Trust-State Standard v1.1 — Normative Specification

The Trust-State Standard defines deterministic requirements for canonical evidence serialization, rule identity derivation, conformity artifact construction, and replay-equivalent evaluation.

Trust-State Standard v1.1

Status: Official Release

Publication Date: June 8, 2026

• Corrected cross-references and canonical publication artifact.

• No normative clauses modified.

Published by VTI Foundation, Inc.

Overview

Version 1.1 establishes the core normative framework for Trust-State deterministic integrity evaluation.

Contents

Canonical Release

The official archival edition of **Trust-State Standard v1.1** is published as a versioned, immutable PDF.

Download:

Trust-State Standard v1.1 — Canonical PDF

SHA-256: Published with canonical release record.

Publication Date: June 8, 2026

Authority: VTI Foundation, Inc.

The PDF edition constitutes the authoritative release record for Version 1.1 and MAY be cited in regulatory, academic, or conformance documentation.

Abstract

The Trust-State Standard defines a deterministic framework for evaluating and recording the integrity and authorization state of digital systems.

The specification establishes canonical evidence serialization requirements, rule identity derivation mechanisms, and replay-equivalent evaluation constraints necessary to ensure consistent trust-state determination across heterogeneous execution environments.

Conformance to this standard requires strict adherence to the defined canonicalization, hashing, invariant binding, and conformity artifact construction procedures specified herein. Identical canonical inputs and rule identities SHALL produce identical conformity artifacts without discretionary override.

1. Scope

The Trust-State Standard defines deterministic computational requirements for evaluating, binding, and verifying digital system integrity.

This standard establishes mandatory rules governing:

- Canonical serialization of evaluation inputs.
- Deterministic rule identity derivation.
- Replay-equivalent evaluation execution.
- Conformity artifact construction and integrity binding.
- Conformance validation and certification eligibility.

This standard applies to digital systems requiring reproducible, auditable, and independently verifiable evaluation outcomes.

This standard does not prescribe:

- Programming language.
- System architecture.
- Deployment model.
- Storage medium.
- Transport protocol.

Implementations SHALL satisfy all normative annex requirements in order to claim conformance.

2. Normative References

The following referenced documents are indispensable for the application of this standard.

- FIPS 180-4 — Secure Hash Standard (SHA-256).
- RFC 8259 — The JavaScript Object Notation (JSON) Data Interchange Format.
- Unicode Standard — Unicode Code Point Ordering.
- RFC 2119 — Key words for use in RFCs to Indicate Requirement Levels.

Where referenced specifications conflict with the requirements of this standard, the requirements of the Trust-State Standard SHALL take precedence.

All hash computations defined within this standard SHALL use SHA-256 as defined in FIPS 180-4.

3. Definitions

3.1 Canonical Serialization

The deterministic transformation of structured data into a byte-identical UTF-8 encoded representation according to Annex A.

3.2 Canonical Evidence

Evaluation input data serialized in canonical form and used as the basis for hash derivation and replay validation.

3.3 Rule Identity

A cryptographic hash derived from the canonical serialized representation of a rule definition as defined in Annex B.

3.4 Deterministic Replay

Re-execution of evaluation logic under identical canonical evidence and rule identity conditions producing identical outcomes.

3.5 Replay Equivalence

A Boolean condition evaluating TRUE only if recomputed evidence hash, rule identity hash, decision, state, and artifact hash match original issuance values.

3.6 Conformity Artifact

A structured record containing evidence hash, rule hash, evaluation outcome, and artifact hash as defined in Annex E.

3.7 Conformance

Verified adherence to all mandatory sections and annexes of the applicable published version of the Trust-State Standard.

3.8 Certification

Formal recognition that an implementation satisfies all conformance requirements under Annex F.

4. Canonical Evidence

This section defines deterministic requirements for canonical serialization and evidence hashing.

