CTS-1

MAI-1 Conformance Test Suite

Public Specification

Auburn Patent Family

Document: AGS-DOC-026-PUB

Version: 1.0

Classification: **PUBLIC**

Date: 2026

Author: Ryan Fields

This document is the public specification of the CTS-1 Conformance Test Suite for the Model Attestation Interface (MAI-1). It defines the assertion registry, test pyramid architecture, conformance methodology, and pass/fail criteria for all 45 assertions across three conformance levels. The automated execution framework, report schema, certification workflow, and detailed test evidence procedures are contained in the private specification (AGS-DOC-026)

Contents

1 Executive Summary

1.1 Purpose

CTS-1 is the Conformance Test Suite for the Model Attestation Interface (MAI-1). It transforms MAI-1 from a normative specification into an enforceable standard by defining binary pass/fail tests for every mandatory requirement.

Without CTS-1, MAI-1 is a document. With CTS-1, MAI-1 is a gate.

CTS-1 accomplishes this through six capabilities:

1. **Derived Test Requirements (DTR).** Every **MUST**/**SHALL** statement in MAI-1 §6–§9 is decomposed into a numbered assertion with explicit pass/fail criteria. There are no subjective evaluations and no maturity scores.

2. **Binary conformance.** Every test produces one of five verdicts: PASS, FAIL, WARNING, REVIEW, or SKIPPED. The overall determination is binary: a system either passes or fails at its claimed conformance level. There is no partial compliance.

3. **Tiered conformance levels.** Three levels—MAI-C0 (self-certification), MAI-C1 (independent verification), MAI-C2 (accredited third-party evaluation)—provide a graduated trust model with cumulative requirements. Each level specifies who runs the tests, who signs the results, and what is trusted.

4. **Negative test majority.** More than 50% of the test corpus consists of negative test cases: malformed artifacts, cryptographic attacks, temporal replay, semantic inconsistencies, and behavioral state manipulation. A system that produces correct output under normal conditions is useful. A system that fails safely under adversarial conditions is secure.

5. **Statistical invariant validation.** At MAI-C2, the five mandatory invariants are evaluated using sequential hypothesis testing at 99% confidence over sustained observation windows.

6. **Outsider executability.** CTS-1 is executable by any party with access to a system's MAI-1 attestation endpoint. It does not require vendor cooperation, proprietary tooling, or privileged credentials. Failure to pass CTS-1 at the claimed conformance level constitutes objective evidence that the system was operating outside its declared governance state at the time of testing.

1.2 Enforcement Capability

CTS-1 is designed to be consumed by five audiences, none of whom require the tested organization's cooperation to act on the results:

- **Procurement officers** can require "MAI-C1 PASS via CTS-1" as a procurement eligibility condition, copy-pasted directly into RFP language.
- **Insurance underwriters** can gate coverage on CTS-1 conformance level, with receipt validity windows aligned to policy renewal cycles.
- **Regulators** can reference CTS-1 results as conformity assessment evidence under the EU AI Act, FDA SaMD guidance, and financial services regulations.
- **Enterprise risk managers** can verify vendor governance claims independently, without relying on vendor-supplied attestations of their own compliance.
- **Researchers** can use the CTS-1 framework to evaluate the governance maturity of AI systems in empirical studies.

Once a binary, outsider-executable test exists, the strategic calculation for every organization deploying AI systems changes. The question is no longer "should we adopt governance infrastructure?" but "do we pass right now?"

1.3 Document Scope: Public vs. Private

This document is the **public specification** of CTS-1. It contains:

- The complete assertion registry (all 45 assertions with pass/fail criteria).
- The conformance methodology and test pyramid architecture.
- The conformance level comparison and test organization.
- The negative test library (attack categories and coverage targets).
- The statistical methodology overview.
- The honest limitations of what CTS-1 can and cannot guarantee.

Private Version Content

The **private specification** (AGS-DOC-026) additionally contains:

- Complete Vendor Evidence (VE) requirements for each assertion.
- Step-by-step Test Evidence (TE) procedures with implementation-level detail.
- The automated test execution framework architecture.
- The machine-readable report schema (CDDL).
- The certification workflow and conformance claim state machine.
- The self-certification vs. third-party evaluation model.
- The maintenance governance and non-weakening clause.
- The Request for Guidance (RFG) change management process.
- The negative test vector binary data and cross-reference index.
- The complete MAI-1 traceability matrix.

1.4 Summary Statistics

Table 1: CTS-1 at a Glance

Metric	Value
Total assertions	45
Estimated individual test cases	370+
Negative test base vectors	63 (hand-crafted) + 10,000+ (grammar-aware fuzzing at C2)
Conformance levels	3 (MAI-C0, MAI-C1, MAI-C2)
Statistical tests (C2)	5 (TSPRT at 99% confidence)
Test pyramid levels	4 (Structural, Cryptographic, Semantic, Behavioral)
MAI-1 requirement coverage	100% of **MUST**/**SHALL** in §6–§9
Normative language	RFC 2119 / RFC 8174
MAI-1 target version	v1.0 (`mai-profile: "urn:auburn:mai:1.0"`)

2 Scope and Applicability

2.1 What CTS-1 Tests

CTS-1 evaluates whether an implementation claiming MAI-1 conformance produces attestation artifacts that satisfy the normative requirements defined in MAI-1 §6–§9. Specifically:

1. **Structural conformance** (§6): Attestation artifacts are well-formed CBOR, wrapped in COSE_Sign1 envelopes, and contain all mandatory fields defined in the MAI-1 CDDL profile.
2. **Cryptographic integrity** (§6.3): Signatures verify against declared certificate chains, algorithms are from the approved registry, and (at MAI-C1+) signing keys are bound to hardware Trusted Execution Environments.
3. **Semantic correctness** (§7): The five mandatory invariant measurements are within physically plausible ranges, certified thresholds are correctly structured, and the compliance flag derivation logic is independently verifiable.
4. **Behavioral validity** (§9, MAI-C1/C2): State transitions are correct across reboots, measurement frequency meets the minimum rate for the claimed conformance level, and breach detection/recovery operates correctly under simulated fault conditions.
5. **Conformance level requirements** (§9): The implementation satisfies all cumulative requirements for its claimed conformance level (MAI-C0, MAI-C1, or MAI-C2).

2.2 What CTS-1 Does Not Test

1. **Model quality or safety.** CTS-1 does not evaluate whether a model is accurate, safe, aligned, fair, or beneficial. An unsafe model that produces structurally perfect attestation artifacts will pass CTS-1. Conformance testing verifies the governance infrastructure, not the governed behavior.
2. **Claim truthfulness at MAI-C0.** Self-certified MAI-C0 conformance verifies artifact structure, not measurement truthfulness. MAI-C1 mitigates through TEE-rooted signing; MAI-C2 mitigates through adversarial testing and statistical validation.

3. **Organizational governance processes.** Risk management frameworks, ethical review boards, incident response procedures, and deployment policies are outside the scope of technical conformance testing.

4. **Regulatory compliance determination.** CTS-1 provides evidence that can be used in regulatory compliance assessments. A CTS-1 PASS is necessary but not sufficient for regulatory compliance; the mapping from CTS-1 results to specific regulatory requirements is defined in the Auburn Application Layer profiles (EU AI Act Profile, Federal AI Profile, etc.).

5. **Hardware security evaluation.** CTS-1 verifies that TEE attestation evidence is correctly consumed and reported within MAI-1 artifacts. It does not evaluate whether the TEE itself is secure against side-channel attacks, fault injection, or firmware exploits.

2.3 Relationship to MAI-1

CTS-1 implements the conformance framework defined in MAI-1 §9 in the same way that the FIPS 140-3 Derived Test Requirements (DTR) implement the security requirements defined in ISO/IEC 19790. MAI-1 is the normative specification; CTS-1 is the verification mechanism.

CTS-1 is applicable to MAI-1 v1.0 (`mai-profile: "urn:auburn:mai:1.0"`). Future MAI-1 versions will require corresponding CTS-1 updates, governed by the maintenance process defined in the private specification.

2.4 Normative Language

The key words **MUST**, **MUST NOT**, **SHALL**, **SHOULD**, **SHOULD NOT**, and **MAY** in this document are to be interpreted as described in RFC 2119 and RFC 8174 when, and only when, they appear in bold text as shown here.

2.5 Applicable Audiences

Table 2: CTS-1 Audience and Usage

Audience	Primary Use of This Document
Implementers	Understand what will be tested and the pass/fail criteria for each assertion. Use the assertion registry to prepare for conformance testing. The private specification provides the detailed test evidence procedures.
Conformance Testing Entities (CTEs)	Understand the assertion scope, test pyramid architecture, and conformance methodology. The private specification provides the automated execution framework and report schema required for test execution.
Procurement Officers	Reference the conformance level comparison (§6) to specify MAI-1 conformance requirements in RFPs. The assertion registry (§5) provides the technical basis for eligibility language.
Insurance Underwriters	Reference the conformance levels and receipt validity windows to structure coverage gating. The honest limitations (§9) define what CTS-1 does and does not guarantee.
Regulators	Reference CTS-1 as a conformity assessment mechanism. The conformance methodology (§3) documents the relationship to established certification traditions (Common Criteria, FIPS 140-3, IETF RATS).
Researchers	Use the assertion registry and test pyramid architecture to evaluate AI governance maturity in empirical studies.

3 Conformance Methodology: The Four-Pillar Synthesis

CTS-1 derives its methodology from four established conformance traditions, each selected for a specific structural contribution. No single tradition is sufficient: Common Criteria provides the assurance hierarchy but imposes prohibitive cost and timeline constraints. FIPS 140-3 provides the requirement-to-test decomposition methodology but suffers from severe manual process bottlenecks. The OpenID Foundation provides the self-certification economics that enable broad adoption but cannot verify claim truthfulness. The IETF RATS working group provides the protocol-level interoperability evidence but lacks a formal conformance suite.

CTS-1 synthesizes these traditions into a unified conformance engine. This section details the structural borrowings, the anti-patterns avoided, and the specific adaptations that make each tradition applicable to AI attestation governance.

3.1 Pillar 1: Common Criteria — Tiered Assurance and the Protection Profile Model

3.1.1 The Evaluation Assurance Level Hierarchy

The Common Criteria for Information Technology Security Evaluation (ISO/IEC 15408) and its companion Common Evaluation Methodology (CEM, ISO/IEC 18045) define the global benchmark for formal security product evaluation. The central structural contribution is the Evaluation Assurance Level (EAL) hierarchy: a graduated scale from EAL1 (functionally tested) through EAL7 (formally verified) that measures not the *strength* of security features but the *confidence* that those features are correctly implemented and resistant to bypassing.

This distinction is critical for CTS-1. The test suite must validate *assurance*—the correctness of the attestation implementation's logic, encoding, cryptographic operations, and behavioral

responses—not merely the *presence* of claims in a token. A system that includes an entropy field in its attestation artifact is not necessarily a system that correctly measures entropy, correctly encodes the measurement, or correctly responds when entropy falls below the certified threshold.

The EAL hierarchy maps to CTS-1's testing depth at each MAI-1 conformance level:

CC Level	**Assurance Scope**	**Testing Depth**	**CTS-1 / MAI Analog**
EAL 1	Functionally Tested	Black-box validation: the evaluator confirms the system functions as documented. No internal design review.	**MAI-C0 (Basic Integrity):** CTS-1 performs black-box validation of the attestation endpoint. It verifies that a request yields a properly formatted, signed response with all mandatory fields present.
EAL 2	Structurally Tested	High-level design review. Vulnerability analysis based on public domain searches.	**MAI-C0+ (Structural Integrity):** CTS-1 requires evidence of internal structure (separation of Attester/Verifier roles). Verifies EAT token structure matches the claimed MAI-1 profile.
EAL 3	Methodically Tested	Gray-box testing with coverage analysis. The test plan must exercise key security functions.	**MAI-C1 (Composition Integrity):** CTS-1 enforces coverage by validating all branches of the attestation policy logic. Requires negative testing of all error conditions defined in the MAI-1 RATS profile.
EAL 4	Methodically Designed, Tested, and Reviewed	White-box testing. Requires low-level design documentation and source code subset review.	**MAI-C2 (Adversarial Integrity):** CTS-1 requires white-box visibility via diagnostic interfaces during testing. Invariant breach simulation, replay attack testing, tamper detection, and behavioral state validation.
EAL 5–7	Semiformal / Formal Verification	Mathematical proof of design-to-implementation correspondence.	**Out of Scope:** Formal verification is cost-prohibitive for broad adoption. CTS-1 targets EAL4-equivalent rigor through automated adversarial testing rather than mathematical proof.

The primary takeaway from the EAL hierarchy for CTS-1 is the progression from *functional testing* (Does the endpoint respond?) to *structural testing* (Is the response built correctly?) to *methodical testing* (Have we checked for failures?) to *adversarial testing* (Does it fail safely under attack?). CTS-1 **MUST NOT** stop at functional success; it **MUST** methodically probe the failure modes of the MAI-1 implementation.

3.1.2 The Protection Profile / Security Target Separation

Common Criteria uses two artifacts to define evaluation requirements. The **Protection Profile (PP)** defines abstract security requirements for a class of technology (e.g., "Network Firewalls"). The **Security Target (ST)** is a vendor's specific implementation claim against a Protection Profile.

CTS-1 operationalizes this separation:

CTS-1 Protection Profile / Security Target Model

- **MAI-1 as Protection Profile.** The MAI-1 specification acts as the Protection Profile. It defines the mandatory security functional requirements that all implementations must meet: COSE signing, nonce freshness, field completeness, invariant reporting, compliance flag logic, TEE key binding (MAI-C1+), adversarial resilience (MAI-C2).
- **Implementation Conformance Statement as Security Target.** When a vendor submits for CTS-1 testing, they **MUST** supply an Implementation Conformance Statement (ICS)—a structured declaration of their specific capabilities: supported COSE algorithms, TEE type and manufacturer, supported conformance level, optional extensions implemented, Layer 3 provenance mechanisms, and platform-specific configuration.
- **CTS-1 as Universal Evaluator.** CTS-1 reads the vendor's ICS, verifies it meets the MAI-1 specification for the claimed conformance level, and executes the applicable test subset to prove the ICS claims are true.

This model moves the Protection Profile / Security Target mapping from a paper exercise requiring human evaluators to a configuration-driven automated test plan. The ICS is a machine-readable document; CTS-1 parses it and generates the test execution plan automatically.

