CONFORMANCE STATEMENT

AARM Conformance Statement

Publisher. Etch (etch.systems)

Statement version. 1.4

Statement date. 2026-08-06

> **Specification reference.** Herman Errico. (2026). Autonomous Action Runtime Management (AARM). Cloud Security Alliance, February 2026. arXiv:2602.09433. Available at https://aarm.dev/spec. Licensed CC BY 4.0.

1. Scope

Etch is a signed audit primitive for autonomous AI agent decisions. Every event on an Etch chain carries a SHA 256 row hash chained to the previous row. Every closed epoch carries a hybrid signature envelope combining Ed25519 (FIPS 186-5) and SLH-DSA-SHA2-128f (FIPS 205), externally anchored into Sigstore Rekor and Bitcoin OpenTimestamps.

This statement addresses correspondence between Etch and the AARM specification (Herman Errico, Cloud Security Alliance, February 2026). Etch is deliberately not an enforcement product and does not sit in the execution path of an autonomous agent. The correspondence therefore covers only the AARM properties that Etch is designed to satisfy, which are tamper evident action receipts, cryptographic identity binding, and durable audit posture across agent sessions.

Properties in the AARM specification that concern runtime authorization decisions (ALLOW, DENY, MODIFY, DEFER, and the fifth decision class) are outside the scope of this statement. Etch integrates with vendors that occupy the enforcement path (see the ingest adapter list in the Etch documentation) and signs the record of what those vendors decided. The signing is what Etch is claiming conformance on.

2. Conformance posture

Etch claims conformance to the AARM specification's tamper evidence and identity binding posture. This is a posture level claim, not a per requirement claim under the AARM Conformance Testing Protocol. A formal per requirement result requires the AARM working group to evaluate an evidence package covering R1 through R6 for Core and R1 through R9 for Extended. That evaluation is a scheduled follow up. This statement will be revised when the working group returns findings.

The purpose of publishing before formal certification is operational. Compliance officers, procurement teams, and auditors already reviewing AI agent security stacks need to know which vendors align with the emerging AARM spec and on which properties. Publishing the correspondence map now, updated on the same schedule as Etch's own feature releases, gives those reviewers a working reference. Publishing the AARM spec itself is permitted under CC BY 4.0. Publishing a conformance claim is permitted because the AARM working group does not gate self declaration.

3. Correspondence to AARM properties

The table below maps AARM specification properties to the specific Etch endpoints or product properties that address them. Every claim is defensible from public materials as of the statement date above. An operational or specification change that would affect any row triggers a revision of this statement.

AARM property	Etch claim	Evidence path
Tamper evident action receipt Every autonomous action produces a receipt that cannot be modified after the fact without detection.	**Conditional conformance**	Every event on the Etch chain carries a SHA 256 row hash chained to the previous row. Every 1024 rows the epoch closes and the Merkle root is signed with a hybrid Ed25519 and SLH-DSA-SHA2-128f envelope. Silent modification is detectable by any auditor running etch-chain-verify against the storage root. This conformance is conditional. The chain and the external anchor catch receipts that were altered or deleted after being written (chain break at the next row) and receipts that were fabricated (invalid entry hash). The chain cannot detect actions that were never routed through the recording surface, because nothing is missing to observe. Ensuring that every autonomous action reaches the recording surface is a property of the credential boundary the operator deploys, which AARM property 8 (Least privilege posture) addresses. If credentials are scoped narrowly enough that the agent cannot act outside the recording surface, the 'every action' claim holds. If credentials are broad, the receipt layer alone cannot make that claim. Etch's pass on this property is therefore conditional on the deploying organization enforcing property 8 within its deployment.

