Universal Structural Integrity Standard (GSIS)
Normative Minimum Safety Framework

Governance Systems Architecture

2026

Contents

1. Mission Statement

1.1 Purpose

The Universal Structural Integrity Standard (GSIS) defines a normative minimum safety framework for evaluating irreversible or high-impact decision processes.

The standard establishes requirements for determining whether a resulting system state could have been produced through structurally permissible transitions.

The objective is prevention of harmful outcomes through verification of process validity, not retrospective assignment of motive or fault.

1.2 Design Principle

The standard operates on the principle:

> When a transition produces irreversible impact, the required verification threshold shall increase proportionally to the difficulty of correcting the resulting state.

Accordingly, GSIS evaluates structural permissibility rather than subjective intent.

1.3 Scope

This standard applies to systems in which outcomes depend on ordered actions performed by one or more independent actors, including but not limited to:

- administrative decision processes
- safety-critical operations
- regulatory enforcement workflows
- automated decision systems
- multi-party adjudication procedures

The standard is domain-neutral and does not prescribe policy outcomes. It evaluates only whether claimed outcomes are structurally supportable.

1.4 Non-Goals

This standard does not:

- determine moral culpability
- evaluate political desirability
- replace democratic decision-making
- interpret subjective narratives

The standard provides a verification layer that allows independent observers to determine whether a state transition was procedurally permissible.

1.5 Structural Accountability Principle

Accountability within GSIS is defined as traceability of a state transition to its enabling conditions.

A decision is considered structurally accountable when:

$$\text{Outcome} \rightarrow \text{Transition} \rightarrow \text{Guards} \rightarrow \text{Actor} \rightarrow \text{Enabling Rule}$$

All elements in the chain shall be deterministically reconstructable from recorded events.

1.6 Safety Orientation

GSIS treats certain transitions as safety-critical. For such transitions:

- absence of verification is not interpreted as validity
- unverifiable transitions remain unresolved
- impossible states invalidate dependent decisions

1.7 Operational Philosophy

The standard assumes that complex systems fail primarily through structural misalignment rather than intentional misconduct.

Therefore GSIS measures:

- whether a transition could occur
- whether prerequisites existed
- whether contradictions are present
- whether the resulting state is reachable

Interpretation of motivation is explicitly outside the scope of the system.

2. Definitions

For purposes of this standard, the following terms have the meanings set forth below. These definitions are intended to ensure consistent interpretation across domains.

2.1 Event

A discrete recorded action, occurrence, or assertion generated by an actor. An event may include metadata such as timestamp, source identifier, supporting reference, or dependency information.

2.2 Actor

Any entity capable of generating an event, including a human individual, automated system, software process, hardware device, or institutional authority.

2.3 State

A representation of the cumulative condition of a system derived from one or more processed events.

A state may be fully defined or partially defined when information is incomplete.

2.4 State Transition

A change from one state representation to another resulting from processing an event.

2.5 Irreversible Transition

A state transition whose effects cannot be fully undone without external corrective intervention.

2.6 Guard

A prerequisite condition that must be satisfied before a state transition is considered structurally permissible.

Guards may reference:

- prior events,
- current state attributes,
- dependency relationships,
- exclusivity constraints,
- external verification sources.

2.7 Valid Transition

A state transition for which all required guards evaluate as satisfied.

2.8 Invalid Transition

A state transition attempted when one or more required guards evaluate as unsatisfied.

2.9 Unverifiable Transition

A state transition for which guard evaluation cannot be completed due to incomplete, missing, or ambiguous information.

2.10 Impossible State

A state that cannot be derived through any sequence of valid transitions from an initial state under defined guard conditions.

2.11 Structural Integrity

The property that each state in a sequence can be derived from a preceding state using only permissible transitions.

2.12 Transition Integrity Record

A machine-verifiable artifact containing:

- processed event sequence,
- guard evaluation results,
- transition classifications,
- anomaly detections,
- computed integrity metrics.

2.13 Accountability Chain

A deterministically reconstructable linkage:

$$\text{Outcome} \rightarrow \text{Transition} \rightarrow \text{Guards} \rightarrow \text{Actor} \rightarrow \text{Enabling Rule}$$

2.14 Opacity Condition

A structural condition in which the relationship between outcome, mechanism, and decision authority cannot be readily reconstructed from available records.

2.15 Mission Capability Classification

A categorical designation describing whether a system remains structurally capable of producing permissible outcomes under observed transition conditions.

