisfies AAL-4. Multi-entity notary 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.

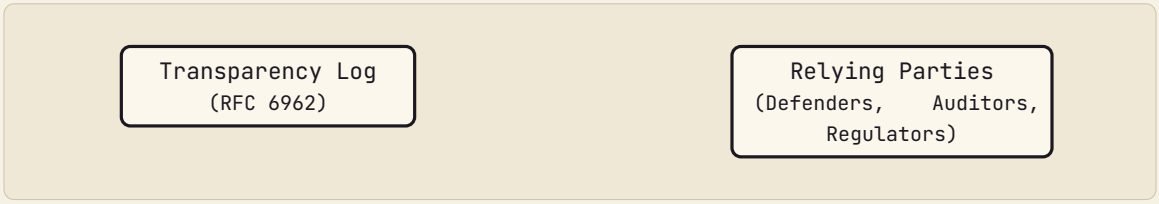
Deployments MAY graduate from AAL-3 to AAL-4 by engaging an independent notary service as specified in ATT-5. The transition SHALL be attested in the transparency log with notary set attestations from both the pre-transition and post-transition configurations.

> ***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 t-of-n notary verification 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 a registered OVERT Protocol 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; no single entity controls t nodes
Transparency log manipulation	Log operator tampers with append-only log	Split-view detection via published Signed Tree Heads; independent monitors
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	t-of-n requirement prevents single-notary censorship; uptime metrics
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

Threat Vector	Description	Required Mitigation
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 policy intent — distinct from prompt injection of the agent	Output containment at the boundary (ATT-3.5(a)); classifier version 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; POST_HOC receipts excluded from contemporaneous coverage (RES-5.2)
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)

Threat Vector	Description	Required Mitigation
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 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: signals whose denominators are independently verifiable (e.g., derived from independent ingress metrics or notary-observed counts) are classified as **independently verifiable signals**; signals whose denominators are operator-declared only 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 source. A signal whose denominator is operator-declared only SHALL NOT be presented as independently verifiable in conformance documentation or public claims.
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 source and its verifiability classification (independently verifiable or operator-dependent).

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 enforcement ratio** — the percentage of in-scope actions processed under optimistic enforcement (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), 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, the denominator source used for coverage and measurement claims, and whether that denominator source is independently verifiable or operator-declared only. Where independently verifiable ingress metrics are available (e.g., load balancer request counts, API gateway telemetry), the coverage ratio SHALL reference those metrics. Where independent ingress metrics are not available, the attestation SHALL disclose this limitation. Level 4 claims SHALL use independently verifiable ingress metrics or a registered-Protocol-Profile equivalent denominator source; absent such evidence, the implementation SHALL NOT claim Level 4 conformance for that scope.

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. 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 validates upstream receipt signatures, co-epoch bindings, and transparency log inclusion proofs before linking, 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.

4.8.7 FAILURE HANDLING

If an upstream receipt is unavailable or invalid at the time the downstream receipt is generated, the downstream receipt SHALL include a `parent_reference_status` field with one of the following values:

Status	Description
VALID	Upstream receipt was available, its signature and co-epoch binding were verified, and the `parent_attestation_id` was successfully computed and included

Status	Description
`UNAVAILABLE`	Upstream receipt was not available within the profile-defined timeout; `parent_attestation_id` could not be populated
`INVALID`	Upstream receipt was available but failed signature verification, co-epoch binding verification, or transparency log inclusion verification
`TIMEOUT`	Upstream transparency log did not respond within the profile-defined timeout for inclusion proof verification

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.

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