4.1 Canonical Serialization Requirements

This standard does not mandate a specific serialization format. Implementations MAY use any structured data representation provided that canonicalization produces deterministic and reproducible outputs consistent with this section.

Canonical serialization MUST:

- Preserve field names and values without semantic alteration.
- Use a consistent character encoding.
- Apply deterministic ordering of object members.
- Exclude non-deterministic metadata unless explicitly defined by this standard.

Implementations MUST NOT introduce runtime-dependent, environment-dependent, or time-dependent fields into canonical evidence unless explicitly required by rule definitions.

4.2 Evidence Hash Derivation

Implementations MUST compute a collision-resistant hash of the canonical serialized evidence.

The resulting hash value MUST be reproducible across independent implementations when provided identical canonical evidence inputs.

The selected hash function MUST be disclosed in the Conformance Statement as specified in Section 6.5.

4.3 Deterministic Constraints

Canonical serialization and hash derivation procedures MUST produce identical outputs when executed in independent environments using identical inputs.

Any deviation from deterministic serialization requirements renders the resulting conformity artifact non-conformant with this standard.

5. Rule Identity

This section defines deterministic requirements for rule identity derivation and binding of evaluation results to invariant rule definitions.

5.1 Invariant Rule Definition

Rule definitions consumed by conformant implementations MUST be treated as invariant logical structures for purposes of identity derivation.

Human-readable version identifiers MAY be used for administrative purposes, but such identifiers MUST NOT substitute for or override the deterministically derived rule identity defined in this section.

Any modification to rule logic, rule parameters that affect evaluation semantics, or rule composition structure MUST result in a new rule identity.

5.2 Canonical Rule Serialization

Rule definitions MUST be serialized in a deterministic manner consistent with the canonicalization requirements defined in Section 4.1.

Identical rule definitions MUST produce identical serialized outputs across independent implementations.

5.3 Rule Identity Hash Derivation

Implementations MUST derive a collision-resistant identifier from the canonical serialized rule definition.

The resulting rule identity MUST uniquely represent the semantic content of the rule definition.

The hash function used for rule identity derivation MUST meet the collision-resistance requirements defined in Section 2.

5.4 Rule Identity Binding

Conformity artifacts MUST include the rule identity associated with the evaluation that produced the artifact.

Implementations MUST NOT substitute or reinterpret rule identities post-evaluation.

Evaluation results that cannot be bound to a deterministically derived rule identity are non-conformant with this standard.

6. Conformance

This section defines mandatory requirements that an implementation MUST satisfy in order to claim conformance with the referenced published version of the Trust-State Standard.

6.1 Normative Annex Binding

Conformance SHALL require full adherence to the following normative annexes:

- Annex A — Canonical Serialization
- Annex B — Rule Identity Derivation
- Annex C — Deterministic Replay Equivalence
- Annex D — Deterministic Test Vectors
- Annex E — Conformity Artifact Structure
- Annex F — Conformance & Certification Criteria

Failure to comply with any mandatory annex SHALL constitute non-conformance.

6.2 Conformance Requirements

An implementation claiming conformance MUST:

- Produce byte-identical canonical serialization under Annex A.
- Derive rule identity hashes exactly as defined in Annex B.
- Ensure deterministic replay equivalence as defined in Annex C.
- Construct conformity artifacts exactly as defined in Annex E.
- Successfully reproduce test vector results defined in Annex D.

All hash computations SHALL use SHA-256 as defined in FIPS 180-4. No alternative hash function SHALL be substituted.

6.3 Conformity Artifact Integrity

The artifact hash MUST:

- Be computed over the canonical serialized artifact structure.
- Be independently reproducible.
- Change deterministically upon any modification to artifact content.

Replay under identical canonical inputs and rule identity conditions MUST reproduce an identical artifact hash.

6.4 Prohibited Modifications

An implementation MUST NOT claim conformance if:

- Canonical ordering rules are altered.
- Rule identity derivation is modified.
- Replay conditions permit non-deterministic variance.
- Artifact structure fields are omitted or redefined.