AARM property	Etch claim	Evidence path
Cryptographic identity binding Every action receipt is bound to the identity that produced the action.	**Full conformance**	POST /v1/etch-chain/authority-receipt records a per event bounded authority claim. The claim is signed with Ed25519 by the acting party and verified server side against a per project pubkey registry before append. Enterprise deployments can bind the signing key to an HSM or TEE via POST /v1/etch-chain/hsm-attestation which accepts tpm2, yubikey, aws-nitro, gcp-shielded, azure-attestation, sev-snp, and intel-tdx attestation blobs.
External anchor for third party verification Signed records reach an independent witness log so third parties can verify inclusion without trusting the operator.	**Full conformance**	Every closed epoch is anchored to Sigstore Rekor via ECDSA-P256 and to Bitcoin OpenTimestamps via multi calendar submission. Both anchors are independent of Etch. An auditor can resolve any anchor with rekor-cli or with the OpenTimestamps client without contacting Etch.
Third party offline verification An auditor can walk the receipt chain without requiring the operator to be online.	**Full conformance**	etch-chain-verify is a standalone Python CLI shipped in the OSS package world-model-mcp on PyPI. Given a storage root the verifier walks the chain, checks row hash continuity, validates the hybrid signature on every closed epoch against pinned public keys, and reports on external anchor coverage. Air gapped environments are supported. No network access to Etch is required after the initial key pinning.

AARM property	Etch claim	Evidence path
Cross agent handoff Actions crossing agent boundaries preserve tamper evidence across the handoff.	**Full conformance**	POST /v1/etch-chain/cross-chain-reference records a cross chain reference by chain_id and epoch_seq and event_hash. The chain_id is deterministically derivable from the target project's identifier and the Etch genesis constant. The verifier supports offline mode which enumerates references and online mode which walks referenced chains in a local copy to validate the pair.
Retrospective governance revision The record supports formal retrospective revision including confidence downgrade for prior decisions.	**Full conformance**	POST /v1/etch-chain/postmortem records a signed postmortem that references a prior OSS event by id, records a finding, records a corrective action with its signer, and optionally records a retroactive confidence downgrade. The postmortem is a first class chain kind. The chain is append only at the SQL layer, so the original event is preserved even when the postmortem revises the assessment.
Runtime authorization decisions The five authorization decision classes (ALLOW, DENY, MODIFY, DEFER, and the block class) in the execution path.	**Outside scope**	Etch is an evidence layer, not a runtime enforcement layer. Etch does not sit in the execution path of an autonomous agent and does not claim conformance on this property. Etch integrates with enforcement vendors via POST /v1/import which accepts adapter output from otel-gen-ai, langsmith, cloudtrail, vercel-ai, and custom formats. When the enforcement vendor emits a decision (any of the five classes), the record of that decision can be signed onto the Etch chain.

AARM property	Etch claim	Evidence path
Least privilege posture The system supports enforcement of least privilege at the credential and tool call level.	Outside scope	Same reasoning as runtime authorization decisions. Etch is not in the enforcement path. Etch signs the record of what an enforcement vendor decided about privilege scope but does not enforce scope itself.
Session scoped disclosure An auditor can review a specific session without exposing the full chain.	Full conformance	The etch-chain-verify CLI accepts a session filter so the verifier walks only the rows associated with the named session. POST /v1/etch-chain/custody-export produces a session scoped self authenticating JSON bundle aligned with the Federal Rules of Evidence 902 and eIDAS regimes. The bundle can be recomputed and verified offline by any auditor with the source chain files.

4. Threat model coverage

The AARM specification enumerates a threat model for autonomous agents. Etch is an evidence layer, not a runtime enforcement layer. The table below is explicit about the division between what Etch helps investigate after the fact and what Etch does not detect or prevent at runtime.

Post-decision integrity threats

For this class of threat the signed chain records who did what under what authority. Attribution and non-repudiation are real. The receipt genuinely supports investigation after the fact.