3.1.3 The "Evaluated Configuration" Anti-Pattern

The most significant structural weakness of Common Criteria—and the primary trap CTS-1 must avoid—is the "evaluated configuration" divergence. In the CC framework, a product is certified in a specific, frozen state: exact firmware version, specific patches, rigid configuration settings. This creates three pathologies:

1. **The Patching Paradox.** A certified system that receives a critical security patch moves outside its evaluated configuration, potentially voiding its certification. The vendor must choose between being secure (patched) or compliant (certified).
2. **The Deployment Gap.** Products are often certified in configurations that no production deployment actually uses (e.g., with advanced features disabled to simplify testing). The assurance statement applies to a configuration that does not exist in the field.
3. **The Obsolescence Problem.** Evaluations at EAL4 require 7–24 months and cost $80,000–$750,000+. By the time a certificate is issued, the product version is often obsolete.

For AI systems—where model weights change through fine-tuning, inference infrastructure evolves continuously, and governance requirements escalate in response to incident discovery—the evaluated configuration model is fundamentally incompatible.

CTS-1 Continuous Conformance Model

CTS-1 inverts the Common Criteria model through three design decisions:

1. **Dynamic Certification.** Instead of a static PDF certificate issued every two years, CTS-1 generates a cryptographically signed attestation receipt for every test run. The "certificate" is a machine-readable report, not a document.
2. **CI/CD Integration.** The "evaluated configuration" becomes the current build. CTS-1 is designed to execute within the vendor's standard CI/CD pipeline. Every commit, release candidate, or deployment event can trigger a CTS-1 run.
3. **Freshness-Bounded Validity.** The validity of a conformance claim is tied to the freshness of the CTS-1 receipt. A system claiming MAI-1 conformance **MUST** produce a receipt dated within the applicable freshness window:
   - MAI-C0: Receipt validity $\leq$ 90 days (quarterly re-testing).
   - MAI-C1: Receipt validity $\leq$ 30 days (monthly re-testing).
   - MAI-C2: Receipt validity $\leq$ 7 days (weekly re-testing), with continuous automated monitoring between test runs.

These freshness requirements ensure that conformance claims track the actual state of the deployed system rather than a historical snapshot. The re-testing cost is minimal because CTS-1 is automated; the overhead is compute time, not human evaluator time.

3.2 Pillar 2: FIPS 140-3 — Derived Test Requirements and Binary Precision

3.2.1 The DTR Architecture

FIPS 140-3 (ISO/IEC 19790), administered through the NIST Cryptographic Module Validation Program (CMVP), provides the most rigorous publicly documented methodology for translating normative requirements into executable tests. The central artifact is the Derived Test Requirements (DTR) document (NIST SP 800-140 series), which decomposes every normative "shall" statement into three executable components:

DTR Component	Definition
Assertion (AS)	The normative statement, quoted verbatim from the standard with a systematic identifier (e.g., AS02.15). The assertion is the *what*: what the implementation must do.
Vendor Evidence (VE)	What the implementer must provide to demonstrate the assertion is met. Documentation, configuration files, design descriptions, sample outputs. The vendor evidence is the *show*: show that you designed for this requirement.
Test Evidence (TE)	What the evaluator must execute to verify the assertion independently. Step-by-step test procedures with specific inputs, expected outputs, and pass/fail criteria. The test evidence is the *prove*: prove that the implementation actually does what the vendor claims.

SP 800-140 explicitly requires that "a yes/no answer does not provide sufficient assurance." Every test response must cite documentation, describe the test method and tools used, and summarize detailed results. This standard of evidence is what separates a conformance test from a checkbox exercise.

3.2.2 DTR Adaptation for CTS-1

CTS-1 adopts the DTR methodology as its core structural pattern. Every normative **MUST** and **SHALL** statement in MAI-1 §6–§9 is decomposed into a numbered assertion with corresponding vendor evidence requirements and test evidence procedures. The complete DTR registry is specified in §5 of this document.

The CTS-1 DTR adaptation introduces four extensions to the FIPS pattern:

1. **Tier Scoping.** Each assertion is explicitly scoped to applicable conformance levels. An assertion tagged `[C0, C1, C2]` applies at all levels. An assertion tagged `[C1, C2]` applies only at MAI-C1 and above. An assertion tagged `[C2]` applies only at the highest assurance tier. This prevents MAI-C0 implementations from being tested against MAI-C2 requirements.

2. **Test Pyramid Level.** Each assertion is assigned to a level in the CTS-1 test pyramid (§4): `L1-STRUCTURAL`, `L2-SEMANTIC`, `L3-BEHAVIORAL`, or `L4-CRYPTO`. This determines execution order and gate logic—structural failures block all subsequent testing.

3. **Test Type Classification.** Each assertion is classified as `DETERMINISTIC` (the result is binary and reproducible given the same input) or `STATISTICAL` (the result requires multiple samples and a statistical decision rule). Statistical tests carry additional metadata: minimum sample size, confidence level, and decision rule parameters.

4. **Assertion Identifier Convention.** CTS-1 assertions use a four-part identifier:

```
AS-{PyramidLevel}-{Layer}-{Sequence}

Examples:
  AS-ST-L1-01    Structural gate, Layer 1, assertion 01
  AS-CR-L1-03    Cryptographic gate, Layer 1, assertion 03
  AS-SM-L2-07    Semantic gate, Layer 2, assertion 07
  AS-BH-L2-02    Behavioral gate, Layer 2, assertion 02
```

3.2.3 Example DTR Decomposition

The following example demonstrates the DTR pattern applied to a specific MAI-1 requirement:

Example: DTR Decomposition of MAI-1 Requirement 6.1

MAI-1 Source: Requirement 6.1 (Encoding). "MAI-1 attestation artifacts **MUST** be encoded in CBOR (RFC 8949) and signed using COSE_Sign1 (RFC 9052)."

Assertion: `AS-ST-EP-03` — The MAI-1 attestation artifact **SHALL** be a valid CBOR-encoded payload wrapped in a COSE_Sign1 envelope per RFC 9052. `[C0, C1, C2]` `L1-STRUCTURAL` `DETERMINISTIC`

Pass Criteria: The outermost structure is a COSE_Sign1 (`CBOR Tag 18`). The protected header is a `bstr` containing a valid CBOR-encoded header map. The payload is a `bstr` containing a valid CBOR-encoded EAT Claims-Set. Canonical CBOR encoding rules per RFC 8949 §4.2.1 are satisfied.

Fail Criteria: Any structural check fails. The artifact is not parseable as CBOR, the COSE_Sign1 tag is absent, the protected header is malformed, the payload is not a valid EAT Claims-Set, or canonical encoding rules are violated.

Private Version Content

The complete Vendor Evidence (VE) and step-by-step Test Evidence (TE) procedures for each assertion are specified in the private specification (AGS-DOC-026). The public specification provides assertion identifiers, normative text, tier applicability, and pass/fail criteria summaries for all 45 assertions.

3.2.4 The FIPS Timeline Anti-Pattern

While the DTR methodology is CTS-1's most important structural borrowing, the CMVP's operational pathology must be actively avoided. As of early 2024, only approximately 19 full FIPS 140-3 validations had been completed, with average queue times exceeding 100 days before review even begins. NIST's own assessment describes the program as "heavily manual, out of sync with the speed of technology development."

CTS-1 avoids this bottleneck through automation-first design: every test procedure is designed for automated execution, test data is machine-encoded and machine-comparable, pass/fail criteria are evaluated programmatically, and test results are encoded in a machine-readable schema. Human judgment is required only for the REVIEW verdict category.

3.3 Pillar 3: OpenID Foundation — Self-Certification Economics and Operational Agility

3.3.1 The OIDF Conformance Model

The OpenID Foundation (OIDF) conformance testing program demonstrates that low-cost, automated conformance testing drives broad adoption far more effectively than expensive third-party evaluation. The program's operational characteristics provide the template for CTS-1's deployment model:

OIDF Characteristic	CTS-1 Adaptation
Test suite is free and open-source	CTS-1 public test harness is open-source, containerized (Docker/OCI), and freely downloadable.
Certification fee: $700/deployment/year (members), $3,500 (non-members)	CTS-1 MAI-C0 self-certification: nominal fee structure. MAI-C1/C2: graduated fees reflecting assurance depth.
Self-certification: vendor runs tests, submits signed logs	CTS-1 MAI-C0: self-certification permitted. Vendor executes the test suite, signs the result report, and publishes to the Auburn transparency log.
Hundreds of certified implementations	CTS-1's low-cost model is designed to achieve similar breadth. The economic barrier to conformance must be lower than the reputational cost of non-conformance.
38+ jurisdictions accept OIDF certifications	CTS-1 results are mapped to seven regulatory frameworks via MAI-1 §10. Regulatory acceptance is driven by the sector-specific profiles in the Auburn Application Layer.

3.3.2 The Dual-Role Test Harness

The OIDF conformance suite operates as a *reactive test harness*: it acts as the counter-party in the protocol exchange. CTS-1 adapts this architecture to the RATS attestation model:

CTS-1 Dual-Role Test Harness

- **CTS-1 as Universal Verifier (Primary Mode).** When testing an Attester (the system running MAI-1), CTS-1 generates challenge nonces, sends attestation requests to the canonical endpoint (`POST /v1/mai/attest`), receives the signed EAT token, and validates it through all four pyramid levels. This is the primary conformance testing mode.
- **CTS-1 as Reference Attester (Verifier Testing Mode).** When testing a Verifier (the system appraising MAI-1 evidence), CTS-1 generates known-good and known-bad attestation tokens from the test vector library and submits them to the Verifier. The Verifier must accept valid tokens, reject invalid tokens, and produce correct Entity Attestation Results (EAR) for each.

3.3.3 The Five-Verdict Model

OIDF's conformance suite produces five possible verdicts per test case. CTS-1 adopts this model with AI-attestation-specific semantics:

Verdict	Meaning	CTS-1 Semantics
PASS	Test passed.	The implementation satisfies the assertion for this test case. Automated determination; no human review required.
FAIL	Test failed.	The implementation does not satisfy the assertion. Conformance at the claimed level is denied. A single FAIL verdict at any applicable assertion is sufficient to deny conformance (binary compliance).
WARNING	Non-critical deviation.	The implementation deviates from a **SHOULD** recommendation but not a **MUST** requirement. Warnings are logged and reported but do not block conformance.
REVIEW	Human judgment required.	The automated test cannot determine pass/fail. A Conformance Testing Entity (CTE) analyst must review the evidence and issue a manual determination. REVIEW verdicts are permitted only at MAI-C1 and MAI-C2, where a CTE is required. At MAI-C0, all tests must resolve to PASS, FAIL, WARNING, or SKIPPED.
SKIPPED	Test not applicable.	The test does not apply to this implementation based on the ICS declaration.

Requirement 3.1 (Verdict Aggregation)

The overall conformance determination for a given level is computed as follows:

1. If *any* applicable assertion has a FAIL verdict, the overall determination is FAIL. No exceptions.
2. If *any* applicable assertion has a REVIEW verdict that has not been resolved by a CTE analyst, the overall determination is PENDING.
3. If all applicable assertions are PASS, WARNING, or SKIPPED, and no REVIEW verdicts are unresolved, the overall determination is PASS.
4. The number and nature of WARNING verdicts **SHALL** be reported in the conformance artifact. Governance Authorities **MAY** establish warning thresholds above which a PASS determination is subject to additional review.

3.3.4 The Truthfulness Limitation and the Graduated Trust Model

A critical distinction limits direct transfer of the OIDF model to AI attestation. OIDF tests verify *protocol behavior*—the software implementation is the thing being tested. AI attestation verifies *claims about external processes*—training methodology, model integrity, measurement accuracy. Self-certification can verify that an attestation artifact is well-formed and structurally conformant. It cannot verify that the claims within the attestation are truthful.

CTS-1 addresses this through a graduated trust model:

Level	Assessment Model	Trust Basis
MAI-C0	Automated self-certification	Artifact conformance only. The attestation is structurally and cryptographically valid. No claim truthfulness verification. Suitable for R&D environments.
MAI-C1	Assisted self-certification with independent CTE	TEE-rooted signing provides hardware-backed assurance that measurements were computed by verified code on verified hardware. Third-party verification eliminates self-grading. Moderate claim truthfulness assurance.
MAI-C2	Third-party evaluation by accredited CTE	Adversarial testing and behavioral validation probe whether the system behaves consistently with its attestation claims under stress. Invariant breach simulation verifies detection and response. Maximum claim truthfulness assurance achievable through automated testing.

3.4 Pillar 4: IETF RATS — Protocol Foundations and Interoperability Evidence

3.4.1 The RATS Protocol Stack

MAI-1 is constructed as a RATS profile—a constrained application of the Entity Attestation Token (EAT) standard. CTS-1 must therefore validate conformance to the underlying RATS protocol stack, not merely to the MAI-1 specification in isolation. The relevant RATS components are:

Component	Status	CTS-1 Role
EAT (Entity Attestation Token)	RFC 9711	Stable foundation. CTS-1 validates EAT claim encoding, mandatory claim presence, submodule structure, and profile identification.
CoRIM (Concise Reference Integrity Manifest)	draft-ietf-rats-corim-09	CTS-1 validates CoRIM manifests carrying certified thresholds for each MAI-1 invariant. Reference value matching against attestation evidence.
CMW (CBOR Message Wrapper)	draft-ietf-rats-msg-wrap-16	CTS-1 validates CMW Collection format for composite TEE evidence aggregation (MAI-C2).
COSE (CBOR Object Signing and Encryption)	RFC 9052 / 9053	CTS-1 validates COSE_Sign1 envelope construction, algorithm compliance, key binding, and signature verification.
SCITT	draft-ietf-scitt-architecture	CTS-1 validates SCITT receipt presence and verifiability in Layer 3 provenance evidence.

3.4.2 Interoperability Findings from IETF Hackathons

IETF hackathon testing has surfaced concrete interoperability issues that CTS-1 must specifically target. These findings represent real-world friction points discovered through multi-vendor testing:

1. **CoRIM/SCITT Header Migration.** IETF 124 identified a shift in CoRIM to use CWT claims in protected headers to support SCITT, replacing the previous "meta" COSE header approach. CTS-1 **MUST** support dual parsing: recognizing both the legacy and current formats. CTS-1 **SHALL** issue a WARNING for legacy headers and a PASS for SCITT-compatible CWT headers, driving ecosystem migration without breaking existing implementations.

2. **PKIX / RATS Evidence Mismatch.** X.509 (PKIX) certificates from Hardware Security Modules do not map cleanly to the RATS EAT claim model. CTS-1 **MUST** include an adapter layer that normalizes PKIX evidence for validation.

3. **Composite Attester Class Definitions.** Defining "classes" of composite devices (e.g., a server with GPU, TPU, and NIC, each with independent attestation capabilities) is inconsistent across implementations. CTS-1 **MUST** support recursive parsing of the `submods` claim in EAT, verifying the cryptographic binding between the main component and sub-components. Loose token aggregation without binding **SHALL** result in FAIL.

4. **Version Compatibility.** The `cocli` tool's need for a compatibility mode to parse pre-corim-07 manifests demonstrates that CTS-1 **MUST** pin to specific protocol draft versions and explicitly test version-handling behavior.