2.16 Minimum Safety Interlock

A mandatory guard condition whose failure automatically blocks an irreversible transition.

3. System Architecture

3.1 Overview

The Universal Structural Integrity Standard defines a verification architecture for evaluating whether a resulting state could be produced through permissible transitions in a multi-actor environment.

The architecture operates as a deterministic evaluation pipeline that converts recorded events into a structural integrity assessment.

The system does not interpret narrative meaning. It evaluates structural continuity.

3.2 Processing Layers

The verification architecture consists of five functional layers:

1. Event Ingestion Layer
2. Guard Validation Layer
3. State Construction Layer
4. Transition Detection Layer
5. Integrity Metric Layer

Each layer produces a reproducible output used by subsequent layers.

3.3 Event Ingestion Layer

The ingestion layer receives events from one or more independent sources.

The layer performs normalization including:

- source attribution
- timestamp normalization
- duplicate detection
- dependency extraction

Events are stored without alteration of original content.

3.4 Guard Validation Layer

For each candidate transition, prerequisite conditions (guards) are evaluated.

Guard evaluation produces one of three outcomes:

- satisfied
- unsatisfied
- indeterminate

No interpretation beyond prerequisite verification is performed.

3.5 State Construction Layer

The system constructs a candidate next state using the current state and event.

The candidate state is accepted only if all required guards evaluate as satisfied. Otherwise the state remains unchanged and the transition is recorded.

3.6 Transition Detection Layer

The system analyzes the sequence of transitions to detect structural anomalies, including:

- missing prerequisite events
- premature transitions
- contradictory states
- unreachable states
- cyclic dependencies

Detected anomalies are recorded without modification of original records.

3.7 Integrity Metric Layer

The system computes quantitative metrics derived solely from transition classifications.

Metrics describe structural reliability of the record independent of narrative interpretation.

3.8 Deterministic Reproducibility

Given the same input event set, the architecture shall produce identical transition classifications and metric outputs.

The verification result shall not depend on evaluator identity, institutional role, or subjective interpretation.

3.9 Output Artifact

The system produces a machine-verifiable artifact containing:

- event dependency relationships
- guard evaluation trace
- transition classifications
- detected anomalies
- computed integrity metrics

The artifact shall allow independent recomputation of results without requiring access to the original evaluator.

4. State Model

4.1 State Evolution Framework

The system models process evolution as a guarded asynchronous state machine.

Let S_t represent the system state after processing events up to index t.

For an incoming event E_t, a candidate next state S^*_{t+1} is constructed using a deterministic state function:

$$S^*_{t+1} = F(S_t, E_t)$$

where F represents the state construction function.

4.2 Guard Evaluation Function

A guard function $G(S_t, E_t)$ evaluates prerequisite conditions required for the candidate transition.

$$G(S_t, E_t) \in \{\text{satisfied}, \text{unsatisfied}, \text{indeterminate}\}$$

The transition classification C_t is defined as:

$$C_t = \begin{cases} \text{valid} & \text{if } G = \text{satisfied} \\ \text{invalid} & \text{if } G = \text{unsatisfied} \\ \text{unverifiable} & \text{if } G = \text{indeterminate} \end{cases}$$

4.3 State Acceptance Rule

The system state is updated according to:

$$S_{t+1} = \begin{cases} S^*_{t+1} & \text{if } C_t = \text{valid} \\ S_t & \text{otherwise} \end{cases}$$

Invalid or unverifiable transitions do not alter the accepted state.

4.4 Partial Ordering of Events

Events may arrive without guaranteed global chronological order.

Let $\prec$ denote a prerequisite dependency relation such that:

$$E_i \prec E_j \Rightarrow E_i \text{ must occur before } E_j$$

If a required predecessor event is absent, the transition shall be classified as invalid or unverifiable depending on guard requirements.

4.5 Impossible State Detection

A state S_t is defined as impossible if no sequence of valid transitions exists from an initial state S_0 to S_t.

Formally:

$$\text{Impossible}(S_t) = \begin{cases} \text{true} & \text{if no valid transition path exists} \\ \text{false} & \text{otherwise} \end{cases}$$

Impossible states invalidate dependent transitions.

4.6 Structural Continuity

Structural continuity exists if every state in the sequence is reachable through a chain of valid transitions:

$$S_0 \rightarrow S_1 \rightarrow \cdots \rightarrow S_n$$

where all intermediate transitions are classified as valid.