AARM threat	Etch coverage posture
Confused deputy	Investigation. Bounded authority receipts and autonomy level metadata record the identity, scope, and autonomy of each acting party. An investigator can identify actions that fell outside the bounded authority claim at write time.
Cross agent propagation	Investigation. Cross chain reference records references across chain boundaries. When two cooperating chains are both on Etch, an investigator can walk the cross reference forward or backward to identify the point at which a compromised action crossed into a second agent.
Data exfiltration	Investigation. The record of an exfiltration event is signed onto the chain like any other event. The signed record binds the exfiltration to the acting identity, credentials, and authority scope at write time. Etch does not prevent exfiltration at runtime.
Over privileged credentials	Investigation. HSM attestation records the credential material's hardware anchoring. Bounded authority receipts record the scope claimed for each acting party. An investigator can identify actions taken under credentials whose scope was broader than the authority receipt attested. Etch does not restrict credential scope at runtime.

Pre-decision corruption threats

For this class of threat the receipt records the reasoning the agent provided for its decision. When the reasoning was constructed from corrupted upstream state, the record faithfully preserves the corrupted reasoning as the audit-visible cause. Auditability buys

attribution and non-repudiation. It does not buy correctness of the pre-decision state. Detecting this class requires upstream integrity controls (signed memory stores, provenance over demonstration and retrieval artifacts, tool sandboxing, environmental integrity checks) that sit outside the evidence layer's boundary.

AARM threat	Etch coverage posture
Memory poisoning	Outside receipt-layer scope. The receipt records the memory content the agent used to reach its decision. When that memory was poisoned upstream of the recording surface, the chain faithfully preserves the poisoned content as the audit-visible input. Detection requires two distinct upstream disciplines that should not be confused. Cryptographic provenance (signed memory writes, content-addressed retrieval artifacts) addresses MODIFICATION of stored content after ingest. It does not address POISONING AT SOURCE. A signed malicious document remains malicious. The signature proves origin, not truth. Detection at source requires editorial or curatorial controls over the ingest pipeline itself, which is a governance discipline separate from cryptographic provenance.
Goal hijacking	Outside receipt-layer scope. When the agent's goal has been hijacked, the reasoning the agent provides on the chain reflects the hijacked goal as its stated intent. The record preserves the hijacked reasoning as legitimate. Detection requires upstream goal-integrity controls (signed goal declarations, human sign-off gates before goal changes) outside the evidence layer.
Intent drift	Partial coverage. The eight category drift detection engine measures shifts in policy, assumptions, confidence, evidence, scope, terminology, authority, and context across sessions. It detects RECORDED drift. It does not detect intent drift that occurred before the intent was recorded onto the chain.
Context accumulation	Outside receipt-layer scope. Continuity tracking can trace the recorded context an agent claimed to use. It cannot verify that the recorded context is a faithful representation of the actual pre-decision state. Detection follows the same two-discipline pattern as memory poisoning. Signed context-store snapshots at each decision detect MODIFICATION of the accumulated context after it was captured. They do not detect CORRUPTION AT SOURCE. Detection at source requires editorial or curatorial controls over what enters the context pipeline in the first place, which is a governance discipline separate from cryptographic provenance.

AARM threat	Etch coverage posture
Malicious tool output	Outside receipt-layer scope. The tool output is recorded on the chain and hash-committed at write time. If the output is malicious and becomes input to a subsequent decision, the record faithfully preserves the malicious content as legitimate input. Detection requires upstream output-integrity controls (tool sandboxing, output filters) outside the evidence layer.
Environmental manipulation	Outside receipt-layer scope. Environmental context recorded in the governance record is the agent's report of what it observed. When the environment was manipulated before the observation, the record preserves the manipulated view as the agent's ground truth. Detection requires environmental integrity controls outside the evidence layer.

5. Verifier resolution rules

The AARM specification does not fix a resolution rule between property 1 (Tamper evident action receipt) and property 6 (Retrospective governance revision). Two conformant implementations can therefore present different current-state reads of the same ledger. This section documents Etch's implementation choices explicitly so procurement teams evaluating multiple AARM-conformant vendors can compare read-path semantics side by side.