6.5 Conformance Declaration

An implementation claiming conformance MUST publish a declaration including:

- The implemented version of the Trust-State Standard.

- The implementation release date.
- A statement affirming adherence to all mandatory annexes.

Conformance claims SHALL explicitly reference the specific published version of the Trust-State Standard.

7. Security Considerations

This standard defines deterministic evaluation and identity binding requirements. It does not, by itself, guarantee system security.

7.1 Deterministic Integrity Boundaries

Conformance to this standard ensures reproducibility of evidence serialization, rule identity derivation, and conformity artifact construction. It does not ensure correctness of rule logic or integrity of external inputs.

7.2 Hash Function Selection

Implementations MUST use collision-resistant hash functions appropriate for current cryptographic best practices.

Use of deprecated or compromised hash algorithms renders the implementation non-conformant.

7.3 Rule Integrity

Rule definitions MUST be protected against unauthorized modification.

This standard assumes that rule definitions supplied to an implementation have not been tampered with prior to canonical serialization and identity derivation.

7.4 Environmental Considerations

This standard does not define transport security, storage encryption, access control mechanisms, or operational controls.

Implementers are responsible for ensuring that deployment environments provide appropriate protections consistent with their risk model.

7.5 Replay Verification

Replay equivalence ensures reproducibility under identical inputs.

It does not prevent malicious replay of valid artifacts in inappropriate operational contexts. Implementations MUST apply contextual validation consistent with their deployment requirements.

7.6 Infrastructure Neutrality

This standard does not require the use of distributed ledger, blockchain, or other consensus-based infrastructure.

Deterministic evidence serialization, rule identity derivation, and conformity artifact construction may be implemented in centralized, distributed, or hybrid environments.

8. Governance

Steward: VTI Foundation, Inc.

Applies To: Trust-State Standard

8.1. Stewardship Authority

The Trust-State Standard is stewarded exclusively by VTI Foundation, Inc. Stewardship preserves deterministic integrity, version continuity, and normative consistency across all published releases.

VTI Foundation, Inc. is the authoritative publisher of:

- Normative text of the Trust-State Standard;
- Annexes designated as normative;
- Version identifiers and publication records; and
- Official conformance terminology.

No other entity may publish modifications as an official Trust-State release.

8.2. Publication and Versioning

Each release SHALL include:

- A version identifier;

- A publication date;
- A stable canonical URL; and
- A description of normative changes.

Published versions are immutable. Corrections or amendments SHALL be issued as a new version. Prior versions SHALL NOT be modified retroactively under any circumstance.

8.3. Normative and Informative Material

The standard MAY contain both normative and informative content.

- **Normative** content defines mandatory requirements and uses SHALL, SHALL NOT, SHOULD, and MAY as defined in RFC 2119.
- **Informative** content provides explanation or context and is non-binding.

Annexes SHALL explicitly state their designation. If not stated, an annex is informative by default.

8.4. Change Control

Amendments SHALL preserve deterministic conformance guarantees.

- Canonicalization and hashing requirements SHALL remain reproducible;
- Replay-equivalent evaluation constraints SHALL NOT be weakened;
- No amendment may introduce discretionary artifact variance.

Any modification permitting multiple valid conformity artifacts from identical canonical inputs is non-conformant and SHALL NOT be adopted.

8.5. Conformance Claims

Conformance claims SHALL reference a specific published version of the standard.

A conformance claim is valid only if the implementation produces replay-equivalent conformity artifacts under the referenced version.

Conformance SHALL NOT be claimed against unpublished or draft text unless explicitly designated as Draft Conformance.

8.6. Interpretation

In the event of interpretive conflict, the order of precedence is:

1. Normative text in the Standard;
2. Normative annexes;
3. Informative material.

Interpretations SHALL favor determinism, reproducibility, and independent verifiability.