Breaks in continuity are recorded as structural defects.

4.7 Recomputation Stability

Given an identical set of input events, recomputation of the state sequence shall produce identical:

- transition classifications
- detected anomalies
- integrity metrics

This property ensures evaluator independence and deterministic verification.

5. Minimum Safety Interlocks

5.1 Purpose

Minimum Safety Interlocks define mandatory guard conditions that must be satisfied before an irreversible or high-impact transition may be accepted.

Interlocks are non-optional. Failure of any required interlock shall block the transition.

5.2 Irreversible Transition Threshold

A transition shall be classified as safety-critical if:

- the resulting state cannot be fully restored through ordinary reversal, or
- the transition materially alters rights, status, access, custody, liberty, property, or institutional standing, or
- the correction cost exceeds the transition cost.

Safety-critical transitions require enhanced guard verification.

5.3 Mandatory Interlock Conditions

For any safety-critical transition, the following minimum guards shall be evaluated:

1. Jurisdiction Validity
2. Notice Confirmation
3. Identity Consistency
4. Evidence Traceability
5. Dependency Path Completeness
6. Contradiction Clearance

5.4 Jurisdiction Validity

The system shall verify that the acting authority possesses defined scope over the subject matter and affected entity.

If jurisdiction cannot be verified, the transition shall be blocked.

5.5 Notice Confirmation

Where applicable, required notification procedures must be recorded and verifiable prior to transition acceptance.

Absence of notice verification shall result in transition blocking.

5.6 Identity Consistency

All referenced entities must maintain consistent identity mapping across event records.

Conflicting or unresolved identity records shall block the transition.

5.7 Evidence Traceability

Any evidentiary basis required for the transition must include:

- traceable origin reference
- unmodified record integrity
- reproducible verification linkage

Untraceable or unsupported evidence shall invalidate the transition.

5.8 Dependency Path Completeness

All prerequisite events required to enable the transition must be present in the record.

Missing required predecessors shall block the transition.

5.9 Contradiction Clearance

If mutually incompatible state attributes are present and unresolved, the transition shall not proceed.

Contradictions must be resolved through additional valid transitions before safety-critical actions are accepted.

5.10 Automatic Blocking Rule

For safety-critical transitions, the validity condition is defined as:

$$\text{Transition Valid} = G_1 \land G_2 \land G_3 \land G_4 \land G_5 \land G_6$$

If any guard evaluates as unsatisfied or indeterminate, the transition classification shall be invalid or unverifiable, and the state shall remain unchanged.

5.11 Override Logging Requirement

If a system permits override capability, the override must:

- be explicitly logged,
- identify the responsible actor,
- record justification,
- trigger automatic review classification.

Silent override is prohibited under this standard.

5.12 Safety Escalation Trigger

Repeated failure of interlocks above a defined threshold shall trigger automatic escalation classification indicating systemic risk.

6. Transition Detection

6.1 Purpose

The transition detection layer identifies structural defects in the sequence of processed events.

The objective is not to interpret meaning but to determine whether the observed state evolution violates prerequisite ordering or produces logically inconsistent conditions.

6.2 Transition Classification History

Each processed event produces a classification:

$$C_t \in \{\text{valid}, \text{invalid}, \text{unverifiable}\}$$

The ordered sequence of classifications forms the transition history:

$$H = \{C_1, C_2, \ldots, C_n\}$$

Structural defect detection operates on this sequence.

6.3 Missing Prerequisite Detection

If a transition depends on a required predecessor event E_p such that:

$$E_p \prec E_t \quad \text{and} \quad E_p \notin \text{Observed Events}$$

the transition shall be classified as invalid or unverifiable.

6.4 Premature Transition Detection

A transition is premature when enabling conditions are unsatisfied at the time of execution.

$$\exists G_i(S_t, E_t) = \text{unsatisfied} \Rightarrow C_t = \text{invalid}$$

6.5 Contradictory State Detection

Let A and B be mutually exclusive state attributes.

If the sequence contains:

$$A \in S_i \quad \text{and} \quad B \in S_j$$

with no valid transition removing A before B appears, a contradiction shall be recorded.

6.6 Unreachable State Detection

A state S_t is unreachable if no valid transition path exists from S_0 to S_t.

$$\neg\exists P = \{C_{i1}, C_{i2}, \ldots, C_{ik}\} \text{ such that all } C_{ij} = \text{valid}$$

Unreachable states invalidate dependent transitions.