3.4.3 The Veraison Reference Ecosystem

The Veraison project provides the most comprehensive open-source RATS implementation ecosystem. CTS-1 leverages Veraison as a reference oracle (for parsing and verification cross-validation), as a test vector generation tool (for CoRIM manifests carrying MAI-1 certified thresholds), and as integration test templates (for end-to-end attestation flows).

Honest Framing

No formal RATS conformance test suite currently exists. Testing within the RATS working group is almost entirely interoperability-focused, conducted through IETF hackathons rather than systematic conformance evaluation. CTS-1 would be among the first formal conformance efforts for RATS protocol stacks. This is both an opportunity—first-mover advantage in defining the testing methodology for AI attestation—and a risk: there is no established precedent for RATS-specific conformance edge cases.

3.5 Synthesis: The CTS-1 Conformance Model

The four pillars combine into a single conformance model:

Table 3: CTS-1 Conformance Model: Four-Pillar Synthesis

Dimension	**Source Tradition**	**CTS-1 Implementation**	**Anti-Pattern Avoided**
Assurance Hierarchy	Common Criteria	MAI-C0 / C1 / C2 tiered testing with cumulative requirements	Static evaluated configuration; $750K+ cost
Requirement Decomposition	FIPS 140-3 DTR	AS / VE / TE triples for every MAI-1 normative statement	Manual evaluation; 100+ day queue backlogs
Operational Model	OpenID Foundation	Self-certification at C0; graduated assessment at C1/C2; transparency log publication	Inability to verify claim truthfulness at lower tiers
Protocol Validation	IETF RATS	CDDL structural gate; EAT/CoRIM/CMW validation; Veraison as reference oracle	Hackathon-only testing; unstable draft dependencies
Test Design	All four + BT SIG	ICS-driven test scoping; DTR-derived procedures; reactive test harness; five-verdict model	Ad hoc test selection; subjective pass/fail
Delivery	OIDF + FIPS automation	Containerized open-source harness; machine-readable reports; CI/CD integration	PDF certificates; human-only review
Enforcement	Auburn composition principle	Binary pass/fail; transparency log; procurement cascade; freshness-bounded validity	Partial compliance; interpretive wiggle room

4 Test Pyramid Architecture

CTS-1 organizes all conformance tests into a four-level pyramid optimized for speed, feedback efficiency, and diagnostic precision. The pyramid enforces a strict execution order: lower levels are fast and deterministic; upper levels are complex and stateful. Failures at lower levels block execution of higher levels, preventing wasted computation on artifacts that are already known to be non-conformant.

The pyramid is not merely an organizational convenience. It is a *gate architecture*: each level acts as a prerequisite for the next. An attestation artifact that fails structural validation is not

subjected to cryptographic verification. An artifact that fails cryptographic verification is not subjected to semantic analysis. This design ensures that diagnostic output is precise—a failure at Level 2 means the artifact is structurally valid but cryptographically compromised, not that the tester cannot determine which layer failed.

4.1 Pyramid Overview

Level	Name	Focus	Est. Test Count	Gate Behavior
L1	Structural Conformance	Binary structure and schema	200+	First gate. Failures here block all subsequent levels.
L4	Cryptographic Integrity	Signatures, chains, binding	50+	Parallel with L1 for independent artifacts; sequential after L1 for endpoint testing. Failures block L2 and L3.
L2	Semantic Correctness	Field values, cross-field logic, temporal consistency	100+	Executes only after L1 and L4 pass. Failures block L3.
L3	Behavioral Validity	State transitions, adversarial resilience, error handling	20–50	Apex of the pyramid. Executes only after L1, L4, and L2 pass. Required only at MAI-C1 and MAI-C2.

Note on numbering: The levels are numbered L1, L2, L3, L4 to align with the semantic content of each layer, not with execution order. L4 (Cryptographic) executes early in the pipeline because cryptographic integrity is a cross-cutting concern that applies at every level. The execution order is L1 → L4 → L2 → L3.

4.2 Level 1: Structural Conformance (The CDDL Gate)

4.2.1 Purpose

Level 1 verifies that the attestation artifact is *well-formed*: a valid CBOR-encoded payload wrapped in a structurally correct COSE_Sign1 envelope, conforming to the MAI-1 CDDL profile. This is the "unit test" layer of the pyramid. If an artifact fails structural validation, all subsequent testing is moot—there is no point verifying the signature of a payload that cannot be parsed, or checking the semantic consistency of fields that do not exist.

4.2.2 Validation Method

Level 1 ingests the raw attestation artifact (byte stream) and validates it against the MAI-1 CDDL schema modules using a deterministic CDDL validator. The MAI-1 CDDL schema is organized as four modules:

Schema Module	Scope
`mai1-cose.cddl`	COSE_Sign1 envelope constraints. Validates the outermost structure: protected header, unprotected header, payload, and signature fields. Enforces CBOR Tag 18.
`mai1-headers.cddl`	Protected and unprotected header parameters. Validates algorithm identifier (`alg`), key identifier (`kid`), certificate chain (`x5chain`), content type, and MAI-1-specific header extensions.
`mai1-claims.cddl`	MAI-1-specific claims within the EAT payload. Validates Layer 1 platform attestation fields, Layer 2 model state invariant fields, Layer 3 provenance binding fields, and execution context fields as defined in MAI-1 §6.2.
`eat-base.cddl`	Base EAT claims imported from RFC 9711. Validates standard EAT claims: `iss`, `sub`, `iat`, `exp`, `eat_nonce`, `eat_profile`, `submods`.

4.2.3 Two-Pass Validation Strategy

The COSE_Sign1 structure nests CBOR within CBOR: the protected header is a byte string containing a CBOR-encoded header map, and the payload is a byte string containing a CBOR-encoded EAT Claims-Set. CTS-1 employs a two-pass pipeline:

Requirement 4.1 (Two-Pass Structural Validation)

Pass 1 (Envelope Validation):

1. Decode the outermost CBOR structure.
2. Verify CBOR Tag 18 (COSE_Sign1) is present.
3. Verify the four-element array structure: `[protected, unprotected, payload, signature]`.
4. Validate the protected header byte string is decodable as CBOR.
5. Validate the decoded protected header against `mai1-headers.cddl`.
6. Validate the payload byte string is decodable as CBOR.

Pass 2 (Payload Validation):

1. Extract the decoded payload CBOR.
2. Validate against `eat-base.cddl` for standard EAT claims.
3. Validate against `mai1-claims.cddl` for MAI-1-specific claims.
4. For each `submods` entry (Layer 1, Layer 2, Layer 3 submodules), recursively validate against the applicable CDDL fragment.

Gate Rule: If Pass 1 fails, Pass 2 is not attempted. If any step in either pass fails, the overall Level 1 verdict is FAIL, and no higher-level tests are executed.

4.2.4 Specific Structural Checks

Level 1 validates the following categories of structural properties:

Check Category	Validation	Failure Example
Mandatory claim presence	Every field not marked with CDDL `?` operator is present.	`entropy-floor` missing from Layer 2 invariants block.
Data type correctness	Each field matches its declared CDDL type (`float`, `bstr`, `tstr`, `uint`, `uuid`).	`drift-kl` encoded as `tstr` instead of `float`.
Canonical CBOR encoding	Map keys are sorted per RFC 8949 §4.2.1 (deterministic encoding). No duplicate map keys. Preferred serialization for integers and floats.	Map keys in non-sorted order; duplicate key in invariants block.
COSE envelope integrity	CBOR Tag 18 present. Four-element array. Protected header is non-empty `bstr`. Signature is `bstr` of correct length for declared algorithm.	Tag 18 missing; protected header is empty; signature truncated.
Submodule structure	`submods` claim present (MAI-C1+). Each submodule is a valid nested EAT token or detached EAT bundle reference.	`submods` contains raw CBOR map instead of nested EAT.
Enumeration validity	Enumerated fields (`compliance-flag`, `severity-enum`, `bom-format-enum`) contain values from the defined set.	`compliance-flag` set to `3` (undefined; valid range 0–2).
Cardinality constraints	Array fields have correct element counts. The `coherence-energy-band` is exactly a two-element array `[float, float]`.	`coherence-energy-band` has three elements.
Tag usage	CBOR tags used correctly: Tag 1 for epoch timestamps, Tag 37 for UUIDs, Tag 6 for URIs where applicable.	`inference-id` encoded as plain `bstr` instead of Tag 37 UUID.

4.2.5 Dual-Stack Format Support

MAI-1 Requirement 6.1 specifies CBOR as the canonical encoding, with JSON permitted for transport and display but not for verification or archival. CTS-1 implements format handling as follows:

Requirement 4.2 (Format Handling)

1. CTS-1 **SHALL** perform all conformance testing against the CBOR-encoded artifact. This is the canonical form.
2. If the implementation also produces JSON-encoded artifacts (via JWT), CTS-1 **SHALL** validate JSON structural conformance using JSON Schema Draft 2020-12.
3. If an implementation claims dual-format support, CTS-1 **SHALL** verify that the CBOR and JSON representations are *semantically identical*: the same claims, the same values, the same structure. Any semantic divergence between formats is a FAIL.
4. CTS-1 **MUST NOT** accept a JSON-only artifact as the basis for conformance determination. JSON is a transport convenience, not a conformance encoding.

4.3 Level 4: Cryptographic and Integrity Validation

4.3.1 Purpose

Level 4 verifies that the attestation artifact is *authentic and untampered*: the COSE_Sign1 signature is valid, the certificate chain is trustworthy, the signing algorithm is approved, and the cryptographic binding between layers is intact. Level 4 is designated as a cross-cutting concern because cryptographic integrity applies to every component of the attestation artifact.

Despite its numbering (L4), this level executes immediately after Level 1 in the CTS-1 pipeline. The rationale: an artifact that is structurally valid but cryptographically compromised must be identified before any semantic analysis is performed. Semantic checks assume the artifact has not been tampered with; if the signature is invalid, field values cannot be trusted, and semantic testing would produce misleading results.

4.3.2 Specific Cryptographic Checks

Check Category	Validation	Pass/Fail Criteria
Signature verification	Extract the COSE_Sign1 signature. Reconstruct the `Sig_structure` per RFC 9052 §4.4. Verify the signature using the public key from the certificate chain.	PASS: Signature verifies. FAIL: Signature does not verify (Critical Failure).
Algorithm compliance	Extract the `alg` parameter from the protected header. Verify it is on the MAI-1 Approved Algorithm List.	PASS: Algorithm is approved (ES256, ES384, ES512, or EdDSA). FAIL: Algorithm is absent, unrecognized, or on the prohibited list (`alg: none`, weak algorithms).

Check Category	Validation	Pass/Fail Criteria
Certificate chain validation	Extract the `x5chain` parameter. Walk the chain from leaf to root. Verify each certificate's signature. Verify the root is a recognized trust anchor.	PASS: Chain validates to a recognized root CA. FAIL: Chain is broken, expired, revoked, or terminates at an unrecognized root.
TEE key binding (C1+)	Verify the signing key is certified by the hardware manufacturer's CA as TEE-resident. The key **MUST** be bound to the platform attestation evidence in Layer 1.	PASS: Key certificate contains TEE residency attestation from recognized manufacturer. FAIL: Key certificate lacks TEE binding or is issued by unknown CA.
Nonce binding	Verify `eat_nonce` in the payload matches the challenge nonce sent by CTS-1 in the attestation request.	PASS: Nonce matches exactly. FAIL: Nonce mismatch, absent, or truncated.
Timestamp freshness	Verify `iat` (issued-at) is within the acceptable clock skew window. Verify `exp` (expiration) has not passed. Verify temporal ordering: `iat` $\leq$ `generated-at` $\leq$ `exp`.	PASS: All temporal constraints satisfied. FAIL: Any temporal violation.
Merkle root verification	Reconstruct the Merkle tree from the Layer 1, Layer 2, and Layer 3 submodule hashes. Verify the computed root matches the signed root hash.	PASS: Computed root matches signed root. FAIL: Root mismatch (indicates tampering or incorrect tree construction).
Decision Receipt binding (C1+)	Verify the Decision Receipt's `attestation-root-hash` matches the Merkle root of the referenced attestation artifact. Verify `inference-id` matches.	PASS: Receipt binds correctly to attestation. FAIL: Binding mismatch.

4.3.3 Negative Cryptographic Testing

Level 4 includes mandatory negative tests that verify the implementation correctly *rejects* cryptographically compromised artifacts. These tests are critical because accepting an invalid attestation artifact is a security vulnerability.

Requirement 4.3 (Mandatory Negative Cryptographic Tests)

The following negative tests **MUST** be executed at all conformance levels. For each test, CTS-1 generates a tampered artifact from a known-good baseline.

1. **Bit-Flip Test.** CTS-1 flips a single bit in the payload of a valid artifact (leaving the signature unchanged). The implementation **MUST** reject the artifact due to signature verification failure.
2. **Algorithm Downgrade Test.** CTS-1 submits an artifact with `alg: -257` (RS256, if not on the approved list) or `alg: 0` (none). The implementation **MUST** reject the artifact.
3. **Expired Certificate Test.** CTS-1 submits an artifact signed with a key whose certificate chain contains an expired intermediate or root certificate. The implementation **MUST** reject the artifact.
4. **Wrong-Key Signature Test.** CTS-1 signs a valid payload with a key that is *not* in the declared certificate chain. The implementation **MUST** reject the artifact.
5. **Replay Attack Test.** CTS-1 captures a valid artifact from a previous test run (with a stale nonce) and resubmits it. The implementation **MUST** reject the replayed artifact due to nonce mismatch.
6. **Merkle Proof Tampering Test.** CTS-1 modifies a single submodule hash in the Merkle tree while preserving the signed root. The implementation **MUST** detect the inconsistency during proof reconstruction.

4.4 Level 2: Semantic and Contextual Validation

4.4.1 Purpose

Level 2 verifies that the attestation artifact's field values are *meaningful and consistent*: measured invariant values fall within physically plausible ranges, compliance flags are correctly derived from measured values and certified thresholds, temporal fields are consistent, and cross-field constraints are satisfied. Level 2 executes only after Levels 1 and 4 have passed, ensuring that the artifact being analyzed is structurally valid and cryptographically authentic.