9. Change Log

Version 1.0 — February 2026

- Initial release of the Trust-State Standard.

Version 1.1 — June 8, 2026

• Corrected cross-references and canonical publication artifact.

• No normative clauses modified.

Annex A — Canonical Serialization

Status: Normative

Version: 1.1

Release Date: June 8, 2026

Authority: VTI Foundation, Inc.

This annex defines mandatory canonical serialization requirements for Trust-State evaluation inputs, rule definitions, and conformity artifacts. This annex SHALL be interpreted as an integral component of the Trust-State Standard v1.1.

A.1 Scope

Canonical serialization ensures deterministic byte representation of structured data prior to hashing, replay validation, or artifact issuance.

All systems claiming conformance with the Trust-State Standard SHALL implement canonical serialization exactly as defined in this annex.

A.2 Encoding Requirements

- All serialized data SHALL be encoded in UTF-8.
- Byte Order Mark (BOM) SHALL NOT be included.
- No trailing whitespace SHALL be permitted.
- No indentation or formatting whitespace SHALL be included unless explicitly defined.

A.3 Object Member Ordering

JSON object members SHALL be ordered lexicographically by key using Unicode code point ordering.

- Ordering SHALL be deterministic.
- Nested objects SHALL apply the same ordering recursively.
- Arrays SHALL preserve original element order.

A.4 Data Normalization Rules

- Strings SHALL be case-sensitive and preserved exactly.
- Numeric values SHALL NOT include leading zeros.

- Boolean values SHALL be lowercase (true, false).
- Null values SHALL be serialized as null.

A.5 Canonical Form Definition

The canonical form of a structured object is defined as the UTF-8 encoded byte sequence resulting from:

1. Recursive lexicographic ordering of all object keys.
2. Removal of non-semantic whitespace.
3. Preservation of exact string and numeric representations.

The resulting byte sequence SHALL be used for all hash computations defined in subsequent annexes.

A.6 Determinism Requirement

Given identical structured input, canonical serialization MUST produce a byte-identical output across independent implementations.

Any deviation in byte sequence SHALL constitute non-conformance.

A.7 Relationship to Hashing

All hash derivations defined in the Trust-State Standard SHALL operate exclusively on canonical serialized byte sequences produced under this annex.

A.8 Normative Language

The key words "MUST", "SHALL", "SHOULD", and "MAY" in this annex are to be interpreted as described in RFC 2119.

Annex B — Rule Identity Derivation

Status: Normative

Version: 1.1

Release Date: June 8, 2026

Authority: VTI Foundation, Inc.

This annex defines deterministic rule identity derivation requirements for the Trust-State Standard.

1. Scope

Rule identity ensures that invariant logic used during evaluation is cryptographically bound to conformity artifacts and replay validation.

All implementations SHALL derive rule identity exactly as defined herein.

2. Canonical Rule Representation

A rule definition SHALL be expressed as a structured object subject to canonical serialization as defined in Annex A.

- All rule fields SHALL be serialized deterministically.
- Rule logic expressions SHALL be preserved exactly as defined.
- No implicit defaults SHALL be assumed.

3. Rule Hash Derivation

Rule identity SHALL be defined as:

RULE_HASH = SHA256(canonical_rule_byte_sequence)

The canonical_rule_byte_sequence SHALL be the UTF-8 encoded byte sequence produced under Annex A.

4. Determinism Requirement

Given identical rule structure and content, independent implementations MUST derive byte-identical canonical representations and identical RULE_HASH values.

Any variation in canonical representation or hash output SHALL constitute non-conformance.

5. Rule Versioning

Modification of any rule field, logic expression, or structural component SHALL result in a new canonical byte sequence and therefore a new RULE_HASH.

Rule identity SHALL NOT be overridden, aliased, or substituted.

6. Binding Requirement

RULE_HASH SHALL be embedded within conformity artifacts and used as a mandatory input to deterministic replay validation.