6.7 Cyclic Dependency Detection

If transitions form a dependency cycle:

$$E_1 \prec E_2 \prec \cdots \prec E_n \prec E_1$$

all involved transitions shall be classified invalid due to circular prerequisite requirements.

6.8 Deferred Resolution Handling

If guard evaluation is indeterminate, the transition shall remain classified as unverifiable until sufficient information becomes available.

Reevaluation shall not silently alter prior valid or invalid classifications. Any change must generate a trace entry.

6.9 Persistent Detection Record

The system shall maintain a transition detection record containing:

- transition index
- defect classification
- related prerequisite references
- supporting trace identifiers

The detection record must allow independent recomputation of results.

7. Opacity Detection

7.1 Purpose

Opacity detection identifies conditions under which the relationship between outcome, mechanism, and responsible authority cannot be reliably reconstructed by observers.

Opacity does not imply misconduct. Opacity indicates loss of practical accountability resolution.

7.2 Definition

An opacity condition exists when reconstruction of the accountability chain:

$$\text{Outcome} \rightarrow \text{Transition} \rightarrow \text{Guards} \rightarrow \text{Actor} \rightarrow \text{Rule}$$

requires specialized knowledge beyond reasonable observer capability.

7.3 Structural Opacity Factors

The system shall evaluate the following contributing conditions:

1. Diffused Impact Distribution
2. Multi-Actor Dependency Chains
3. Indirect Incentive Structures
4. Record Fragmentation
5. Identity Substitution
6. Guard Indeterminacy Frequency
7. Contradiction Density
8. Technical Complexity Threshold

7.4 Diffused Impact Distribution

Opacity increases when negative outcomes are distributed across many entities such that no individual instance reveals the pattern.

7.5 Multi-Actor Dependency Chains

Let L be the length of the minimal dependency path required to reconstruct a transition.

$$L > L_{human} \Rightarrow \text{opacity increases}$$

where L_{human} represents reasonable human traceability capacity.

7.6 Indirect Incentive Structures

If actor behavior depends on variables not visible within the local decision record, the transition contributes to opacity.

7.7 Record Fragmentation

If required guard verification depends on records stored across multiple independent repositories, reconstruction difficulty increases.

7.8 Identity Substitution

If equivalent functional roles are performed by multiple interchangeable actors, responsibility attribution becomes degraded.

7.9 Guard Indeterminacy Frequency

Let U represent the ratio of unverifiable transitions:

$$U = \frac{T_u}{T_{total}}$$

If U exceeds a defined threshold, the system shall classify the process as structurally opaque.

7.10 Contradiction Density

Let D_c represent contradiction density:

$$D_c = \frac{N_c}{T_{total}}$$

Increasing contradiction density correlates with reduced observer confidence in reconstruction accuracy.

7.11 Technical Complexity Threshold

Opacity exists when correct interpretation requires domain expertise not available to general observers responsible for oversight.

7.12 Opacity Classification

The system shall classify opacity as:

- Transparent
- Complex but Traceable
- Opaque
- Structurally Non-Resolvable

7.13 Operational Consequence

When a system is classified as Opaque or Structurally Non-Resolvable, human accountability mechanisms relying on individual case evaluation shall be considered unreliable.

Such conditions require structural review rather than case-by-case review.

8. Integrity Metrics

8.1 Purpose

Integrity metrics provide quantitative measures describing structural reliability of an observed event sequence.

Metrics are computed solely from transition classifications and do not depend on interpretation of event meaning.

8.2 Transition Counts

Let:

$$T_v = \text{number of valid transitions}$$

$$T_i = \text{number of invalid transitions}$$

$$T_u = \text{number of unverifiable transitions}$$

Total transitions:

$$T_{total} = T_v + T_i + T_u$$

8.3 Truth Resolution Score (TRS)

The primary integrity metric is defined as:

$$TRS = \frac{T_v}{T_{total}}$$

TRS represents the proportion of structurally permissible transitions.

8.4 Weighted Integrity Variant

For safety-critical systems, weighted scoring may be applied:

$$TRS_w = \frac{\sum w_v T_v}{\sum w_v T_v + \sum w_i T_i + \sum w_u T_u}$$

where weights reflect severity classification.