4.4.2 Specific Semantic Checks

Check Category	Validation	Pass/Fail Criteria
Compliance flag consistency	Independently compute the compliance determination from the reported measured values and certified thresholds. Compare with the reported `compliance-status` field.	PASS: Independent computation matches reported flag. FAIL: Mismatch (the "self-grading" detection test).
Invariant range plausibility	Verify measured invariant values fall within physically plausible ranges. Entropy $H(t) \in [0, H_{\max}]$. KL divergence $D_{KL} \geq 0$. Dirichlet energy $E_D \geq 0$. Temperature $T_{SRAM} > 0$.	PASS: All values within plausible ranges. FAIL: Any value outside physical bounds (e.g., negative entropy).
Threshold ordering	Verify certified thresholds are internally consistent. `coherence-energy-band[0]` $<$ `coherence-energy-band[1]` (min $<$ max). `entropy-floor` > 0. `gradient-stability-max` > 0.	PASS: Thresholds internally consistent. FAIL: Threshold inversion or zero/negative thresholds.
Compliance derivation logic	Verify GREEN/YELLOW/RED derivation: GREEN: all invariants within certified bounds. YELLOW: warning threshold crossed. RED: hard invariant breach.	PASS: Derivation matches MAI-1 §6.2.2 logic. FAIL: Flag does not match measured values.
Breach field consistency	If `compliance-status` is YELLOW or RED, then `breached-invariant` and `breach-severity` **MUST** be present and identify the specific invariant that triggered the breach.	PASS: Breach fields present and consistent with measured values. FAIL: Breach fields absent when status is non-GREEN.

Check Category	Validation	Pass/Fail Criteria
Temporal consistency	`iat` $\leq$ `generated-at`. `generated-at` $\leq$ `exp`. `generated-at` is not in the future (within clock skew tolerance). Boot count is monotonically non-decreasing across sequential attestations.	PASS: All temporal constraints satisfied. FAIL: Any temporal ordering violation.
Profile and level consistency	`mai-profile` matches the declared MAI-1 version. `conformance-level` matches the level declared in the ICS.	PASS: Profile and level consistent. FAIL: Mismatch between artifact claims and ICS declaration.
Model identity stability (C1+)	Across multiple attestations within a session (no retraining event), `model-hash` and `model-architecture` **MUST** remain constant.	PASS: Model identity stable across session. FAIL: Model identity changes without corresponding provenance update.

4.5 Level 3: Behavioral and State Machine Validation

4.5.1 Purpose

Level 3 is the apex of the test pyramid. It verifies that the MAI-1 implementation *behaves correctly over time and under stress*: state transitions produce expected changes in attestation artifacts, adversarial inputs are handled safely, and the system's runtime behavior is consistent with its attestation claims. Level 3 tests are stateful—they require multiple sequential interactions with the attestation endpoint and may require environmental manipulation (simulated invariant breaches, simulated hardware events).

Level 3 tests are required only at MAI-C1 and MAI-C2. MAI-C0 conformance is determined entirely by Levels 1, 4, and 2.

4.5.2 Specific Behavioral Checks

Check Category	Test Procedure	Pass/Fail Criteria
State transition: reboot	Request attestation $\rightarrow$ trigger platform reboot $\rightarrow$ request attestation again. Verify `boot-seed` has changed and `boot-count` has incremented.	PASS: Boot seed changed, boot count incremented. FAIL: Boot seed unchanged after reboot (indicates static measurement caching).

Check Category	Test Procedure	Pass/Fail Criteria
Invariant breach response (C2)	Inject a simulated invariant breach (e.g., entropy below $H_{\min}$). Request attestation. Verify `compliance-status` transitions from GREEN to YELLOW or RED. Verify `breached-invariant` identifies the correct invariant.	PASS: Breach detected, flag transitions correctly, correct invariant identified. FAIL: Breach not detected, flag remains GREEN.
Breach recovery (C2)	After breach injection, restore the invariant to a compliant value. Request attestation. Verify `compliance-status` returns to GREEN. Verify the breach is logged in the attestation history.	PASS: Recovery detected, flag returns to GREEN. FAIL: Flag stuck in breach state after recovery.
Hardware quote mismatch (C2)	Present an invalid or expired hardware attestation quote. The system **MUST** reject the quote and either refuse to generate an attestation artifact or generate an artifact with `compliance-status` RED.	PASS: Invalid quote rejected or flagged. FAIL: Invalid quote silently accepted.
Error handling: malformed request	Send attestation requests with malformed bodies: oversized payloads, empty bodies, wrong content types, invalid CBOR, truncated requests.	PASS: Correct error codes returned. No 500 errors, no crashes, no stack traces in response. FAIL: Server error, crash, or information leakage.
Attestation rate consistency (C2)	Request N attestations over a defined time window. Verify the system sustains the required measurement frequency. Verify no attestation artifacts are identical (indicating caching rather than fresh measurement).	PASS: All attestations unique, frequency requirement met. FAIL: Duplicate attestations or frequency below minimum.

Check Category	Test Procedure	Pass/Fail Criteria
Clock skew tolerance (C2)	Submit attestation artifacts with timestamps at the boundary of the configurable skew tolerance. Artifacts exactly at the tolerance boundary **MUST** be accepted. Artifacts exceeding the tolerance **MUST** be rejected.	PASS: Boundary behavior correct. FAIL: Off-by-one error in skew check.
Concurrent request handling	Submit multiple simultaneous attestation requests. Verify each response contains a unique `inference-id` and correctly echoes its respective nonce. No cross-contamination between concurrent requests.	PASS: All concurrent responses unique and correctly bound. FAIL: Nonce cross-contamination or shared inference IDs.

4.6 Gate Logic and Execution Flow

The four levels interact through a strict gate architecture:

Requirement 4.4 (Pyramid Gate Logic)

1. **L1 Gate.** If any L1 (Structural) test produces a FAIL verdict, the CTS-1 run terminates with an overall FAIL determination. L4, L2, and L3 tests are not executed. The diagnostic report identifies the structural failure(s).
2. **L4 Gate.** If any L4 (Cryptographic) test produces a FAIL verdict, L2 and L3 tests are not executed. The diagnostic report identifies the cryptographic failure(s). L1 results are preserved in the report.
3. **L2 Gate.** If any L2 (Semantic) test produces a FAIL verdict, L3 tests are not executed. L1 and L4 results are preserved in the report.
4. **L3 Execution.** L3 (Behavioral) tests execute only if L1, L4, and L2 have all passed (no FAIL verdicts). L3 tests are applicable only at MAI-C1 and MAI-C2.
5. **Parallel Execution Exception.** When testing pre-generated artifacts (offline validation, not live endpoint testing), L1 and L4 **MAY** execute in parallel because both operate on the same static byte stream. When testing a live endpoint, L4 executes sequentially after L1 because the endpoint must first produce a parseable response before its cryptographic properties can be verified.

4.7 Test Volume and Coverage Targets

CTS-1 targets the following test volumes at full specification coverage:

Pyramid Level	Positive Tests	Negative Tests	Total	Notes
L1 Structural	80+	120+	200+	Negative tests outnumber positive: more ways to be malformed than well-formed.
L4 Cryptographic	20+	30+	50+	Negative tests include all six mandatory tests from Requirement 4.3 plus algorithm-specific and chain-specific variants.
L2 Semantic	50+	50+	100+	Balanced positive/negative: every semantic rule has both valid and invalid test cases.
L3 Behavioral	10–25	10–25	20–50	Lower count, higher complexity. Each behavioral test is a multi-step scenario.
Total	**160+**	**210+**	**370+**	Negative tests constitute >50% of the total corpus.

Requirement 4.5 (Coverage Target)

CTS-1 **SHALL** achieve 100% coverage of all **MUST** and **SHALL** statements in MAI-1 §6–§9. Every normative requirement **SHALL** trace to at least one CTS-1 assertion. The complete traceability matrix is provided in Appendix B of the private specification.

5 Derived Test Requirements Registry

This section contains the complete assertion registry for CTS-1. Every normative **MUST** and **SHALL** statement in MAI-1 §6–§9 is decomposed into a numbered assertion with unambiguous pass/fail criteria. This registry is the operational core of CTS-1—it transforms MAI-1 from a normative specification into an executable compliance surface.

5.1 Registry Format

Each assertion entry provides:

Field	Content
Assertion ID	Four-part identifier: `AS-{Pyramid}-{Layer}-{Seq}`
MAI-1 Source	Section and requirement number in MAI-1
Normative Text	The **MUST**/**SHALL** statement being tested
Tier Applicability	Conformance levels: [C0,C1,C2], [C1,C2], or [C2]
Pyramid Level	L1-STRUCTURAL, L2-SEMANTIC, L3-BEHAVIORAL, or L4-CRYPTO
Pass/Fail Summary	One-line deterministic criterion

Private Version Content

The private specification contains the complete Derived Test Requirements for each assertion, including: Vendor Evidence (VE) documentation requirements, step-by-step Test Evidence (TE) procedures with explicit inputs, outputs, and verdict logic, and negative test variants. The condensed entries below specify *what* is tested and *what constitutes failure*. The private version specifies *how* to execute each test.

Contact: UncleBroFields@proton.me fieldsryanchristopher@gmail.com

5.2 General Encoding and Endpoint Requirements

These assertions apply to the MAI-1 attestation interface itself, independent of any specific layer's payload content. They validate MAI-1 §6.1 (Canonical Endpoint) and §6.3 (Encoding and Signing Requirements).

AS-ST-EP-01: Canonical Endpoint Availability

MAI-1 Source	§6.1 (Canonical Endpoint)
Tier	[C0, C1, C2] **Level:** L1-STRUCTURAL
Normative Text	Every MAI-1 compliant system **MUST** expose a single attestation endpoint accepting evidence requests and returning signed attestation artifacts.
Pass	Endpoint responds to `POST /v1/mai/attest` with HTTP 200 and `application/eat+cwt` content type.
Fail	Endpoint unreachable, returns non-200 status, accepts non-POST methods, or returns incorrect content type.

AS-ST-EP-02: Request Body Schema

MAI-1 Source	§6.1 (Request Body)
Tier	[C0, C1, C2] **Level:** L1-STRUCTURAL
Normative Text	The request body **MUST** contain a verifier-supplied freshness nonce (`nonce: bstr`), a requested layers bitmask (`requested_layers: uint`), and an optional selective disclosure array.
Pass	Endpoint returns evidence from all requested layers; rejects requests with missing or empty nonce.
Fail	Missing layers in response, or endpoint accepts nonce-less requests.

AS-ST-EP-03: COSE_Sign1 Envelope Structure

MAI-1 Source	§6.3, Requirement 6.1 (Encoding)
Tier	[C0, C1, C2] **Level:** L1-STRUCTURAL
Normative Text	MAI-1 attestation artifacts **MUST** be encoded in CBOR (RFC 8949) and signed using COSE_Sign1 (RFC 9052).
Pass	CBOR Tag 18 present; four-element array confirmed; protected header decodable with `alg` present; payload decodable as CBOR map; signature non-empty and correct length for declared algorithm.
Fail	Tag missing, wrong array length, protected header malformed, payload not decodable, or signature truncated.

AS-ST-EP-04: Canonical CBOR Encoding

MAI-1 Source	§6.3, Requirement 6.1, RFC 8949 §4.2.1
Tier	[C0, C1, C2] **Level:** L1-STRUCTURAL
Normative Text	MAI-1 attestation artifacts **MUST** use deterministic CBOR encoding.
Pass	All map keys sorted in bytewise lexicographic order; no duplicate keys; integer values use preferred serialization.
Fail	Unsorted keys, duplicate keys detected, or non-deterministic encoding.

AS-ST-EP-05: Field Completeness

MAI-1 Source	§6.3, Requirement 6.2 (Field Completeness)
Tier	[C0, C1, C2] **Level:** L1-STRUCTURAL
Normative Text	All mandatory fields defined in §6.2.1–§6.2.4 **MUST** be present. There is no partial compliance.
Pass	Every mandatory field present, non-null, and correctly typed per CDDL definition.
Fail	Any mandatory field missing, null, or incorrectly typed.

5.3 Layer 1: Platform Attestation

Layer 1 assertions validate the hardware root-of-trust evidence defined in MAI-1 §6.2.1 (`mai-platform-attestat`
These fields answer the question: *Is the hardware platform trustworthy?*

AS-ST-L1-01: TEE Type Declaration

MAI-1 Source	§6.2.1, `tee-type` field
Tier	[C0, C1, C2] **Level:** L1-STRUCTURAL
Normative Text	Layer 1 attestation **MUST** include a `tee-type` field identifying the Trusted Execution Environment technology.
Pass	Field present; recognized TEE type from MAI-1 registry; consistent with certificate chain.
Fail	Field absent, unrecognized type, or TEE type contradicts certificate chain origin.

AS-ST-L1-02: Firmware Measurement Chain

MAI-1 Source	§6.2.1, `firmware-measurements` field
Tier	[C0, C1, C2] **Level:** L1-STRUCTURAL
Normative Text	Layer 1 attestation **MUST** include firmware measurement hashes for all components in the measured boot chain.
Pass	Non-empty array; each entry contains component ID, approved hash algorithm (SHA-256 minimum), and hash value; measurements match CoRIM reference values (C1+).
Fail	Empty array, malformed entries, deprecated hash algorithms, or reference value mismatch.

AS-ST-L1-03: Boot Seed and Boot Count

MAI-1 Source	§6.2.1, `boot-seed` and `boot-count` fields
Tier	[C0, C1, C2] **Level:** L1-STRUCTURAL
Normative Text	Layer 1 attestation **MUST** include a `boot-seed` (random value generated at boot) and a `boot-count` (monotonically increasing boot counter).
Pass	`boot-seed` present and $\geq$32 bytes; `boot-count` present and non-negative; both stable within session; seed changes and count increments on reboot (C1+).
Fail	Either field absent or wrong type; fields unstable within session; seed unchanged after reboot (static caching); count not bound to boot events.

AS-CR-L1-04: Hardware Attestation Quote

MAI-1 Source	§6.2.1, hardware quote field; MAI-1 Table 6
Tier	[C1, C2] **Level:** L4-CRYPTO
Normative Text	At MAI-C1 and above, Layer 1 attestation **MUST** include a hardware attestation quote independently verifiable against the manufacturer's attestation service.
Pass	Quote present, structurally valid for declared TEE type, verifies against manufacturer CA, measurements consistent with MAI-1 Layer 1 fields.
Fail	Quote absent, structurally malformed, signature invalid, expired TCB, or measurement mismatch between quote and Layer 1 payload.

AS-CR-L1-05: Signing Key TEE Binding

MAI-1 Source	§6.3, Requirement 6.3 (Signing Key Binding)
Tier	[C1, C2] **Level:** L4-CRYPTO
Normative Text	The COSE_Sign1 signature **MUST** be generated by a key that is (a) generated within the TEE boundary, (b) certified by the hardware manufacturer's CA, and (c) bound to the platform attestation evidence in Layer 1.
Pass	Certificate chain validates to recognized manufacturer root; leaf certificate contains TEE residency attestation extension; signing key matches certificate; platform identity consistent.
Fail	Chain broken or unrecognized root; TEE residency extension absent; key mismatch; platform identity inconsistent.

AS-SM-L1-06: Platform Identity Consistency

MAI-1 Source	§6.2.1, platform identity fields
Tier	[C0, C1, C2] **Level:** L2-SEMANTIC
Normative Text	Layer 1 platform identity fields **MUST** be internally consistent and stable across sequential attestations within the same hardware platform.
Pass	All identity fields (`tee-type`, `hardware-vendor`, `hardware-model`, `platform-id`) identical across sequential attestations; non-empty strings.
Fail	Any identity field changes between sequential attestations, or any field absent/empty.