Annex C — Deterministic Replay Equivalence

Status: Normative

Version: 1.1

Release Date: June 8, 2026

Authority: VTI Foundation, Inc.

This annex defines deterministic replay equivalence requirements for Trust-State evaluations.

1. Scope

Deterministic replay ensures that identical evidence and rule identity produce identical evaluation outcomes across independent executions.

2. Replay Inputs

Replay SHALL be executed using:

- Canonical evidence byte sequence
- Derived RULE_HASH
- Invariant evaluation logic bound to the rule definition

3. Replay Equivalence Condition

Replay equivalence SHALL evaluate TRUE if and only if:

(recomputed_evidence_hash == original_evidence_hash)

AND

(recomputed_rule_hash == original_rule_hash)

AND

(recomputed_decision == original_decision)

AND

(recomputed_state == original_state)

4. Determinism Requirement

Evaluation logic SHALL be free of:

- Non-deterministic randomness
- External time dependencies
- Uncontrolled environmental variables

Any non-deterministic influence SHALL invalidate replay equivalence.

5. Failure Condition

If replay equivalence evaluates FALSE under identical canonical inputs and rule identity, the implementation SHALL be considered non-conformant.

6. Relationship to Certification

Deterministic replay equivalence is a mandatory condition for Trust-State conformance and certification eligibility**.**

Annex D — Deterministic Test Vectors

Status: Normative

Version: 1.1

Release Date: June 8, 2026

Authority: VTI Foundation, Inc.

This annex defines deterministic reference vectors supporting validation of canonical serialization, rule identity derivation, deterministic replay equivalence, and conformity artifact hashing.

1. Hashing Requirements

- All hashing operations SHALL use SHA-256 as defined in FIPS 180-4.
- All canonical forms SHALL be encoded using UTF-8 without BOM.
- No additional whitespace SHALL be introduced beyond canonical form.
- Hash computation SHALL operate exclusively on canonical byte sequences.

2. Canonical Serialization Vector

{"artifact":{"decision":"approve"},"context":{"a":1,"b":2},"expected":{"decision":"approve","state":"valid"},"input":{"x":3,"y":9},"metadata":{"system":"demo","timestamp":"2026-01-01T00:00:00Z","version":"1.0"}}

EVIDENCE_HASH_A1 = SHA256(canonical UTF-8 byte sequence above)

3. Rule Identity Vector

{"decision_logic":{"else":"deny","if":"input.age >= min_age","then":"approve"},"min_age":18,"rule":"min-age-check"}

RULE_HASH_B1 = SHA256(canonical UTF-8 byte sequence above)

4. Replay Equivalence Vector

(recomputed_evidence_hash == EVIDENCE_HASH_A1) AND

(resolved_rule_hash == RULE_HASH_B1) AND

(recomputed_decision == "approve") AND

(recomputed_state == "valid")

Replay SHALL evaluate TRUE only if all conditions are satisfied.

5. Artifact Hash Vector

ARTIFACT_HASH_D1 = SHA256(canonical_conformity_artifact_structure)

6. Conformance Condition

An implementation SHALL be considered Trust-State conformant under this annex only if all canonical forms, hash derivations, and replay evaluations produce byte-identical and deterministic results.

Annex E — Conformity Artifact Structure

Status: Normative

Version: 1.1

Release Date: June 8, 2026

Authority: VTI Foundation, Inc.

This annex defines required structural elements and integrity binding requirements for all Trust-State conformity artifacts.

1. Required Fields

A conformity artifact SHALL include the following fields:

- decision
- evidence_hash
- rule_hash
- outcome_hash
- timestamp

2. Canonical Structure

The artifact SHALL be serialized in canonical form as defined in Annex A.

All fields SHALL be present. Field omission SHALL constitute non-conformance.

3. Outcome Hash Derivation

Outcome hash SHALL be computed as:

OUTCOME_HASH = SHA256(decision || "|" || state)

The delimiter SHALL be the ASCII pipe character (0x7C).