Supersession chain read semantics

etch-chain-verify walks the full supersession chain by default and returns the resulting DAG. There is no 'current state' query that returns only the latest head. A caller asking 'what does this decision say now' receives the head of the DAG plus the walkable history that produced it. The AARM specification does not fix a rule here. This is an implementation choice. Two AARM-conformant implementations may make different choices and present different current-state reads of the same ledger. Procurement teams comparing vendors should verify this rule explicitly.

Retrospective revision integrity

A postmortem event (property 6) is appended to the chain like any other event and inherits property 1 tamper evidence. The prior event it references remains unchanged in the log. The postmortem does not overwrite, replace, or mutate the prior event. The verifier resolves the pair by returning both, with the postmortem linked to its target via the about_event_id field.

Confidence downgrade semantics

A postmortem may carry an optional retroactive_confidence field. The verifier surfaces the retroactive confidence alongside the original decision's stated confidence rather than replacing it. A caller sees both values. This preserves the auditability of the original decision-time confidence claim while making the revised assessment walkable.

Canonical collapse for current-state queries

The DAG-only rule above keeps the audit surface honest but leaves multiple consumers of the same ledger free to collapse the DAG differently, which relocates the divergence rather than removing it. For callers that need a single current-state answer (dashboards, procurement reviews, downstream systems), Etch documents a canonical collapse rule consumers MAY apply. The rule is keyed on the supersession intent field (see property 6 retrospective governance revision). For intent values 'correction', 'breaking', and 'refinement', the head of the DAG is the canonical current-state and the prior decision is

superseded. For intent 'deprecation', the original decision stands with a 'deprecated' flag, and the deprecation itself is advisory context. For intent 'compat', the original decision stands and the supersession is additive context rather than a replacement. When two supersession events target the same decision at identical timestamps, tie-break by highest chain sequence number. The rule is deterministic. Two consumers who apply the canonical collapse to the same ledger arrive at the same collapsed state. The full DAG remains available to any consumer that prefers to walk the evidence rather than accept the collapse. This is Etch's implementation choice and is offered as one concrete proposal to the AARM working group.

Intent-attester-controlled floor on canonical collapse

The canonical collapse rule above is deterministic given the supersession intent field. That field is attested by whoever performs the supersession. A verifier cannot distinguish an honest deprecation from a strategic mislabel. The determinism the rule claims is therefore determinism given trust in the attester's intent classification, not determinism without further assumptions. To make gaming the intent field visible without shifting the classification authority to a role that in practice cannot exist, the verifier surfaces the distribution of intent labels across a chain (intent-mix report). A chain whose owner has called 90 percent of supersessions 'compatible refinement' while a peer baseline sits at 10 percent is a signal for external review, not a verifier judgment on any single supersession. A count, not a judgment. This is evidence, not enforcement.

Concurrent supersession fail-loud rule

The canonical collapse rule and its intent-mix floor resolve supersession along one line of causation. In a multi-actor deployment, two actors may supersede the same prior decision independently, producing concurrent heads that cannot be linearized without an additional rule. The verifier does not linearize concurrent heads and does not pick one. Canonical current-state for the affected decision is undefined until the consumer explicitly resolves the fork. The verifier surfaces the concurrent-heads condition and refuses to return a single collapsed state for that decision. Silent collapse under undefined semantics is exactly the class of overclaim the property-1 tamper evidence guarantee was built to reject. Failing loudly is the correct failure direction.

Receipt outcome enum is required

A receipt schema that only records completed calls proves what got through and is silent about what was refused. Etch requires every decision-shaped event to name its outcome as one of 'allowed', 'denied', or 'abstained', and to carry the reason for a denial or abstention as a first-class in-band field alongside requester, action, and target. Refused calls are enforcement-layer output. The record of the refusal is evidence-layer surface and is signed with the same integrity guarantee as an allowed call. An independent verifier walking the chain can count refused-to-allowed ratios by requester and by action, which is

a stronger audit claim than reading a completed-only log. Without this, the audit trail collapses to the operator's own word about what was refused.