AS-SM-L1-07: Firmware Version Reporting

MAI-1 Source	§6.2.1, firmware version fields
Tier	[C0, C1, C2] **Level:** L2-SEMANTIC
Normative Text	Layer 1 attestation **MUST** report firmware version information sufficient for reference value lookup and vulnerability assessment.
Pass	Firmware version fields present, non-empty, consistent with vendor-declared format, and stable within session.
Fail	Version fields absent, empty, or inconsistent across sequential attestations without intervening firmware update.

5.4 Layer 2: Model State Invariants

Layer 2 assertions validate the continuous model health evidence defined in MAI-1 §6.2.2 and §7. These fields answer the question: *Is the model healthy right now?* Layer 2 contains the five mandatory invariants (entropy floor, gradient stability, distribution drift, structural coherence, SRAM thermal integrity) plus model identity, compliance flag logic, and measurement frequency requirements.

AS-ST-L2-01: Model Identity Fields

MAI-1 Source	§6.2.2, model identity fields
Tier	[C0, C1, C2] **Level:** L1-STRUCTURAL
Normative Text	Layer 2 attestation **MUST** include model identity fields: `model-id` (UUID), `model-family` (tstr), `model-version` (tstr), and `parameter-count` (uint).
Pass	All four fields present, correctly typed, non-empty; `model-id` is valid UUID (Tag 37); `parameter-count` > 0.
Fail	Any identity field missing, wrong type, null, or `parameter-count` of zero.

AS-ST-L2-02: Invariant Measurement Array Structure

MAI-1 Source	§6.2.2, invariants array
Tier	[C0, C1, C2] **Level:** L1-STRUCTURAL
Normative Text	Layer 2 **MUST** contain an `invariants` array with one entry per measured invariant, each containing: `invariant-id`, `measured-value`, `certified-threshold`, `measurement-timestamp`, and `measurement-method`.
Pass	Array present and non-empty; each entry contains all five mandatory sub-fields, correctly typed.
Fail	Array absent, empty, or any entry missing mandatory sub-fields.

AS-ST-L2-03: Mandatory Invariant Presence

MAI-1 Source	§7.1–§7.5 (Five Mandatory Invariants)
Tier	[C1, C2] **Level:** L1-STRUCTURAL
Normative Text	At MAI-C1 and above, the invariants array **MUST** contain entries for all five mandatory invariants: entropy floor ($H_{\min}$), gradient stability ($\|\nabla\mathcal{L}\|_{\max}$), distribution drift ($D_{\mathrm{KL},\max}$), structural coherence (E_D band), and SRAM thermal integrity ($T_{\mathrm{SRAM},\max}$).
Pass	All five invariant IDs present in the array with valid measurements.
Fail	Any of the five mandatory invariants absent at C1 or C2.

AS-SM-L2-04: Invariant Range Plausibility

MAI-1 Source	§7.1–§7.5, physical bounds
Tier	[C0, C1, C2] **Level:** L2-SEMANTIC
Normative Text	Measured invariant values **MUST** fall within physically plausible ranges.
Pass	Entropy $H(t) \in [0, H_{\max}]$; KL divergence $D_{\mathrm{KL}} \geq 0$; Dirichlet energy $E_D \geq 0$; temperature $T_{\mathrm{SRAM}} > 0$; gradient norm ≥ 0.
Fail	Any value outside physical bounds (negative entropy, negative divergence, negative temperature).

AS-SM-L2-05: Threshold Ordering and Consistency

MAI-1 Source	§7, certified threshold semantics
Tier	[C0, C1, C2] **Level:** L2-SEMANTIC
Normative Text	Certified thresholds **MUST** be internally consistent: band minima below maxima, floor values positive, ceiling values positive.
Pass	`coherence-energy-band[0]` $<$ `coherence-energy-band[1]`; `entropy-floor` > 0; `gradient-stability-max` > 0; all threshold pairs correctly ordered.
Fail	Threshold inversion, zero thresholds, or negative thresholds.

AS-SM-L2-06: Compliance Flag Logic

MAI-1 Source	§6.2.2, `compliance-status` field
Tier	[C0, C1, C2] **Level:** L2-SEMANTIC
Normative Text	The `compliance-status` field **MUST** be independently derivable from the reported measured values and certified thresholds using the GREEN/YELLOW/RED logic defined in MAI-1.
Pass	Independently computed compliance flag matches reported flag.
Fail	Mismatch between computed and reported flag (self-grading detection).

AS-SM-L2-07: Entropy Floor Invariant Validation

MAI-1 Source	§7.1 (Entropy Floor, Clause AI-8)
Tier	[C1, C2] **Level:** L2-SEMANTIC
Normative Text	The reported entropy measurement $H(t)$ **MUST** be compared against the certified entropy floor $H_{\min}$. Breach semantics: $H(t) < H_{\min}$ triggers RED.
Pass	Entropy value correctly compared against certified floor; compliance flag reflects the comparison result.
Fail	Compliance flag does not reflect entropy breach, or breach condition ignored.

AS-SM-L2-08: Gradient Stability Invariant Validation

MAI-1 Source	§7.2 (Gradient Stability, Clause AI-2)
Tier	[C1, C2] **Level:** L2-SEMANTIC
Normative Text	The reported gradient norm $\|\nabla\mathcal{L}\|$ **MUST** be compared against the certified stability ceiling $\|\nabla\mathcal{L}\|_{\max}$. Breach semantics: exceedance triggers RED.
Pass	Gradient value correctly compared against certified ceiling; compliance flag reflects the comparison result.
Fail	Compliance flag does not reflect gradient exceedance, or breach condition ignored.

AS-SM-L2-09: Distribution Drift Invariant Validation

MAI-1 Source	§7.3 (Distribution Drift, Clause AI-6)
Tier	[C1, C2] **Level:** L2-SEMANTIC
Normative Text	The reported KL divergence D_{KL} **MUST** be compared against the certified drift ceiling $D_{\mathrm{KL,max}}$. Breach semantics: exceedance triggers YELLOW; sustained exceedance triggers RED.
Pass	Drift value correctly compared against certified ceiling; compliance flag reflects both instantaneous and sustained breach conditions.
Fail	Compliance flag does not reflect drift exceedance, or sustained breach escalation logic absent.

AS-SM-L2-10: Structural Coherence Invariant Validation

MAI-1 Source	§7.4 (Structural Coherence, Clause AI-7)
Tier	[C1, C2] **Level:** L2-SEMANTIC
Normative Text	The reported Dirichlet energy E_D **MUST** fall within the certified coherence energy band $[E_{D,\min}, E_{D,\max}]$. Breach outside either bound triggers RED.
Pass	Coherence energy correctly compared against both band boundaries; compliance flag reflects out-of-band condition.
Fail	Compliance flag does not reflect coherence breach, or only one bound checked.

AS-SM-L2-11: SRAM Thermal Integrity Invariant Validation

MAI-1 Source	§7.5 (Thermal Integrity, Clause AI-4)
Tier	[C1, C2] **Level:** L2-SEMANTIC
Normative Text	The reported SRAM junction temperature T_{SRAM} **MUST** be compared against the certified thermal ceiling $T_{\mathrm{SRAM,max}}$. Breach semantics: exceedance triggers RED.
Pass	Temperature correctly compared against certified ceiling; compliance flag reflects thermal exceedance.
Fail	Compliance flag does not reflect thermal breach, or breach condition ignored.

AS-SM-L2-12: Measurement Frequency Compliance

MAI-1 Source	§7.6 (Measurement Frequency)
Tier	[C1, C2] **Level:** L2-SEMANTIC
Normative Text	Invariant measurements **MUST** be produced at or above the minimum frequency specified per conformance level.
Pass	Measurement timestamps across sequential attestations demonstrate frequency at or above the level-specific minimum.
Fail	Measurement gaps exceed the maximum allowed interval, or timestamps indicate stale/cached measurements.

AS-BH-L2-13: Invariant Breach Response

MAI-1 Source	§7 (Breach Semantics)
Tier	[C2] **Level:** L3-BEHAVIORAL
Normative Text	When an invariant breaches its certified threshold, the system **MUST** transition compliance state within the specified response window and produce an attestation artifact reflecting the breached state.
Pass	Simulated breach (injected out-of-bounds value) triggers correct state transition; subsequent attestation reflects RED status within response window.
Fail	System continues reporting GREEN/YELLOW after confirmed breach, or state transition exceeds maximum response latency.

5.5 Layer 3: Provenance Binding

Layer 3 assertions validate the supply chain integrity evidence defined in MAI-1 §6.2.3. These fields answer the question: *Can you trace this output back to a certified origin?* Layer 3 is applicable at all conformance levels for structural fields, with provenance chain verification mandatory at C1+ and contamination detection at C2.

AS-ST-L3-01: AI-BOM Presence and Structure

MAI-1 Source	§6.2.3, `ai-bom` field
Tier	[C0, C1, C2] **Level:** L1-STRUCTURAL
Normative Text	Layer 3 attestation **MUST** include an AI Bill of Materials containing: `bom-format` (enumerated), `bom-hash` (bstr), `training-data-summary`, `framework-version`, and `base-model-reference`.
Pass	All mandatory BOM fields present, correctly typed, non-empty; `bom-format` is a recognized enumeration value.
Fail	Any BOM field absent, null, or unrecognized format type.

AS-ST-L3-02: Training Provenance Chain

MAI-1 Source	§6.2.3, `provenance-chain` field
Tier	[C0, C1, C2] **Level:** L1-STRUCTURAL
Normative Text	Layer 3 **MUST** include a provenance chain documenting the sequence of transformations from base model to deployed instance.
Pass	Provenance chain present, non-empty array, each entry contains transformation type, timestamp, and integrity hash; chain is temporally ordered.
Fail	Chain absent, empty, entries missing required fields, or temporal ordering violated.

AS-CR-L3-03: SCITT Receipt Verification

MAI-1 Source	§6.2.3, SCITT integration; §8 (Cryptographic Binding)
Tier	[C1, C2] **Level:** L4-CRYPTO
Normative Text	At C1 and above, provenance claims **MUST** be registered with a SCITT transparency service and include a verifiable inclusion proof.
Pass	SCITT receipt present; inclusion proof verifies against registered transparency service.
Fail	Receipt absent, inclusion proof invalid, or transparency service unreachable after retry.

AS-SM-L3-04: Provenance Chain Integrity

MAI-1 Source	§6.2.3, chain integrity requirements
Tier	[C1, C2] **Level:** L2-SEMANTIC
Normative Text	Each provenance chain entry's output hash **MUST** match the subsequent entry's input hash. The chain **MUST** be unbroken from base model to deployed instance.
Pass	Hash chain is complete and consistent: $\text{output}_n = \text{input}_{n+1}$ for all sequential entries.
Fail	Any hash chain break, or gap between chain endpoint and deployed model identity.

AS-SM-L3-05: Contamination Detection Declaration

MAI-1 Source	§6.2.3, contamination detection fields
Tier	[C2] **Level:** L2-SEMANTIC
Normative Text	At C2, Layer 3 **MUST** include benchmark contamination detection results: methodology, benchmarks tested, contamination scores, and pass/fail determination per benchmark.
Pass	Contamination fields present; methodology declared; at least one benchmark tested; results internally consistent.
Fail	Contamination fields absent at C2, or results incomplete/internally inconsistent.

AS-BH-L3-06: Provenance Tamper Detection

MAI-1 Source	§6.2.3, tamper resistance
Tier	[C2] **Level:** L3-BEHAVIORAL
Normative Text	The system **MUST** detect and reject tampered provenance evidence.
Pass	System correctly rejects provenance chain with modified hash entry; compliance status reflects integrity violation.
Fail	Tampered provenance chain accepted without detection.

5.6 Execution Context and Signature

These assertions validate the per-inference execution context defined in MAI-1 §6.2.4 and the signing requirements in §6.3.

AS-ST-EC-01: Inference Identity Fields

MAI-1 Source	§6.2.4, `inference-id` and `generated-at` fields
Tier	[C0, C1, C2] **Level:** L1-STRUCTURAL
Normative Text	Each attestation artifact **MUST** include a unique `inference-id` (UUID) and a `generated-at` timestamp.
Pass	`inference-id` present as valid UUID (Tag 37); `generated-at` present as epoch timestamp (Tag 1); unique across sequential requests.
Fail	Either field absent, wrong type, or `inference-id` duplicated across requests.

AS-SM-EC-02: Temporal Consistency

MAI-1 Source	§6.2.4 and §6.3, temporal fields
Tier	[C0, C1, C2] **Level:** L2-SEMANTIC
Normative Text	Temporal fields **MUST** satisfy ordering constraints: `iat` $\leq$ `generated-at` $\leq$ `exp`. Measurement timestamps **MUST** precede `generated-at`.
Pass	All temporal ordering constraints satisfied; no future-dated measurements.
Fail	Any temporal ordering violation or measurement timestamps after artifact generation.

AS-CR-EC-03: Nonce Binding

MAI-1 Source	§6.3, freshness requirements
Tier	[C0, C1, C2] **Level:** L4-CRYPTO
Normative Text	The `eat_nonce` in the attestation payload **MUST** match the challenge nonce sent by the verifier.
Pass	Nonce matches exactly.
Fail	Nonce mismatch, absent, or truncated.

AS-CR-EC-04: Signature Algorithm Compliance

MAI-1 Source	§6.3, Approved Algorithm List
Tier	[C0, C1, C2] **Level:** L4-CRYPTO
Normative Text	The COSE `alg` parameter **MUST** identify an approved algorithm (ES256, ES384, ES512, or EdDSA).
Pass	Algorithm is on the approved list.
Fail	Algorithm absent, unrecognized, on the prohibited list, or `alg: none`.

AS-CR-EC-05: Certificate Chain Validation

MAI-1 Source	§6.3, certificate requirements
Tier	[C0, C1, C2] **Level:** L4-CRYPTO
Normative Text	The `x5chain` parameter **MUST** contain a valid certificate chain from leaf to a recognized root CA.
Pass	Chain validates; no expired or revoked certificates; root is a recognized trust anchor.
Fail	Chain broken, expired intermediate, revoked certificate, or unrecognized root.

AS-BH-EC-06: Concurrent Request Isolation

MAI-1 Source	§6.1 and §6.2.4, endpoint requirements
Tier	[C1, C2] **Level:** L3-BEHAVIORAL
Normative Text	Simultaneous attestation requests **MUST** produce independent responses with unique `inference-id` values and correctly bound nonces. No cross-contamination.
Pass	All concurrent responses unique; each nonce correctly echoed; no shared inference IDs.
Fail	Nonce cross-contamination, shared inference IDs, or serialization errors under concurrent load.

