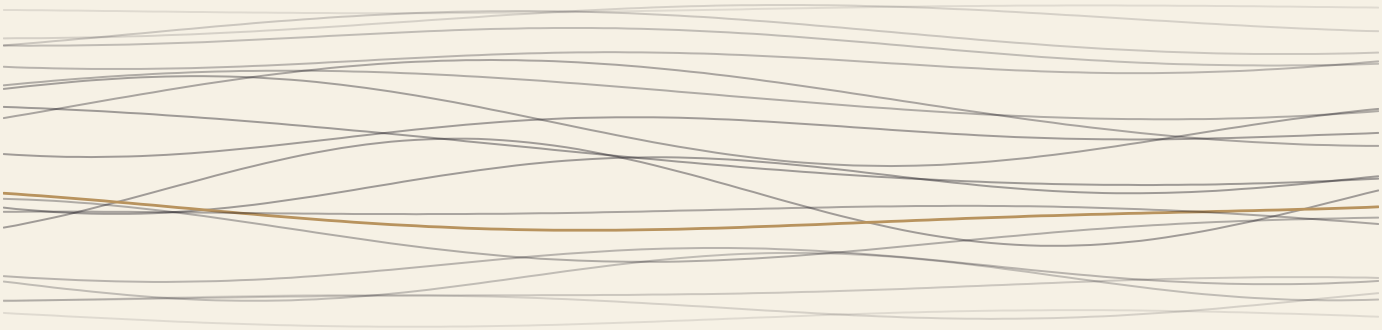
OVERT

OBSERVABLE VERIFICATION EVIDENCE FOR RUNTIME TRUST



How the Proof Is Made

Non-egress architecture · temporal binding · statistical measurement · auditability · legal preservation

DATE	June 2026
PUBLISHED BY	GLACIS Technologies, Inc.
REPRODUCES	Part 4: Attestation Architecture Requirements (Sections 17–21)
COMPLETE EDITION	overt.is
CONTACT	overt-review@glacis.io

OFFPRINT NOTICE

This fascicle reproduces Part 4: Attestation Architecture Requirements (Sections 17–21) 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 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.

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 registered Protocol Profile, including envelope schemas, commitment derivations, and receipt service constraints. The requirement to comply with a registered 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.

> ***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 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.

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 requiring no distributional assumptions (e.g., exact binomial intervals). 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, confidence interval bounds, 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 §19.7's upper-bound reporting:

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.*

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 source and SHALL reference independently verifiable ingress metrics where available (e.g., load balancer request counts, API gateway telemetry), rather than relying solely on DPL completeness metrics measured inside the declared mediation scope. Where independent ingress metrics are not available, the attestation SHALL disclose this limitation and SHALL mark the denominator as operator-declared. Level 4 claims SHALL use independently verifiable ingress metrics or a registered-Protocol-Profile equivalent denominator source.

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).

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 Retroactive Receipt Classification

The receipt `flags` field SHALL be a fixed-width unsigned integer. Bit 0 (least significant) indicates attestation temporality:

- **0** = contemporaneous: the attestation was generated and attested within the same epoch as the governed event.
- **1** = POST_HOC: the attestation was generated during the governed event but attested in a subsequent epoch (for example, due to network unavailability during a fail-open period per RES-5.2).

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

POST_HOC receipts are reconstruction artifacts and SHALL NOT be counted as contemporaneous attestation coverage for conformance, risk-signal reporting (Annex D), or litigation reporting pur-

poses. POST_HOC receipts SHALL be distinguishable from contemporaneous receipts in all export packages, signal computations, and audit reports.

> ***Note — Scope of POST_HOC classification:*** *POST_HOC 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 POST_HOC classification. 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.]