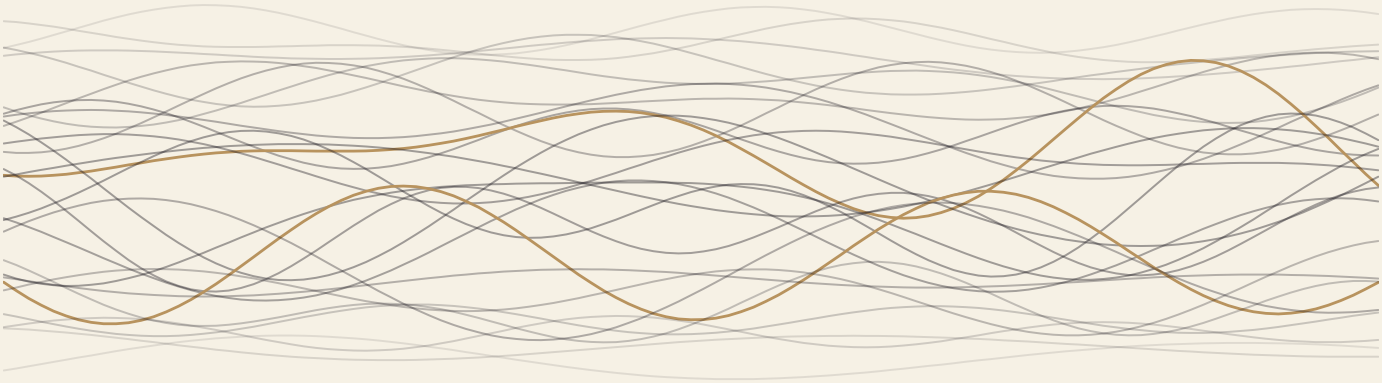
OVERT

OBSERVABLE VERIFICATION EVIDENCE FOR RUNTIME TRUST



Trust When Agents Act

Tool-call governance · multi-agent controls · capability-based access · disclosure · human-in-the-loop · drift

DATE	June 2026
PUBLISHED BY	GLACIS Technologies, Inc.
REPRODUCES	Part 3: Agentic AI Controls (Sections 11–16)
COMPLETE EDITION	overt.is
CONTACT	overt-review@glacis.io

OFFPRINT NOTICE

This fascicle reproduces Part 3: Agentic AI Controls (Sections 11–16) 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 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 verifia-*

> *bility 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 proves that context existed and was destroyed; 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.