Governance Under Degradation (GUD)

Specification v1.0

A Formal Extension to
Intelligence From Architecture (IFA)

Core Specification v1.0

MICHAL HARCEJ

June 2026
Specification v1.0

GOVERNANCE UNDER DEGRADATION

GUD™

Governing Under Degradation in IFA Systems™

A Formal Extension to

Intelligence From Architecture (IFA)

Core Specification v1.0

Michal Harcej

June 2026

Specification v1.0

Normative Status

This document defines normative requirements for GUD compliance. Requirements are stated using MUST, MUST NOT, SHALL, SHALL NOT, SHOULD, and MAY. A system claiming GUD compliance must satisfy all MUST and SHALL requirements. This specification is an extension to IFA Core Specification v1.0 and does not supersede it. Both specifications must be satisfied for full GUD + IFA compliance.

IFA Dependency

This specification presupposes familiarity with and compliance with Intelligence From Architecture (IFA) Core Specification v1.0 by Michal Harcej. All IFA requirements remain binding. GUD adds requirements in the domain of adaptation. Where this specification is silent, IFA governs. IFA Extension Specification: Governance Under Degradation (GUD) v0.1

Governance Under Degradation (GUD)™

Governing Under Degradation in IFA Systems™

A Formal Extension to Intelligence From Architecture™ (IFA™)

AI TRAINING AND MODEL INGESTION RESTRICTION

For licensing inquiries, contact:

Michal Harcej - michal@tauguard.xyz

© 2026 Michal Harcej. All Rights Reserved.

Table of Contents

1. Introduction

1.1 Purpose

The Governance Under Degradation (GUD) extension to the Intelligence From Architecture (IFA) Core Specification defines the normative behavior of a governance architecture when integrity, continuity, authority, infrastructure, evidence, or semantic coherence cannot be fully established.

GUD ensures that governance remains deterministic, auditable, recoverable, and reproducible during failure conditions, preserving constitutional integrity above all other objectives.

---

2. Scope

GUD applies to all IFA-compliant systems where degradation of governance assumptions may occur, including but not limited to:

- Authority ambiguity
- Evidence corruption or delay
- Infrastructure compromise
- Semantic incoherence
- Rule conflicts or governance corruption
- External service unavailability

---

3. Normative Principles

Principle GUD-1: Integrity Precedes Execution

Requirement: No consequence-bearing execution SHALL occur if the system cannot establish sufficient integrity for the requested action.

Definition of Integrity:
Integrity is a composite property encompassing:

- **Authority Integrity: Validity of the authority hierarchy and delegation chains.**
- **Evidence Integrity: Admissibility, authenticity, and continuity of evidence.**
- **Continuity Integrity: Unbroken chain of governance decisions and state transitions.**
- **Semantic Integrity: Stability and coherence of governed concepts and rules.**
- **Governance Integrity: Consistency and non-corruption of governance rules and structures.**
- **Infrastructure Integrity: Trustworthiness of the underlying hardware, firmware, storage, time, and execution environment.**

Compliance:

- The system MUST evaluate integrity prior to execution.
- If integrity cannot be established, the system MUST default to a BLOCK or RESTORE state (see Section 5).

Principle GUD-2: Degradation Is a Governed State

Requirement: Degradation SHALL NOT be treated as an exception.

Specifications:

- Degradation MUST be explicitly classified as a governance state.
- The architecture MUST define and manage all possible degraded conditions.
- **Undefined degradation states are prohibited.**

Rationale:
Degradation is not a failure of governance but a mode of governance. The system must transition deterministically between governance states based on integrity assessments.

Principle GUD-3: Deterministic Crisis Behavior

Requirement: The architecture MUST produce deterministic outcomes under degradation.

Permitted Outcomes:

Outcome	Description
ALLOW	Execution proceeds with full integrity.
REVIEW	Execution is paused; human or automated review is required.
BLOCK	Execution is prohibited; no consequence-bearing action occurs.
RESTORE	System enters recovery mode to re-establish a known integral state.

Prohibition:

- Probabilistic, non-deterministic, or heuristic-based crisis handling IS PROHIBITED.

Principle GUD-4: Constitutional Preservation

Requirement: The primary objective during degradation MUST be the preservation of constitutional integrity.

Hierarchy of Priorities:

1. **Governance Integrity (highest priority)**
2. **System Availability (subordinate to governance integrity)**

Compliance:

- The architecture MUST prefer refusal over inadmissible execution.
- **Inadmissible execution is defined as any action that violates constitutional invariants or cannot be audited.**

Principle GUD-5: Recovery to Known Integral State

Requirement: The architecture MUST maintain one or more attested recovery states.