Effect verification is a candidate AARM property

AARM as currently written verifies that a decision was made under the declared policy and signed under the declared authority. It does NOT verify that the downstream state-change the decision authorized actually took effect in the outside world. A future AARM property revision cycle SHOULD add an effect-verification requirement: a signed record that the authorized effect landed (a database row exists, an email was delivered, an API call was accepted). Etch surfaces this as an open extension because a decision-only audit trail is silent about the difference between an authorized-but-failed action and an authorized-and-completed action, and a regulator asking 'did the effect happen' has to trust the operator's own reporting for the answer. Property surfaced here so the working group can score the shape at the next revision cycle.

Schema-injection justification is required for evidence writers

Every field emitted onto the audit chain MUST have a documented justification for its presence in the signed shape. Fields added without a corresponding justification create surface area for an attacker to inject payloads that a future consumer will trust as evidence. The Etch chain applies this by requiring every new chain kind to ship a signed shape (pydantic model), a bounded enum for any discriminator, and a documented what-this-field-proves note in the release commit. External verifiers walking the chain rely on the justified-field invariant to reason about which fields are load-bearing for their audit claim and which are informational. Prior art: arXiv:2606.04990 argues that schema-injection attacks against evidence layers are structurally different from prompt-injection because the receiver is a verifier not a model, and the receiver's trust boundary is the schema itself.

Fresh authorization at the tool boundary is required

A tool_call event carries an authorization_event_id_ref field that MUST point at an authority_receipt or governance_record chain event landed within a staleness window that precedes the call. The default window is fifteen minutes. Deployments that require stricter latency MAY narrow it in configuration and MUST NOT widen it beyond one hour without a signed governance record that explains the exception. A tool_call with no authorization_event_id_ref, or one that references an event outside the window, is either unauthorized or authorized against stale credentials. Both cases collapse the audit trail in the same direction. The operator cannot show a regulator that the caller was authorized under a specific policy at the specific moment the call fired. Etch's verifier walks the chain and reports every tool_call whose authorization link is missing or exceeds the window, and the chain-of-custody export inherits that walk so a regulator sees the same report without contacting the operator.

6. Statement of intent

Etch intends to complete the AARM Conformance Testing Protocol for Extended (R1 through R9) once the working group's evidence package template stabilizes. The evidence package will cover the full requirement text, the reproducible test procedure for each requirement, and the chain of custody for the test artifacts. This document is a public posture statement pending that formal result.

Any operational or specification change that would affect the correspondence table above triggers a revision of this statement. The revision history is maintained at etch.systems/aarm and in the release notes for the affected Etch version.

7. References

1. Errico, H. (2026). Autonomous Action Runtime Management (AARM). Cloud Security Alliance.
   https://aarm.dev/spec
2. Errico, H. (2026). AARM Specification (preprint). arXiv:2602.09433 [cs.CR].
   https://arxiv.org/abs/2602.09433
3. Bradner, S. (1997). Key words for use in RFCs to Indicate Requirement Levels. IETF RFC 2119.
   https://www.rfc-editor.org/rfc/rfc2119
4. NIST FIPS 205. Stateless Hash Based Digital Signature Standard. 2024.
   https://csrc.nist.gov/pubs/fips/205/final
5. NIST FIPS 186-5. Digital Signature Standard. 2023.
   https://csrc.nist.gov/pubs/fips/186-5/final
6. Wang, Y. et al. (2026). From Agent Traces to Trust: A Survey of Evidence Tracing and Execution Provenance in LLM Agents (preprint). arXiv:2606.04990 [cs.CR].
   https://arxiv.org/abs/2606.04990

8. Contact

Compliance verification queries and requests for the underlying evidence package should be directed to the operator email on file at etch.systems. Correspondence regarding this statement is welcome from the AARM working group and from compliance offices evaluating vendors on the AARM axis.