4. Artifact Hash Binding

The complete canonical artifact structure SHALL form the input to artifact hash derivation.

ARTIFACT_HASH = SHA256(canonical_artifact_byte_sequence)

5. Integrity Requirement

Any modification to artifact fields SHALL produce a distinct artifact hash.

Artifact integrity SHALL be independently recomputable and verifiable.

6. Immutability Requirement

Issued conformity artifacts SHALL NOT be retroactively altered. Any state transition SHALL result in issuance of a new artifact.

Annex F — Conformance & Certification Criteria

Status: Normative

Version: 1.1

Release Date: June 8, 2026

Authority: VTI Foundation, Inc.

This annex defines conformance verification requirements and certification issuance criteria under the Trust-State Standard.

1. Conformance Requirements

An implementation SHALL be considered conformant only if:

- Canonical serialization is byte-identical across independent executions.
- Rule identity hashes are deterministically reproducible.
- Replay equivalence evaluates TRUE under identical inputs.
- Artifact hash recomputation matches original issuance value.

2. Validation Process

Conformance validation SHALL include:

- Independent recomputation of evidence hash
- Independent recomputation of rule hash
- Replay execution verification
- Artifact integrity validation

3. Certification Eligibility

Certification MAY be granted only upon successful validation under all mandatory annex conditions.

Certification SHALL be:

- Version-bound
- Rule-identity-bound

- Revocable upon non-conformance

4. Ongoing Compliance

Certified systems SHALL maintain deterministic replay capability throughout the certification period.

Failure to maintain replay equivalence SHALL result in certification suspension.

5. Governance Authority

Certification governance SHALL be administered by VTI Foundation, Inc. or its designated accredited validators**.**

Annex Z — Editorial Governance Annex

Status: Informative

Version: 1.1

Release Date: June 8, 2026

Authority: VTI Foundation, Inc.

Governing Document Control and Language Discipline

Z.1 Status and Purpose

This Annex establishes editorial governance controls for the drafting, revision, interpretation, and maintenance of the Trust-State Standard.

This Annex governs:

- Terminology stability
- Modal verb discipline
- Normative and informative separation
- Definition control
- Structural consistency
- Version continuity discipline

This Annex does not create implementation requirements.

This Annex does not modify normative clauses.

This Annex governs the maintenance and editorial integrity of the Standard as a publication artifact.

Z.2 Relationship to Normative Content

Normative sections of the Trust-State Standard establish requirements applicable to implementations.

This Annex governs the editorial discipline under which normative sections are drafted and maintained.

In the event of inconsistency between editorial guidance in this Annex and a normative clause in the Standard, the normative clause controls.

Z.3 Editorial Governance Authority

Editorial governance of the Trust-State Standard is maintained under the standards maintenance function of VTI Foundation Inc.

Editorial governance is distinct from:

- Certification administration
- Certification decision issuance
- Conformance evaluation
- Registry administration

Editorial governance does not create certification authority.

Editorial governance does not modify certification program policy.

Z.4 Stability Principle

Defined terms remain stable across revisions unless formally amended through version increment and change record.

Normative modal usage remains consistent across releases.

Editorial revisions do not alter substantive requirements without version designation.

Informative material is not expanded in a manner that creates implicit normative obligations.

Z.5 Interpretation Discipline

Defined terms control interpretation of the Standard.

Normative clauses govern implementation requirements.

Informative clauses provide explanation and do not create requirements.

Editorial revision does not introduce interpretive ambiguity.

Citation

When referencing the Trust-State Standard in technical documentation, academic publications, regulatory submissions, or commercial materials, the following citation format SHOULD be used:

VTI Foundation, Inc.

Trust-State Standard v1.1

June 8, 2026

Available at: https://truststatestandard.org/standard/

Conformance claims MUST reference the specific version number of the standard.