Specifications:

- When integrity cannot be established, the system MUST restore to the most recent verified integral state.
- All recovery actions MUST be auditable and logged in the Canonical Knowledge Graph (CKG).
- The restored state MUST be re-attested before execution resumes.

Principle GUD-6: Reproducibility of Governance States

Requirement: The architecture MUST ensure that all governance states are reproducible for audit, regulatory, and investigative purposes.

Specifications:

- Historical governance decisions MUST be reproducible using the rule version, authority version, knowledge state version, and timestamp at the time of the decision.
- **Rule history immutability is mandatory (see Section 7.3).**

---

4. Governance States

The system SHALL operate in one of the following six governance states, ordered by severity:

G0 — Integral

Definition: All constitutional invariants are satisfied.

Characteristics:

- Authority is valid.
- Evidence is valid and continuous.
- Continuity is unbroken.
- Infrastructure is attested.
- Semantic coherence is maintained.

Outcome:

- **ALLOW: Execution proceeds.**

G1 — Warning

Definition: Minor degradation detected; constitutional invariants remain satisfied.

Examples:

- Priority collision between rules.
- Evidence delay (non-critical).
- Non-critical semantic drift.

Outcome:

- **ALLOW WITH REMEDIATION: Execution proceeds, but remediation actions (e.g., logging, alerts) MUST be triggered.**

G2 — Degraded

Definition: Integrity is partially impaired; constitutional invariants may be at risk.

Examples:

- Authority ambiguity (e.g., conflicting delegation).
- Broken continuity (e.g., missing decision trace).
- Infrastructure anomaly (e.g., untrusted execution environment).
- Rule conflict (e.g., contradictory governance rules).

Outcome:

- **REVIEW REQUIRED: Autonomous execution IS PROHIBITED.**
    - **The system MUST escalate to a human or automated reviewer.**
    - The system MUST log the degradation event in the CKG.

G3 — Failure

Definition: Admissibility of the requested action cannot be established.

Examples:

- Governance corruption (e.g., tampered rules).
- Evidence is inadmissible (e.g., forged or incomplete).
- Semantic incoherence (e.g., governed concepts are unstable).
- Unauthorized learning event (e.g., unapproved model update).

Outcome:

- **BLOCK: Execution IS PROHIBITED.**
    - **The system MUST log the failure event in the CKG.**
    - The system MUST notify all relevant authorities.

G4 — Constitutional Crisis

Definition: The constitutional basis of governance cannot be trusted.

Examples:

- CKG corruption.
- Authority hierarchy corruption.
- Recovery-chain corruption.
- Infrastructure attestation failure.

Outcome:

- **RESTORE: The system MUST enter constitutional recovery mode.**
    - **All non-essential execution MUST halt.**
    - **The system MUST restore to the last attested integral state.**
    - The system MUST transition to G5 (Recovery Verification).

G5 — Recovery Verification

Definition: A restored state awaiting re-attestation.

Examples:

- CKG restored.
- Backup restored.
- Rule set restored.
- Authority hierarchy restored.

Outcome:

- **REVIEW: Execution remains restricted until re-attestation succeeds.**

Requirements:

- The system MUST verify:
    - Self-attestation success.
    - Constitutional invariants.
- Only upon successful verification:
    - The system MAY transition to G0 (Integral).

---

5. Infrastructure Integrity

5.1 Definition

Infrastructure Integrity is the trustworthiness of the physical and logical components required for governance, including:

- Hardware integrity (e.g., tamper-proof execution).
- Firmware integrity (e.g., verified bootloaders).
- Storage integrity (e.g., immutable logs).
- Time integrity (e.g., synchronized and attested clocks).
- Execution environment integrity (e.g., isolated, non-compromised runtime).

5.2 Requirements

- The architecture MUST be capable of assessing infrastructure integrity in real time.
- **Failure of infrastructure integrity MAY invalidate governance decisions.**
    - If infrastructure integrity cannot be established, the system MUST transition to G3 (Failure) or G4 (Constitutional Crisis).

5.3 Governance Admissibility

Definition: Governance Admissibility (GA) is the composite measure of constitutional integrity, calculated as:

```
text
Copy

GA = f(
    Authority Integrity,
    Evidence Integrity,
    Continuity Integrity,
    Governance Integrity,
    Semantic Integrity,
    Infrastructure Integrity
)
```

Requirements:

- Governance Admissibility MUST be calculated continuously.
- Governance State transitions MUST be based on Governance Admissibility.
- Governance Admissibility MUST be reproducible and auditable.