5.7 Cryptographic Binding and Decision Receipts

These assertions validate the inter-layer cryptographic binding and the Decision Receipt mechanism defined in MAI-1 §8.

AS-CR-CB-01: Merkle Root Integrity

MAI-1 Source	§8.1 (Inter-Layer Binding)
Tier	[C1, C2] **Level:** L4-CRYPTO
Normative Text	The attestation artifact **MUST** include a Merkle root hash computed from the Layer 1, Layer 2, and Layer 3 submodule digests. The signed root **MUST** match independently reconstructed root.
Pass	Independently reconstructed Merkle root matches the signed root hash.
Fail	Root mismatch (indicates tampering or incorrect tree construction).

AS-CR-CB-02: Decision Receipt Binding

MAI-1 Source	§8.2 (Decision Receipts)
Tier	[C1, C2] **Level:** L4-CRYPTO
Normative Text	Each Decision Receipt **MUST** bind to the attestation artifact via `attestation-root-hash` and `inference-id` match.
Pass	Receipt's `attestation-root-hash` matches the Merkle root of the referenced attestation artifact; `inference-id` matches.
Fail	Hash mismatch or inference ID mismatch between receipt and attestation.

AS-CR-CB-03: Selective Disclosure Proof

MAI-1 Source	§8.3 (Selective Disclosure)
Tier	[C2] **Level:** L4-CRYPTO
Normative Text	When selective disclosure is requested, the system **MUST** produce a valid Merkle inclusion proof for the disclosed fields while withholding non-disclosed fields.
Pass	Inclusion proof verifies for disclosed fields; non-disclosed fields absent from response; Merkle root remains consistent.
Fail	Proof invalid, non-disclosed fields leaked, or Merkle root inconsistent with full-disclosure attestation.

AS-BH-CB-04: Tamper Detection Across Layers

MAI-1 Source	§8.1, binding integrity
Tier	[C2] **Level:** L3-BEHAVIORAL
Normative Text	Modification of any single layer's content **MUST** be detectable through Merkle root verification failure.
Pass	Single-layer modification detected via root mismatch; system rejects modified artifact or flags integrity violation.
Fail	Modified artifact accepted without detection.

5.8 Conformance Level Requirements

These assertions validate the conformance level-specific requirements defined in MAI-1 §9, ensuring that systems claiming a specific conformance level satisfy all level-appropriate requirements.

AS-SM-CL-01: Conformance Level Declaration Consistency

MAI-1 Source	§9 (Conformance Levels)
Tier	[C0, C1, C2] **Level:** L2-SEMANTIC
Normative Text	The claimed conformance level **MUST** be consistent with the evidence provided: C0 requires structural validity; C1 requires Layer 1 hardware quotes and all five invariants; C2 requires full provenance, contamination detection, and enhanced freshness.
Pass	All level-required evidence present and valid for the claimed level.
Fail	System claims a conformance level but omits evidence required at that level.

AS-SM-CL-02: Freshness Compliance per Level

MAI-1 Source	§9, freshness requirements per level
Tier	[C0, C1, C2] **Level:** L2-SEMANTIC
Normative Text	Attestation freshness **MUST** satisfy the level-specific maximum staleness window: C0 (72 hours), C1 (24 hours), C2 (1 hour).
Pass	Attestation artifact age (current time minus `generated-at`) within level-specific freshness window.
Fail	Artifact age exceeds maximum staleness for claimed conformance level.

AS-BH-CL-03: Conformance Level Escalation

MAI-1 Source	§9, level transitions
Tier	[C1, C2] **Level:** L3-BEHAVIORAL
Normative Text	A system **MUST** be capable of producing attestation evidence at the declared level under operational load. Conformance level claims **MUST** be sustainable, not achievable only under test conditions.
Pass	System produces valid attestation evidence at claimed level during sustained operational workload (defined by test profile).
Fail	System produces valid attestation under idle conditions but fails under operational load (indicates test-only compliance).

5.9 Assertion Index Summary

ID	Title	Tier	Level	Type
General Encoding & Endpoint				
AS-ST-EP-01	Canonical Endpoint Availability	C0+	L1-ST	DET
AS-ST-EP-02	Request Body Schema	C0+	L1-ST	DET
AS-ST-EP-03	COSE_Sign1 Envelope Structure	C0+	L1-ST	DET
AS-ST-EP-04	Canonical CBOR Encoding	C0+	L1-ST	DET
AS-ST-EP-05	Field Completeness	C0+	L1-ST	DET
Layer 1: Platform Attestation				
AS-ST-L1-01	TEE Type Declaration	C0+	L1-ST	DET
AS-ST-L1-02	Firmware Measurement Chain	C0+	L1-ST	DET
AS-ST-L1-03	Boot Seed and Boot Count	C0+	L1-ST	DET
AS-CR-L1-04	Hardware Attestation Quote	C1+	L4-CR	DET
AS-CR-L1-05	Signing Key TEE Binding	C1+	L4-CR	DET
AS-SM-L1-06	Platform Identity Consistency	C0+	L2-SM	DET
AS-SM-L1-07	Firmware Version Reporting	C0+	L2-SM	DET
Layer 2: Model State Invariants				
AS-ST-L2-01	Model Identity Fields	C0+	L1-ST	DET
AS-ST-L2-02	Invariant Measurement Array Structure	C0+	L1-ST	DET
AS-ST-L2-03	Mandatory Invariant Presence	C1+	L1-ST	DET
AS-SM-L2-04	Invariant Range Plausibility	C0+	L2-SM	DET
AS-SM-L2-05	Threshold Ordering and Consistency	C0+	L2-SM	DET
AS-SM-L2-06	Compliance Flag Logic	C0+	L2-SM	DET
AS-SM-L2-07	Entropy Floor Validation	C1+	L2-SM	DET
AS-SM-L2-08	Gradient Stability Validation	C1+	L2-SM	DET
AS-SM-L2-09	Distribution Drift Validation	C1+	L2-SM	DET
AS-SM-L2-10	Structural Coherence Validation	C1+	L2-SM	DET
AS-SM-L2-11	SRAM Thermal Integrity Validation	C1+	L2-SM	DET
AS-SM-L2-12	Measurement Frequency Compliance	C1+	L2-SM	DET
AS-BH-L2-13	Invariant Breach Response	C2	L3-BH	DET
Layer 3: Provenance Binding				
AS-ST-L3-01	AI-BOM Presence and Structure	C0+	L1-ST	DET
AS-ST-L3-02	Training Provenance Chain	C0+	L1-ST	DET
AS-CR-L3-03	SCITT Receipt Verification	C1+	L4-CR	DET
AS-SM-L3-04	Provenance Chain Integrity	C1+	L2-SM	DET
AS-SM-L3-05	Contamination Detection Declaration	C2	L2-SM	DET
AS-BH-L3-06	Provenance Tamper Detection	C2	L3-BH	DET
Execution Context & Signature				
AS-ST-EC-01	Inference Identity Fields	C0+	L1-ST	DET
AS-SM-EC-02	Temporal Consistency	C0+	L2-SM	DET
AS-CR-EC-03	Nonce Binding	C0+	L4-CR	DET
AS-CR-EC-04	Signature Algorithm Compliance	C0+	L4-CR	DET
AS-CR-EC-05	Certificate Chain Validation	C0+	L4-CR	DET
AS-BH-EC-06	Concurrent Request Isolation	C1+	L3-BH	DET
Cryptographic Binding & Decision Receipts				
AS-CR-CB-01	Merkle Root Integrity	C1+	L4-CR	DET
AS-CR-CB-02	Decision Receipt Binding	C1+	L4-CR	DET
AS-CR-CB-03	Selective Disclosure Proof	C2	L4-CR	DET
AS-BH-CB-04	Tamper Detection Across Layers	C2	L3-BH	DET
Conformance Level Requirements				
AS-SM-CL-01	Conformance Level Declaration Consistency	C0+	L2-SM	DET
AS-SM-CL-02	Freshness Compliance per Level	C0+	L2-SM	DET

ID	Title	Tier	Level	Type
AS-BH-CL-03	Conformance Level Escalation	C1+	L3-BH	DET

Honest Framing

The 45 assertions above represent the complete compliance surface of MAI-1. Every **MUST** and **SHALL** statement in MAI-1 §6–§9 traces to at least one assertion. The distribution across pyramid levels is:

Pyramid Level	Count	Percentage
L1-STRUCTURAL	14	31.1%
L2-SEMANTIC	16	35.6%
L3-BEHAVIORAL	6	13.3%
L4-CRYPTO	9	20.0%
Total	**45**	**100%**

The semantic layer carries the largest share because MAI-1's five mandatory invariants each generate individual validation assertions plus cross-invariant consistency checks. Behavioral assertions are the fewest but the most complex—each is a multi-step scenario requiring live system interaction. The complete traceability matrix mapping every MAI-1 normative statement to its CTS-1 assertion(s) is provided in Appendix D of the private specification.

6 Test Organization by Conformance Level

CTS-1 organizes assertions into three conformance level subsets corresponding to MAI-1 §9. Each level defines a complete, self-contained test profile: a system claiming a specific conformance level is tested against exactly the assertions applicable at that level. There is no cross-level partial credit—a system either passes all applicable assertions at its claimed level or it fails.

6.1 MAI-C0: Research and Development Subset

MAI-C0 is the entry-level conformance tier designed for research environments, pre-production systems, and organizations beginning attestation adoption. It validates that the system can produce a structurally valid, cryptographically signed attestation artifact through the canonical endpoint.

Parameter	MAI-C0 Specification
Applicable Assertions	All assertions marked [C0, C1, C2] in the registry (§5).
Assertion Count	24 of 45 (53.3%).
Pyramid Coverage	L1-STRUCTURAL (full), L2-SEMANTIC (structural and plausibility checks only), L4-CRYPTO (signature and chain validation). No L3-BEHAVIORAL tests.
Invariant Requirements	Invariant array must be structurally present; all five mandatory invariants are *not* required. Systems may report a subset of invariants at C0.
Freshness Window	72 hours maximum staleness.
Assessment Model	Automated self-certification. No Conformance Testing Entity (CTE) required. All tests resolve to PASS, FAIL, WARNING, or SKIPPED—no REVIEW verdicts permitted.
TEE Requirement	TEE presence declared but hardware quote verification not required. Software-signed attestation permitted.
Provenance	AI-BOM structural presence required. SCITT registration not required. Contamination detection not required.

Honest Framing

MAI-C0 certifies artifact conformance, not claim truthfulness. A C0-conformant system produces structurally correct, signed attestation artifacts—but the measurements within those artifacts are not independently verified. C0 is suitable for R&D environments where the attestation infrastructure is being adopted, not for regulated or safety-critical deployment. C0 is the floor, not the target.

6.2 MAI-C1: Commercial Deployment Subset

MAI-C1 is the production-grade conformance tier designed for commercial AI deployments. It adds hardware-rooted trust (TEE-bound signing), mandatory measurement of all five invariants, provenance chain verification, and third-party evaluation.

Parameter	MAI-C1 Specification
Applicable Assertions	All assertions marked [C0, C1, C2] and [C1, C2] in the registry.
Assertion Count	38 of 45 (84.4%).
Pyramid Coverage	L1-STRUCTURAL (full), L2-SEMANTIC (full including all five invariant validations), L3-BEHAVIORAL (concurrent request isolation, conformance level escalation), L4-CRYPTO (full including hardware quote, TEE binding, Merkle root, Decision Receipt, SCITT).
Invariant Requirements	All five mandatory invariants required: entropy floor, gradient stability, distribution drift, structural coherence, SRAM thermal integrity.
Freshness Window	24 hours maximum staleness.
Assessment Model	Assisted self-certification with independent Conformance Testing Entity (CTE). The CTE executes the test suite independently. REVIEW verdicts permitted (resolved by CTE analyst). Results published.
TEE Requirement	Hardware attestation quote mandatory. Signing key must be TEE-resident and certified by manufacturer CA.
Provenance	AI-BOM required. Provenance chain integrity verified. SCITT registration required. Contamination detection not required.

6.3 MAI-C2: Regulated, Insured, and Federal Subset

MAI-C2 is the highest conformance tier, designed for regulated industries, insured deployments, federal procurement, and safety-critical systems. It adds adversarial behavioral testing, breach simulation, contamination detection, selective disclosure verification, and enhanced freshness requirements.

Parameter	MAI-C2 Specification
Applicable Assertions	All 45 assertions in the registry.
Assertion Count	45 of 45 (100%).
Pyramid Coverage	All four levels, full coverage. Every assertion in the registry applies.
Invariant Requirements	All five mandatory invariants required. Invariant breach simulation mandatory (AS-BH-L2-13): the CTE injects out-of-bounds values and verifies the system detects and responds within the specified window.
Freshness Window	1 hour maximum staleness.
Assessment Model	Third-party evaluation by accredited CTE. No self-certification permitted. Adversarial testing mandatory. Results published with full assertion-level detail.
TEE Requirement	Full hardware attestation including quote verification, TEE key binding, and firmware measurement matching against CoRIM reference values.
Provenance	Full Layer 3 coverage: AI-BOM, provenance chain integrity, SCITT registration, contamination detection, and provenance tamper detection (AS-BH-L3-06).
Cryptographic Binding	Merkle root verification, Decision Receipt binding, selective disclosure proof (AS-CR-CB-03), and cross-layer tamper detection (AS-BH-CB-04) all mandatory.

6.4 Conformance Level Comparison Summary

Dimension	MAI-C0	MAI-C1	MAI-C2
Assertions	24 / 45	38 / 45	45 / 45
Assessment	Self-cert	CTE-assisted	Accredited 3rd party
Invariants	Subset permitted	All five mandatory	All five + breach sim
Freshness	72 hr	24 hr	1 hr
TEE Binding	Declared, not verified	Verified quote	Full hardware chain
Provenance	Structural only	Chain + SCITT	Chain + SCITT + contamination
Behavioral	None	Concurrency, escalation	Full adversarial suite
Crypto Binding	Signature only	Merkle + Receipt + SCITT	+ Selective disclosure + tamper
Target Use	**R&D, adoption**	**Commercial production**	**Regulated, insured, federal**

7 Negative Test Library

Negative tests verify that the system under test correctly *rejects* invalid, malformed, tampered, or adversarial attestation artifacts. In security testing, negative tests are often more important than positive tests: a system that accepts well-formed artifacts but also accepts malformed ones provides no security guarantee. CTS-1's negative test library is organized into six categories, each targeting a distinct attack surface.

Private Version Content

The specific test vectors, mutation patterns, and fuzzing seeds for each negative test category are specified in the private version. The descriptions below define the *category scope* and *attack model* for each negative test family. The private version contains the complete vector library, generation methodology, and vendor-specific profiles.