8.5 Uncertainty Ratio

Uncertainty represents unverifiable structural segments:

$$U = \frac{T_u}{T_{total}}$$

Higher uncertainty reduces reliability of conclusions derived from the record.

8.6 Contradiction Density

Let N_c be number of contradiction detections:

$$D_c = \frac{N_c}{T_{total}}$$

Contradiction density measures internal logical inconsistency.

8.7 Continuity Metric

Continuity measures uninterrupted valid operation:

$$C = \frac{\text{longest valid transition sequence}}{T_{total}}$$

Low continuity indicates fragmented structural evolution.

8.8 Impossible State Indicator

$$I_s = \begin{cases} 1 & \text{if impossible state detected} \\ 0 & \text{otherwise} \end{cases}$$

Presence of impossible states invalidates dependent conclusions.

8.9 Metric Output Artifact

The system shall produce a verifiable metric record containing:

- transition counts
- computed scores
- anomaly indicators
- reproducibility metadata

Independent recomputation using identical inputs must produce identical metrics.

9. Mission Capability Classification

9.1 Purpose

Mission Capability Classification converts integrity metrics into operational status categories.

The objective is to provide clear, non-interpretive evaluation of whether a system remains structurally capable of producing permissible outcomes.

9.2 Classification Levels

Based on computed integrity metrics, the system shall assign one of the following:

1. Fully Mission Capable (FMC)
2. Partially Mission Capable (PMC)
3. Non-Mission Capable (NMC)
4. Structurally Impossible (SI)

9.3 Fully Mission Capable (FMC)

A system is classified as Fully Mission Capable if:

- $TRS = 1.0$
- $U = 0$
- $D_c = 0$
- $I_s = 0$

All transitions are structurally valid. No contradictions or uncertainty are present.

9.4 Partially Mission Capable (PMC)

A system is classified as Partially Mission Capable if:

- $TRS < 1.0$ but above defined reliability threshold
- U within acceptable uncertainty bounds
- D_c below contradiction threshold
- $I_s = 0$

Structural reliability is degraded but not invalidated.

9.5 Non-Mission Capable (NMC)

A system is classified as Non-Mission Capable if:

- TRS falls below defined reliability threshold
- D_c exceeds contradiction threshold

- or repeated safety interlock failures occur

Structural continuity is compromised.

9.6 Structurally Impossible (SI)

A system is classified as Structurally Impossible if:

- $I_s = 1$

The claimed state cannot be derived through any sequence of valid transitions.

All dependent conclusions are invalid under this classification.

9.7 Threshold Definition

Threshold values shall be defined prior to evaluation and may vary by domain.

Thresholds must be:

- documented,
- reproducible,
- independent of evaluator identity.

9.8 Operational Consequence

Classification results shall determine whether:

- normal operation may continue,
- corrective review is required,
- transition blocking is triggered,
- systemic audit escalation is required.

10. Accountability Mapping

10.1 Purpose

Accountability Mapping establishes deterministic traceability between a structural defect and its enabling conditions.

The objective is not fault assignment, but reconstruction of the structural pathway that permitted the observed transition.

10.2 Accountability Chain

For each transition, the system shall maintain the following mapping:

Outcome → Transition → Guards → Actor → Enabling Rule → Incentive Vector (if applicable)

Each element must be reconstructable from recorded data.

10.3 Transition Trace Record

For every transition, the system shall record:

- transition identifier
- originating state reference
- candidate next state
- guard evaluation results
- acting entity
- enabling authority reference
- timestamp or dependency position

10.4 Structural Failure Attribution

When a transition is classified as invalid or contributes to Non-Mission Capable status, the system shall identify:

- which guard failed
- which prerequisite event was missing
- which authority invoked the transition
- which enabling rule permitted invocation

This attribution is structural, not moral.

10.5 Enabling Rule Identification

An enabling rule may include:

- statutory authority
- regulatory provision
- procedural policy
- internal directive
- automated system configuration

Rules must be referenced by identifier.

10.6 Incentive Vector (Optional Layer)

Where applicable, the system may record whether the transition was associated with:

- financial compensation triggers
- performance metrics
- quota structures
- automated reward signals
- risk avoidance incentives

Incentive mapping does not imply intent. It documents structural influence pathways.

10.7 Diffusion Index

To evaluate responsibility dispersion, the system may compute:

$$DI = \frac{\text{Number of actors contributing to transition}}{\text{Total dependency nodes in chain}}$$

High diffusion indicates reduced direct accountability clarity.