9. Revision history

> This statement is iterated in public in response to peer review the conformance page was designed to attract. Each revision below is prompted by a specific published critique and shipped same-day or the following morning. The chain of critique to reply to code change to deploy is recorded on the Etch chain and independently verifiable. This is the receipt-shape discipline the standard was designed to attract, applied to the standard's own conformance page.

Every change to a claim or evidence path in this statement is recorded below. The current version's changes appear first.

Version	Date	Change
1.4	2026-08-06	Four content additions prompted by continued public review critique the conformance page was designed to attract. First, an intro paragraph framing the revision history as the receipt-shape demonstration the page was designed to attract, so a reader sees four same-day revisions as intentional discipline rather than sloppy first draft. Second, an intent-attester-controlled floor added to the canonical collapse rule (v1.3), naming the verifier's inability to distinguish an honest deprecation from a strategic mislabel and specifying an intent-mix report as the evidence-not-enforcement mitigation. Third, a concurrent-supersession fail-loud rule for the case where two actors supersede the same decision independently in a multi-actor deployment. The verifier does not linearize concurrent heads and refuses to return a single collapsed state until the consumer resolves the fork. Silent linearization would be exactly the class of overclaim property 1 was built to reject. Fourth, the receipt schema now names the outcome (allowed / denied / abstained) as required and carries denial reason as a first-class in-band field. A completed-only log collapses to the operator's own word about what was refused.
1.3	2026-08-05	Two refinements to v1.2 prompted by a further round of public review critique the conformance page was designed to attract. First, the remedy text on the pre-decision corruption threats quietly inherited the same class of overclaim the v1.2 category split was supposed to remove. Signed memory stores and content-addressed provenance detect MODIFICATION of stored content, not CORRUPTION at source. A signed malicious document remains malicious. The signatureproves origin, not truth. The memory poisoning and context accumulation rows are tightened to name this distinction explicitly and to identify the separate discipline

Version	Date	Change
		(editorial or curatorial control over the ingest pipeline) that would actually address the poisoning-at-source failure mode. Second, the verifier resolution rules section is extended with a canonical collapse rule for current-state queries over supersession chains. The v1.2 rule (DAG-only, no current-state query) was defensible but relocated implementation-time divergence to consumer-time divergence rather than removing it. The v1.3 addition specifies a canonical collapse consumers MAY apply, keyed on supersession intent, so multiple consumers of the same ledger arrive at the same collapsed state.
1.2	2026-08-05	Threat coverage table split into two categories with distinct coverage semantics: post-decision integrity threats (confused deputy, over privileged credentials, data exfiltration, cross agent propagation) where the signed receipt genuinely supports investigation via attribution and non repudiation, and pre-decision corruption threats (memory poisoning, goal hijacking, intent drift, context accumulation, malicious tool output, environmental manipulation) where the receipt faithfully preserves corrupted upstream state as the audit-visible cause and the receipt layer alone cannot verify correctness of the pre-decision state. The prior version 1.1 label of 'Investigation' for all ten threats overstated what the primitive can do for the pre-decision corruption class. Also added a new section 9 documenting Etch's verifier resolution rule for supersession chains, since the AARM specification does not fix a resolution rule between properties 1 and 6 and two conformant implementations could present different current-state reads of the same ledger. Change prompted by a public review critique the conformance page was designed to attract.
1.1	2026-08-05	Property 1 (Tamper evident action receipt) downgraded from Full conformance to Conditional conformance. The chain plus external anchor catches receipts that were altered, deleted, or fabricated after being written, but cannot detect actions that were never routed through the recording surface at all. Ensuring every autonomous action reaches the recording surface is a property of the credential boundary the operator deploys, which AARM property 8 (Least privilege posture) addresses. The conditional pass on property 1 therefore depends on the deploying organization enforcing property 8 within its deployment. This distinction was implicit in version 1.0 and is made explicit here. Change prompted by a public review critique the conformance page was designed to attract.
1.0	2026-08-02	

Version	Date	Change
		Initial statement. Full conformance claim on the seven evidence layer properties. Runtime authorization and least privilege posture marked outside scope.