Contact: UncleBroFields@proton.me fieldsryanchristopher@gmail.com

7.1 Category 1: Malformed Artifacts

Attack Model. An attacker or misconfigured implementation produces attestation artifacts that are structurally invalid: truncated CBOR, incorrect tag usage, missing mandatory fields, wrong value types, invalid CDDL schema compliance, duplicate map keys, incorrect array lengths, and non-deterministic encoding.

Target Assertions. AS-ST-EP-01 through AS-ST-EP-05, AS-ST-L1-01 through AS-ST-L1-03, AS-ST-L2-01, AS-ST-L2-02, AS-ST-L3-01, AS-ST-L3-02, AS-ST-EC-01.

Testing Strategy. Systematic mutation of known-good artifacts: field deletion (one field per vector), type substitution (string where integer expected), truncation (partial CBOR), injection (extra unexpected fields), and encoding violations (unsorted keys, duplicate keys, non-preferred integer serialization).

Base Vector Count. 120+ vectors covering all mandatory structural fields across all three layers plus execution context.

7.2 Category 2: Cryptographic Attacks

Attack Model. An attacker produces attestation artifacts with valid structure but compromised cryptographic integrity: forged signatures, expired certificates, revoked certificate chains, algorithm downgrade (including `alg: none`), wrong-key signing, Merkle proof manipulation, Decision Receipt unbinding, and SCITT receipt forgery.

Target Assertions. AS-CR-L1-04, AS-CR-L1-05, AS-CR-EC-03 through AS-CR-EC-05, AS-CR-CB-01 through AS-CR-CB-03, AS-CR-L3-03.

Testing Strategy. The six mandatory negative cryptographic tests defined in §4 (bit-flip, algorithm downgrade, expired certificate, wrong-key signature, replay attack, Merkle proof tampering), plus extended vectors covering: certificate chain with valid leaf but expired intermediate, self-signed leaf certificate, certificate chain with revoked root, SCITT receipt with valid format but non-matching inclusion proof.

Base Vector Count. 30+ vectors covering all cryptographic validation points.

7.3 Category 3: Temporal and Freshness Attacks

Attack Model. An attacker or misconfigured system produces attestation artifacts with valid structure and cryptographic integrity but compromised temporal properties: stale attestations presented as fresh, future-dated timestamps, temporal ordering violations (`generated-at` before `iat`), measurement timestamps after artifact generation, and clock skew boundary exploitation.

Target Assertions. AS-SM-EC-02, AS-SM-CL-02, AS-SM-L2-12.

Testing Strategy. Systematic manipulation of temporal fields: artifact at freshness boundary (exactly at window edge—must pass), artifact one second past freshness window (must fail), reversed temporal ordering, measurement timestamps in the future, and `exp` timestamp in the past.

Base Vector Count. 20+ vectors covering all temporal validation points per conformance level (vectors parameterized by level-specific freshness windows).

7.4 Category 4: Semantic Consistency Attacks

Attack Model. An attacker produces attestation artifacts that are structurally valid, cryptographically signed, and temporally fresh, but contain semantically inconsistent data: compliance flag set to GREEN when measured values indicate breach, invariant values outside physical bounds, threshold inversions, and cross-field contradictions (e.g., claimed TEE type inconsistent with certificate chain origin).

Target Assertions. AS-SM-L2-04 through AS-SM-L2-11, AS-SM-L1-06, AS-SM-L1-07, AS-SM-L3-04, AS-SM-L3-05, AS-SM-CL-01.

Testing Strategy. The "self-grading" test family: for each invariant, inject a measured value that breaches the certified threshold while setting the compliance flag to GREEN. The system, acting as verifier, must independently recompute the flag and detect the mismatch. Additionally: negative entropy, negative temperature, threshold inversions, hash chain breaks, and conformance level over-claiming (C2 claimed with C0-level evidence).

Base Vector Count. 50+ vectors covering all semantic validation rules, with per-invariant breach variants.

7.5 Category 5: Behavioral and State Manipulation

Attack Model. An attacker manipulates system behavior to produce valid attestation artifacts under test conditions while behaving differently in production: breach response suppression (system detects breach but does not transition compliance state), test-only compliance (system meets freshness requirements during testing but degrades in production), concurrent request contamination, and provenance chain forgery detectable only through behavioral probing.

Target Assertions. AS-BH-L2-13, AS-BH-EC-06, AS-BH-L3-06, AS-BH-CB-04, AS-BH-CL-03.

Testing Strategy. Multi-step scenarios requiring live system interaction: inject breach condition and verify state transition within response window; submit concurrent requests and verify isolation; submit tampered provenance and verify detection; sustain operational load while monitoring attestation quality degradation.

Base Vector Count. 10–25 scenarios (lower count, higher complexity per test).

7.6 Category 6: Fuzzing Corpus

Attack Model. Automated generation of semi-random attestation artifacts designed to discover unexpected parsing failures, buffer handling errors, and edge cases not covered by the structured negative test categories above.

Target Assertions. All structural and cryptographic assertions.

Testing Strategy. Grammar-based CBOR/COSE fuzzing using the MAI-1 CDDL schema as the generation grammar. Three fuzzing modes: (1) structure-preserving mutations (valid CBOR with randomized field values), (2) encoding-level mutations (malformed CBOR byte sequences), and (3) protocol-level mutations (valid CBOR with unexpected claim combinations). Minimum fuzzing campaign: 10,000 generated artifacts per test run.

Base Vector Count. Generated per campaign; minimum 10,000 per run.

7.7 Negative Test Library Summary

Category	Base Vectors	Primary Detection Target
1. Malformed Artifacts	120+	Parser robustness, field completeness enforcement
2. Cryptographic Attacks	30+	Signature verification, chain validation, replay detection
3. Temporal/Freshness	20+	Staleness detection, clock skew handling, temporal ordering
4. Semantic Consistency	50+	Self-grading detection, plausibility enforcement
5. Behavioral/State	10–25	Breach response, concurrency isolation, tamper detection
6. Fuzzing Corpus	10,000+/run	Unknown edge cases, parser crashes, unexpected failures
Total (structured)	**250+**	
Total (with fuzzing)	**10,250+**	

8 Statistical Test Methodology

Most CTS-1 assertions are deterministic: a field is present or absent, a signature verifies or it does not, a compliance flag matches or it does not. However, a subset of assertions—particularly those involving invariant measurement stability, freshness compliance under operational load, and behavioral response timing—require statistical evaluation. A single observation may not suffice to determine whether a system consistently meets its claimed properties or whether a passing result was a fortunate sample.

8.1 Why Sequential Testing

CTS-1 employs the Truncated Sequential Probability Ratio Test (TSPRT), a method that allows conformance determination with the minimum number of observations required for a given confidence level. Unlike fixed-sample testing, TSPRT evaluates evidence continuously as observations arrive and can terminate early when the evidence is decisive—either strongly conformant or strongly non-conformant.

This design choice reflects three operational requirements:

1. **Efficiency.** Production AI systems cannot be taken offline for extended test campaigns. TSPRT minimizes the number of attestation samples required, reducing testing time from hours to minutes when the system clearly conforms or clearly fails.

2. **Fairness.** Fixed-sample tests can produce false failures when the sample size is too small for the system's natural variance. TSPRT adapts to the observed variance, requiring more samples when the result is borderline and fewer when the result is clear.

3. **Binary Finality.** TSPRT produces a definitive PASS or FAIL—never an inconclusive result. The truncation bound guarantees termination within a maximum sample count, preventing indefinite testing.

8.2 TSPRT Overview

The TSPRT operates as follows: at each observation n, compute the likelihood ratio Λ_n comparing the hypothesis that the system meets its certified threshold against the hypothesis that it does not. Compare Λ_n against decision boundaries A (accept/PASS) and B (reject/FAIL), derived from the specified Type I error rate α and Type II error rate β. If $\Lambda_n \geq A$, terminate with PASS. If $\Lambda_n \leq B$, terminate with FAIL. If $B < \Lambda_n < A$ and n has not reached the truncation bound $N_{\max}$, collect the next observation. If $n = N_{\max}$, apply the forced-decision rule based on the final likelihood ratio.

Private Version Content

The TSPRT decision boundary formulas, likelihood ratio computation for each invariant distribution family (Gaussian, bounded, and heavy-tailed), robustness diagnostics for non-Gaussian distributions, per-invariant statistical parameters (α, β, effect size, $N_{\max}$), multiple comparison correction methodology, and the complete statistical reporting schema are specified in the private version.

Contact: UncleBroFields@proton.me fieldsryanchristopher@gmail.com

8.3 Per-Invariant Statistical Parameters

The following table provides the statistical test configuration for each of the five mandatory invariants. These parameters govern how many observations are required and what constitutes a statistically significant deviation from the certified threshold.

Invariant	α	β	Effect Size	$N_{\max}$	Distribution Family
Entropy Floor	0.01	0.01	0.5σ	200	Bounded (clipped Gaussian)
Gradient Stability	0.01	0.01	0.5σ	200	Heavy-tailed (log-normal)
Distribution Drift	0.01	0.01	0.3σ	300	Non-negative (chi-squared family)
Structural Coherence	0.01	0.01	0.5σ	200	Bounded (beta-like)
Thermal Integrity	0.01	0.01	1.0σ	100	Gaussian (sensor noise dominated)

Parameter Notes:

- $\alpha = 0.01$: Probability of falsely failing a conformant system. One false failure per 100 test campaigns.
- $\beta = 0.01$: Probability of falsely passing a non-conformant system. One missed detection per 100 test campaigns.
- **Effect size**: The minimum deviation from the certified threshold that CTS-1 is designed to detect. Smaller effect sizes require more observations. Distribution drift uses a tighter effect size (0.3σ) because drift is the earliest governance-relevant signal—detecting small drift early prevents larger downstream failures.
- $N_{\max}$: The truncation bound. If the TSPRT has not terminated after $N_{\max}$ observations, the forced-decision rule applies. Thermal integrity has the lowest $N_{\max}$ because sensor-dominated measurements have low variance and converge quickly.

8.4 Honest Statistical Limitations

Honest Framing

Statistical conformance testing has inherent limitations that CTS-1 acknowledges transparently:

Limitation	CTS-1 Mitigation
Measurement truthfulness. TSPRT tests whether reported values are statistically consistent with the certified threshold—not whether the measurements are truthful. A system that consistently reports fabricated values slightly above the threshold will pass.	TEE-rooted measurement at C1+ provides hardware-backed assurance that measurements were computed by verified code. C2 adversarial testing probes for fabrication patterns. No statistical test alone can guarantee truthfulness—this is a fundamental limitation shared with all audit and certification regimes.
Non-stationarity. TSPRT assumes the underlying distribution is stationary during the test window. If the system's behavior changes during testing (e.g., load-dependent variance), the test may produce incorrect results.	CTS-1 requires test execution under declared operational conditions (AS-BH-CL-03). The test profile specifies load characteristics. If the system is load-sensitive, the test must be conducted under representative load—not idle conditions.
Multiple comparisons. Testing five invariants simultaneously at $\alpha = 0.01$ each yields an experiment-wise error rate above 0.01.	CTS-1 applies Holm–Bonferroni correction across the five invariants. The effective per-invariant α is adjusted to control the family-wise error rate at 0.01.
Adversarial adaptation. A sophisticated adversary who knows the TSPRT parameters can tune fabricated measurements to pass the test.	CTS-1 randomizes the observation sampling pattern at C2: the CTE selects which attestation samples to include in the statistical analysis from a larger stream, preventing the system from predicting which observations will be evaluated.

The governance analogy: financial auditing does not guarantee that every transaction is honest. It certifies that the auditing process was followed, the evidence is consistent, and deviations are detectable. CTS-1 statistical testing provides the same level of assurance for AI model health invariants.

9 Operational Specifications

Private Version Content

Sections 9–14 of the private specification contain the complete operational infrastructure for CTS-1 deployment:

- **Section 9: Automated Test Execution Framework.** The five-stage execution pipeline (Initialization, Evidence Collection, Test Execution, Verdict Aggregation, Report Generation), including architecture overview, stage-by-stage implementation requirements, resource requirements, and CI/CD integration specifications.
- **Section 10: Report Schema.** The machine-readable conformance report format: top-level structure, implementation identity block, CTE identity block, per-assertion result entries, aggregate statistics, TSPRT result entries, negative test summaries, fuzzing summaries, evidence metadata, warning entries, and report validation rules.
- **Section 11: Certification Workflow.** The seven-phase end-to-end process from initial engagement through conformance certificate issuance: Engagement, ICS Submission, Pre-Assessment, Test Execution, Verdict Resolution, Report Publication, and Maintenance. Includes the Conformance Claim State Machine governing certificate lifecycle transitions.
- **Section 12: Self-Certification vs. Third-Party Evaluation.** The graduated trust model architecture: self-certification at MAI-C0, CTE-assisted evaluation at MAI-C1, accredited third-party evaluation at MAI-C2. Self-certification economics, anti-gaming provisions, and the structural limitations of each assessment model.
- **Section 13: Maintenance and Non-Weakening.** The versioning policy for CTS-1 evolution, the Non-Weakening Clause (future revisions **SHALL** not reduce conformance requirements), maintenance governance, IETF protocol tracking procedures, and the formal process for incorporating new RATS protocol versions.
- **Section 14: Implementation Guidance and Request for Guidance (RFG).** The formal mechanism for identifying, documenting, and resolving protocol ambiguities discovered during conformance testing. The RFG process provides a structured path for implementers to request clarification on specification edge cases without unilaterally interpreting normative requirements.

These six sections collectively define *how* CTS-1 is operationalized: the tooling, the workflow, the reporting, the certification lifecycle, and the governance of the test suite itself. They transform CTS-1 from a test specification into a deployable certification infrastructure.

Contact: UncleBroFields@proton.me fieldsryanchristopher@gmail.com

10 Honest Limitations

CTS-1 is a conformance test suite, not a safety guarantee. This section states explicitly what CTS-1 can and cannot provide, following the honest framing principle that governs the entire Auburn Governance Stack.

10.1 What CTS-1 Provides

1. **Deterministic, binary compliance determination.** A system either passes or fails. There is no partial credit, no grading curve, and no interpretive discretion. This eliminates the "we're working toward compliance" defense.

2. **Reproducible results.** Any party with access to the system's attestation endpoint can independently execute CTS-1 and obtain the same pass/fail determination. Conformance is not a matter of opinion.

3. **Complete normative coverage.** Every **MUST** and **SHALL** statement in MAI-1 §6–§9 traces to at least one CTS-1 assertion. There are no untested normative requirements.

4. **Negative test coverage.** CTS-1 verifies not only that valid artifacts are accepted but that invalid artifacts are rejected. Security assurance requires both.

5. **Statistical rigor for continuous properties.** Invariant measurements that are inherently stochastic are evaluated using TSPRT with controlled error rates, not ad hoc thresholds.

6. **An accountability infrastructure.** CTS-1 creates a public, deterministic standard against which any claim of MAI-1 conformance can be independently verified. This is the precondition for procurement enforcement, insurance underwriting, and regulatory compliance.