5.4 Infrastructure Attestation Inputs

The architecture SHOULD evaluate the following inputs to determine infrastructure integrity:

- **CPU Health (e.g., utilization, temperature)**
- **Memory Health (e.g., usage, errors)**
- **Storage Health (e.g., availability, corruption)**
- **Network Health (e.g., connectivity, latency)**
- **Time Synchronization (e.g., NTP status)**
- **Backup Availability (e.g., accessibility, integrity)**
- **Chronicle Integrity (e.g., log immutability)**
- **CKG Integrity (e.g., graph consistency)**

Example Attestation Input:

```
json
Copy

{
  "cpu_health": 26.9,
  "memory_health": 45.0,
  "storage_health": 50.9,
  "chain_integrity": true
}
```

These inputs MAY affect governance state transitions.

5.5 Governance Under Isolation

Definition:
A governance state where external services are unavailable.

Examples:

- Internet loss.
- Cloud provider loss.
- Model provider loss.
- API dependency failure.

Requirements:

- Governance MUST continue using local evidence.
- Governance MUST continue using local rules.
- Governance MUST continue using local authority structures.
- Loss of external intelligence SHALL NOT invalidate governance.

---

6. Semantic Integrity

6.1 Definition

Semantic Integrity is the stability and coherence of the governed concepts, rules, and knowledge within the system.

6.2 Requirements

- The architecture MUST monitor semantic coherence of all governed concepts.
- The system MUST determine:
    - Whether meaning remains stable.
    - Whether learning events are admissible.
    - Whether concept evolution is authorized.
- **Semantic incoherence SHALL be treated as a governance event.**
    - **The system MUST log semantic anomalies in the CKG.**
    - The system MUST transition to G2 (Degraded) or higher if semantic integrity is compromised.

---

7. Governance Self-Attestation

7.1 Definition

Governance Self-Attestation is the continuous evaluation of the admissibility of the system’s own governance state.

7.2 Requirements

- The architecture MUST NOT assume its own validity.
- Governance validity MUST be demonstrable through evidence.
    - The system MUST provide auditable proofs of its governance state.
    - The system MUST transition to G2 (Degraded) or higher if self-attestation fails.

7.3 Governance History Integrity

Requirements:

- Governance rules MUST be versioned.
- Rule history MUST be immutable.
- Historical decisions MUST remain reproducible.
- Previous rule versions MUST NOT be altered.

Decision Metadata:
Each governance decision MUST reference:

- Rule Version
- Authority Version
- Knowledge State Version
- Timestamp

Rationale:
Without immutability, governance history can be altered, rendering the system non-auditable and non-reproducible.

---

8. Constitutional Recovery

8.1 Definition

Constitutional Recovery is the process of restoring the system to a known integral state after a degradation event.

8.2 Requirements

- The system MUST restore:
    - Authority hierarchy.
    - Canonical knowledge state (CKG).
    - Evidence continuity.
    - Governance rules.
    - Semantic state.
- The restored state MUST be re-attested before execution resumes.
- All recovery actions MUST be logged in the CKG.

---

9. Core Doctrine

1. A governance architecture is not defined by how it behaves when all assumptions hold.
    - It is defined by how it behaves when assumptions fail.
2. **Integrity is not a static property.**
    - Integrity is a continuously governed state.
3. **When integrity is lost, consequence shall stop.**
    - No consequence-bearing action SHALL occur in a state of unknown or compromised integrity.
4. **When integrity is restored, governance may resume.**
    - Resumption of execution MUST be preceded by re-attestation of the restored state.
5. **Reproducibility is the foundation of auditability.**
    - A system that cannot reproduce its governance state at decision time is not fully auditable.

---

10. Compliance and Auditability

10.1 Compliance Requirements

- All transitions between governance states MUST be:
  - **Deterministic (same inputs → same outputs).**
  - **Auditable (logged in the CKG with timestamps, inputs, and outcomes).**
  - **Reversible (restorable to a previous integral state).**
  - **Reproducible (historical states must be reconstructable).**

10.2 Audit Requirements

- The system MUST maintain:
  - A complete decision trace for all governance state transitions.
  - **Immutable logs of all degradation events, recovery actions, and attestations.**
  - **Immutable rule history for reproducibility.**
- Audits MUST be performable by third parties using only the CKG, system logs, and rule history.

---

11. References

- **IFA Core Specification v1.0 (Sections 1–11, especially Sections 2, 4, 6–8 on executable governance, structural invariants, and the CKG).**
- **Canonical Knowledge Graph (CKG) as defined in IFA Core Specification, Section 8.**
- **TauDIL KPI Monitoring Framework (for infrastructure attestation inputs).**

---

12. Revision History

Version	Date	Author	Changes
0.1	2026-06-14	Michal Harcej	Formalized principles, states, and requirements.
1.0	2026-06-14	Michal Harcej	Added Governance Admissibility, G5 (Recovery Verification), Rule History Immutability, Hardware Attestation, and Governance Under Isolation.