10.8 Escalation Trigger

If repeated invalid transitions originate from the same enabling rule or actor classification, the system shall flag a systemic accountability concentration.

10.9 Reconstruction Requirement

Accountability mapping must allow independent observers to reconstruct:

- who executed the transition
- under what authority
- under which prerequisites
- with which structural dependencies

without reliance on narrative explanation.

11. Automatic Safety Controls

11.1 Purpose

Automatic Safety Controls define required system responses when structural risk thresholds are exceeded.

The objective is prevention of escalation of structurally invalid states through deterministic protective actions.

Safety responses shall be automatic and not dependent on discretionary interpretation.

11.2 Control Activation Sources

Safety controls may be triggered by:

- safety interlock failure
- impossible state detection
- contradiction density threshold
- uncertainty threshold
- repeated invalid transition pattern
- opacity classification of Opaque or Structurally Non-Resolvable

11.3 Blocking Control

If a safety-critical transition fails a required interlock:

- the transition shall not execute
- the system state shall remain unchanged
- a blocking record shall be generated

Blocking cannot be bypassed silently.

11.4 Protective Hold State

Upon detection of elevated structural risk, the system shall enter Protective Hold:

- no additional irreversible transitions permitted
- only corrective or verification actions allowed
- all new transitions classified provisional

11.5 Automatic Review Trigger

The system shall generate mandatory review when:

- Non-Mission Capable classification occurs
- Impossible State indicator equals 1
- repeated guard failure threshold exceeded

11.6 Escalation Levels

Safety escalation shall be categorized:

1. Advisory Notice
2. Mandatory Review
3. Transition Freeze
4. Systemic Audit

Escalation level depends on severity and recurrence.

11.7 Override Protocol

If override capability exists, the system must:

- require explicit justification
- identify authorizing actor
- create permanent override record
- trigger automatic audit classification

Unlogged override is prohibited.

11.8 Containment Rule

Once a system enters Transition Freeze:

- no irreversible transitions allowed
- only verification and correction permitted
- affected records protected from modification

11.9 Notification Requirement

Safety activation shall generate notification to:

- responsible authority
- oversight entity (if defined)
- verification record artifact

Notification shall not depend on manual reporting.

11.10 Recovery Condition

Normal operation may resume only when:

- structural continuity restored
- impossible state cleared
- integrity metrics exceed reliability threshold

12. Verification Proof Framework

12.1 Purpose

The Verification Proof Framework defines requirements ensuring that structural integrity evaluations are deterministic, reproducible, and independent of evaluator identity.

The framework establishes that verification results arise from the record itself rather than interpretation.

12.2 Deterministic Evaluation

Given an identical event set E, the system shall produce identical:

- transition classifications
- anomaly detections
- integrity metrics
- mission capability classifications

Formally:

$$Evaluate(E) = R \Rightarrow Evaluate(E) = R$$

for any evaluator or execution environment.

12.3 Order Independence

Evaluation shall depend on dependency relations, not arrival order.

For any valid permutation preserving prerequisites:

$$Permute(E) \equiv E$$

the resulting classification set shall remain unchanged.

12.4 Recomputation Stability

Reprocessing the same event record at different times shall produce identical results.

Addition of new events may update indeterminate classifications but shall not silently alter prior valid or invalid determinations without a trace entry.

12.5 Tamper Detection

If an event record is modified, removed, or replaced, recomputation shall produce a different verification result.

$$Modify(E) \Rightarrow Evaluate(E) \neq R$$

This property enables detection of record alteration.

12.6 Independent Verifiability

The verification artifact shall contain sufficient information to allow a third party to recompute results without relying on the original evaluating system.

Required components:

- event dependency graph
- guard evaluation trace
- classification history
- metric computation inputs

12.7 Evaluator Neutrality

Verification results shall not depend on:

- institutional role
- professional expertise
- organizational authority
- narrative interpretation

Only the recorded event structure may influence outcome.

12.8 Consistency Requirement

Multiple independent implementations operating on identical inputs shall produce identical outputs.

12.9 Certification Test Set

Implementations claiming compliance with this standard must pass a reference test suite including:

- valid sequence detection
- missing prerequisite detection

- contradiction detection
- impossible state detection
- reorder invariance
- tamper detection

13. Verification Proof Framework

13.1 Purpose

The Verification Proof Framework defines requirements ensuring that structural integrity evaluations are deterministic, reproducible, and independent of evaluator identity.