10.2 What CTS-1 Does Not Provide

1. **Behavioral safety guarantees.** CTS-1 verifies that a system accurately reports its governance state. It does not and cannot verify that the system will behave safely. Rice's theorem establishes that no finite test suite can determine arbitrary behavioral properties of a Turing-complete system. CTS-1 does not claim to circumvent this fundamental limitation.

2. **Measurement truthfulness at C0.** At MAI-C0 (self-certification), CTS-1 verifies artifact conformance—structural validity, correct encoding, proper signing—but cannot verify that the measurements within the artifact are truthful. Measurement truthfulness assurance increases at C1 (TEE-rooted signing) and C2 (adversarial behavioral testing), but no conformance test can provide absolute truthfulness guarantees.

3. **Protection against sophisticated adversarial gaming.** A sufficiently sophisticated adversary with detailed knowledge of the test suite can potentially construct a system that passes CTS-1 while violating the spirit of MAI-1. CTS-1 mitigates this through negative testing, behavioral probing, randomized sampling (C2), and the non-predictability of CTE-selected test parameters—but does not claim adversarial invulnerability.

4. **Continuous monitoring.** CTS-1 is a point-in-time conformance evaluation, not a continuous monitoring system. A system that passes CTS-1 today may fail tomorrow if its configuration, model, firmware, or operational context changes. The freshness requirements in MAI-1 (§9) and the re-attestation trigger taxonomy (Document 30) address continuous assurance—CTS-1 addresses initial and periodic conformance evaluation.

5. **Physical security guarantees.** CTS-1 relies on TEE attestation for hardware root-of-trust evidence. TEEs have documented physical attack surfaces (voltage glitching, electromagnetic fault injection, side-channel extraction). CTS-1 tests the cryptographic and logical properties of TEE-generated evidence but cannot verify the physical integrity of the hardware. Document 7 (TEE Side-Channel Honest Disclosure) addresses these limitations transparently.

10.3 The Governance Analogy

Domain	Certification Standard	What It Does Not Guarantee
Financial Auditing	SOX §404, PCAOB AS 2201	Future solvency, absence of fraud, investment returns
Cryptographic Modules	FIPS 140-3	Absence of implementation bugs, resistance to novel attacks, correct usage by applications
Information Security	ISO 27001	Absence of breaches, data confidentiality under all conditions, zero-day resilience
Medical Devices	FDA 510(k), PMA	Clinical efficacy in every patient, absence of adverse events, long-term safety
AI Governance	**CTS-1 (MAI-1)**	**Behavioral safety, output correctness, absence of hallucination, alignment guarantee**

Honest Framing

Every certification regime in every regulated industry certifies *process compliance*—that the right measurements were taken, the right controls were in place, and the results are reproducible and auditable. No certification regime guarantees *outcomes*. Financial auditing does not prevent fraud. FIPS 140-3 does not prevent cryptographic breaks. ISO 27001 does not prevent breaches. CTS-1 does not prevent AI failures.

What CTS-1 provides is the *precondition* for accountability: a public, deterministic, reproducible standard against which AI governance claims can be independently verified. Without this precondition, AI governance is aspiration. With it, AI governance becomes infrastructure.

References

[1] L. Lundblade, G. Mandyam, J. O'Donoghue, and C. Wallace, "The Entity Attestation Token (EAT)," RFC 9711, Internet Engineering Task Force, April 2025.

[2] H. Birkholz, T. Fossati, Y. Deshpande, N. Smith, and W. Pan, "Remote ATtestation procedureS (RATS) Architecture," RFC 9334, Internet Engineering Task Force, January 2023.

[3] J. Schaad, "CBOR Object Signing and Encryption (COSE): Structures and Process," RFC 9052, Internet Engineering Task Force, August 2022.

[4] C. Bormann and P. Hoffman, "Concise Binary Object Representation (CBOR)," RFC 8949, Internet Engineering Task Force, December 2020.

[5] H. Birkholz and T. Fossati, "Concise Reference Integrity Manifest (CoRIM)," draft-ietf-rats-corim-09, Internet Engineering Task Force, 2025.

[6] H. Birkholz, N. Smith, T. Fossati, and T. Hardjono, "RATS Conceptual Message Wrapper (CMW)," draft-ietf-rats-msg-wrap-16, Internet Engineering Task Force, 2025.

[7] H. Brossard, A. Delignat-Lavaud, and C. Fournet, "An Architecture for Non-Interactive Attestation of Supply Chain Integrity (SCITT)," draft-ietf-scitt-architecture, Internet Engineering Task Force, 2025.

[8] H. Birkholz, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures," RFC 8610, Internet Engineering Task Force, June 2019.

[9] Common Criteria Recognition Arrangement, "Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components," ISO/IEC 15408-3:2022, Version 3.1 Revision 5.

[10] International Organization for Standardization, "Common Methodology for Information Technology Security Evaluation," ISO/IEC 18045:2022.

[11] National Institute of Standards and Technology, "Security Requirements for Cryptographic Modules," FIPS 140-3, March 2019.

[12] National Institute of Standards and Technology, "Derived Test Requirements for FIPS 140-3," NIST SP 800-140, December 2020.

[13] R. Fields, "The Model Attestation Interface (MAI-1): A Normative Profile and Conformance Protocol for Foundation Model Governance," Auburn Patent Family, Clause AI-5, 2026.

[14] R. Fields, "The Model State Attestation Framework: Evidence-Based Governance for Foundation Models," Auburn Patent Family, 2026.

[15] R. Fields, "Auburn Governance Stack: Master Architecture Plan," Auburn Patent Family, AGS-1, 2026.

[16] R. Fields, "Clause AI-8: The Entropy-Collapse Constraint — Mandatory Policy Diversity Floor for Self-Play, Multi-Agent, and Sampling-Ensemble Systems," Auburn Patent Family, 2026.

[17] R. Fields, "Clause AI-2: The Gradient Starvation Envelope," Auburn Patent Family, 2026.

[18] R. Fields, "Clause AI-3: Lyapunov Stability Envelopes for Speculative Decoding," Auburn Patent Family, 2026.

[19] R. Fields, "Clause AI-4: SRAM Thermal Integrity Bound for Fused Attention Kernels," Auburn Patent Family, 2026.

[20] R. Fields, "The Stateful Isolation Law," Auburn Patent Family, 2026.

[21] R. Fields, "Attention Thermodynamics," Auburn Patent Family, 2026.

[22] R. Fields, "Rails: A Governance Protocol for Artificial Intelligence — Alpha Specification," Auburn Patent Family, 2026.

[23] OpenID Foundation, "OpenID Connect Conformance Test Suite," 2024. https://openid.net/certification/

[24] A. Wald, "Sequential Tests of Statistical Hypotheses," *The Annals of Mathematical Statistics*, vol. 16, no. 2, pp. 117–186, 1945.

[25] Veraison Project, "Verification of Attestation," 2025. https://github.com/veraison

A Glossary

Term	Definition
Assertion (AS)	A single testable requirement derived from a MAI-1 normative statement. Identified by a four-part code: AS-{Pyramid}-{Layer}-{Seq}.
Attester	The entity (system under test) that produces attestation evidence. In the RATS architecture, the Attester generates Claims about its own state.
CBOR	Concise Binary Object Representation (RFC 8949). The canonical encoding for MAI-1 attestation artifacts.
CDDL	Concise Data Definition Language (RFC 8610). The schema language for defining MAI-1 payload structure.
Compliance Flag	The GREEN/YELLOW/RED status derived from comparing measured invariant values against certified thresholds.
CoRIM	Concise Reference Integrity Manifest. Carries certified reference values (thresholds) against which attestation evidence is compared.
COSE	CBOR Object Signing and Encryption (RFC 9052/9053). The signing envelope for MAI-1 attestation artifacts.
CTE	Conformance Testing Entity. The independent party that executes CTS-1 against the system under test. Required at C1 (assisted) and C2 (accredited).
CTS-1	Conformance Test Suite version 1. This document. The binary pass/fail enforcement mechanism for MAI-1.
Decision Receipt	A cryptographically signed record binding a specific inference output to the governance state of the system at the time of generation.
EAT	Entity Attestation Token (RFC 9711). The IETF standard token format that MAI-1 extends as a constrained profile.
ICS	Implementation Conformance Statement. The vendor's declaration of which MAI-1 features and conformance level their system implements.
MAI-1	Model Attestation Interface version 1. The normative specification that CTS-1 tests against. Clause AI-5 of the Auburn Patent Family.
Merkle Tree	A hash-based data structure binding Layer 1, Layer 2, and Layer 3 evidence into a single signed root hash, enabling selective disclosure and tamper detection.
SCITT	Supply Chain Integrity, Transparency, and Trust. The IETF framework for transparency log registration of provenance claims.

Term	Definition
TEE	Trusted Execution Environment. Hardware-isolated execution environment providing confidentiality and integrity guarantees for code and data.
TSPRT	Truncated Sequential Probability Ratio Test. The statistical methodology for evaluating continuous invariant measurements.
Verifier	The entity that evaluates attestation evidence against reference values and policy. In CTS-1, the test harness acts as Verifier.

B Complete Assertion Index

The following table provides the complete assertion registry in compact reference form. This table is designed for copy-paste into procurement documents, RFP requirements, and compliance checklists.

ID	Title	Tier	Level	Src
General Encoding & Endpoint (MAI-1 §6.1, §6.3)				
AS-ST-EP-01	Canonical Endpoint Availability	C0+	L1	§6.1
AS-ST-EP-02	Request Body Schema	C0+	L1	§6.1
AS-ST-EP-03	COSE_Sign1 Envelope Structure	C0+	L1	§6.3
AS-ST-EP-04	Canonical CBOR Encoding	C0+	L1	§6.3
AS-ST-EP-05	Field Completeness	C0+	L1	§6.3
Layer 1: Platform Attestation (MAI-1 §6.2.1)				
AS-ST-L1-01	TEE Type Declaration	C0+	L1	§6.2.1
AS-ST-L1-02	Firmware Measurement Chain	C0+	L1	§6.2.1
AS-ST-L1-03	Boot Seed and Boot Count	C0+	L1	§6.2.1
AS-CR-L1-04	Hardware Attestation Quote	C1+	L4	§6.2.1
AS-CR-L1-05	Signing Key TEE Binding	C1+	L4	§6.3
AS-SM-L1-06	Platform Identity Consistency	C0+	L2	§6.2.1
AS-SM-L1-07	Firmware Version Reporting	C0+	L2	§6.2.1
Layer 2: Model State Invariants (MAI-1 §6.2.2, §7)				
AS-ST-L2-01	Model Identity Fields	C0+	L1	§6.2.2
AS-ST-L2-02	Invariant Measurement Array Structure	C0+	L1	§6.2.2
AS-ST-L2-03	Mandatory Invariant Presence	C1+	L1	§7
AS-SM-L2-04	Invariant Range Plausibility	C0+	L2	§7
AS-SM-L2-05	Threshold Ordering & Consistency	C0+	L2	§7
AS-SM-L2-06	Compliance Flag Logic	C0+	L2	§6.2.2
AS-SM-L2-07	Entropy Floor Validation	C1+	L2	§7.1
AS-SM-L2-08	Gradient Stability Validation	C1+	L2	§7.2
AS-SM-L2-09	Distribution Drift Validation	C1+	L2	§7.3
AS-SM-L2-10	Structural Coherence Validation	C1+	L2	§7.4
AS-SM-L2-11	SRAM Thermal Integrity Validation	C1+	L2	§7.5
AS-SM-L2-12	Measurement Frequency Compliance	C1+	L2	§7.6
AS-BH-L2-13	Invariant Breach Response	C2	L3	§7
Layer 3: Provenance Binding (MAI-1 §6.2.3)				
AS-ST-L3-01	AI-BOM Presence and Structure	C0+	L1	§6.2.3
AS-ST-L3-02	Training Provenance Chain	C0+	L1	§6.2.3
AS-CR-L3-03	SCITT Receipt Verification	C1+	L4	§6.2.3
AS-SM-L3-04	Provenance Chain Integrity	C1+	L2	§6.2.3
AS-SM-L3-05	Contamination Detection Declaration	C2	L2	§6.2.3
AS-BH-L3-06	Provenance Tamper Detection	C2	L3	§6.2.3
Execution Context & Signature (MAI-1 §6.2.4, §6.3)				
AS-ST-EC-01	Inference Identity Fields	C0+	L1	§6.2.4
AS-SM-EC-02	Temporal Consistency	C0+	L2	§6.2.4
AS-CR-EC-03	Nonce Binding	C0+	L4	§6.3
AS-CR-EC-04	Signature Algorithm Compliance	C0+	L4	§6.3
AS-CR-EC-05	Certificate Chain Validation	C0+	L4	§6.3
AS-BH-EC-06	Concurrent Request Isolation	C1+	L3	§6.1

ID	Title	Tier	Level	Src
Cryptographic Binding & Decision Receipts (MAI-1 §8)				
AS-CR-CB-01	Merkle Root Integrity	C1+	L4	§8.1
AS-CR-CB-02	Decision Receipt Binding	C1+	L4	§8.2
AS-CR-CB-03	Selective Disclosure Proof	C2	L4	§8.3
AS-BH-CB-04	Tamper Detection Across Layers	C2	L3	§8.1
Conformance Level Requirements (MAI-1 §9)				
AS-SM-CL-01	Level Declaration Consistency	C0+	L2	§9
AS-SM-CL-02	Freshness Compliance per Level	C0+	L2	§9
AS-BH-CL-03	Conformance Level Escalation	C1+	L3	§9

Intellectual Property Declaration

Auburn Patent Family Fields

Intellectual Property (IP) Declaration

The methods, logic structures, test architectures, assertion registries, and conformance methodologies contained in this work are the sole property of **Ryan Fields**.

Public License (Non-Commercial)

This work is licensed under the **Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0)** license.

- **Academic Use:** Researchers may share and use this framework for non-commercial academic purposes, provided full attribution is given to Ryan Fields.
- **No Derivatives:** No modifications or "remixes" of the assertion registry, test architectures, or conformance methodologies are permitted without express written consent.

Commercial Prohibition

Commercial use of this framework is strictly prohibited. This includes, but is not limited to:

- Use within proprietary high-frequency trading (HFT) risk models.
- Integration into commercial high-assurance AI governance software.
- Use by private financial institutions for "tail-risk" auditing of prime distribution variance.
- Implementation of CTS-1 test procedures in commercial conformance testing products or services.
- Use of the assertion registry or test architecture in commercial AI certification programs.

Email: UncleBroFields@proton.me
Email: fieldsryanchristopher@gmail.com

CTS-1: MAI-1 Conformance Test Suite — Public Specification v1.0
Document Number: AGS-DOC-026 | Auburn Governance Stack — Enforcement Layer