The framework establishes that verification results arise from the record itself rather than interpretation.

13.2 Deterministic Evaluation

Given an identical event set E, the system shall produce identical:

- transition classifications
- anomaly detections
- integrity metrics
- mission capability classifications

Formally:

$$Evaluate(E) = R \Rightarrow Evaluate(E) = R$$

for any evaluator or execution environment.

13.3 Order Independence

Evaluation shall depend on dependency relations, not arrival order.

For any valid permutation preserving prerequisites:

$$Permute(E) \equiv E$$

the resulting classification set shall remain unchanged.

13.4 Recomputation Stability

Reprocessing the same event record at different times shall produce identical results.

Addition of new events may update indeterminate classifications but shall not silently alter prior valid or invalid determinations without a trace entry.

13.5 Tamper Detection

If an event record is modified, removed, or replaced, recomputation shall produce a different verification result.

$$Modify(E) \Rightarrow Evaluate(E) \neq R$$

This property enables detection of record alteration.

13.6 Independent Verifiability

The verification artifact shall contain sufficient information to allow a third party to recompute results without relying on the original evaluating system.

Required components:

- event dependency graph
- guard evaluation trace
- classification history
- metric computation inputs

13.7 Evaluator Neutrality

Verification results shall not depend on:

- institutional role
- professional expertise
- organizational authority
- narrative interpretation

Only the recorded event structure may influence outcome.

13.8 Consistency Requirement

Multiple independent implementations operating on identical inputs shall produce identical outputs.

13.9 Certification Test Set

Implementations claiming compliance with this standard must pass a reference test suite including:

- valid sequence detection
- missing prerequisite detection
- contradiction detection
- impossible state detection
- reorder invariance
- tamper detection

14. Appendix: Formal Model

14.1 Event Set

Let $E = \{E_1, E_2, \ldots, E_n\}$ be the set of observed events.

Each event is associated with an actor and optional dependency relation.

14.2 State Function

Let S_0 be the initial state.

A candidate state transition is produced by:

$$S^*_{t+1} = F(S_t, E_t)$$

14.3 Guard Function

Each event is evaluated by guard function:

$$G(S_t, E_t) \in \{\text{satisfied}, \text{unsatisfied}, \text{indeterminate}\}$$

14.4 Transition Classification

$$C_t = \begin{cases} \text{valid} & \text{if } G = \text{satisfied} \\ \text{invalid} & \text{if } G = \text{unsatisfied} \\ \text{unverifiable} & \text{if } G = \text{indeterminate} \end{cases}$$

14.5 State Acceptance

$$S_{t+1} = \begin{cases} S^*_{t+1} & \text{if } C_t = \text{valid} \\ S_t & \text{otherwise} \end{cases}$$

14.6 Dependency Relation

Let $\prec$ represent prerequisite ordering:

$$E_i \prec E_j \Rightarrow E_i \text{ must precede } E_j$$

14.7 Impossible State

$$Impossible(S_t) = \begin{cases} 1 & \text{if no valid path from } S_0 \text{ to } S_t \\ 0 & \text{otherwise} \end{cases}$$

14.8 Transition Counts

$$T_v = |\{C_t = valid\}|$$

$$T_i = |\{C_t = invalid\}|$$

$$T_u = |\{C_t = unverifiable\}|$$

$$T_{total} = T_v + T_i + T_u$$

14.9 Integrity Metrics

$$TRS = \frac{T_v}{T_{total}}$$

$$U = \frac{T_u}{T_{total}}$$

$$D_c = \frac{N_c}{T_{total}}$$

$$C = \frac{\text{longest valid sequence}}{T_{total}}$$

14.10 Validity Condition for Safety-Critical Transition

$$Transition\ Valid = \bigwedge_{k=1}^{n} G_k$$

14.11 Mission Capability Classification

$$FMC : TRS = 1,\ U = 0,\ D_c = 0,\ Impossible = 0$$

$$PMC : TRS \geq T_{reliable},\ Impossible = 0$$

$$NMC : TRS < T_{reliable} \text{ or } D_c > D_{threshold}$$

$$SI : Impossible = 1$$

14.12 Deterministic Evaluation Property

$$Evaluate(E) = R \Rightarrow Evaluate(E) = R$$

14.13 Tamper Detection

$$Modify(E) \Rightarrow Evaluate(E) \neq R$$