WORKING PAPER
EVIDE Governance Lab — May 2026

Recursive Semantic Governance:
Preserving Accountability Across AI Boundary Transformations

From Static Audit Logs to Recursive Evidentiary Governance: Semantic Custody, Continuity Propagation, and Layered Accountability in AI Systems

---

Emanuel Celano
Informatica in Azienda — EVIDE Governance Lab — app.certifywebcontent.com

---

"Governance is not event recording.
Governance is transformation qualification."
— RSG Canonical Principles

"Capability expansion must not silently become authority expansion."

Version 1.1 — May 27, 2026

Abstract

Current AI governance architectures treat accountability as a collection of discrete, static artifacts: audit logs, explainability outputs, post-hoc event records, and snapshot-based compliance states. This model fails at the boundary -- the moment when a decision crosses from one system, agent, or governance layer to another.

This working paper proposes Recursive Semantic Governance (RSG), a theoretical framework that treats governance not as documentation but as the progressive propagation and stabilization of semantic state across accountability boundaries. RSG introduces five core primitives:

- Semantic Custody -- measurable preservation of governance-relevant meaning across boundaries
- Governance Vectors -- structured multi-dimensional representations of accountability state
- Boundary-Trained Connectors -- controlled semantic translation mechanisms between layers
- Recursive Boundary Alignment -- iterative stabilization cycles at every crossing
- Recursive Evidentiary Governance -- externally anchored, independently verifiable governance chronology

We introduce formal notation for governance vector semantics, semantic divergence measurement, and causal persistence scoring. We present three canonical architectural diagrams and a full end-to-end walkthrough scenario. We also catalog eight distinct governance failure modes that existing architectures cannot detect.

The framework is grounded in the EVIDE evidentiary boundary layer (app.certifywebcontent.com) as an operational reference implementation, validated through the first externally anchored cross-layer composition test conducted on May 27, 2026.

RSG Canonical Principles: 1. Governance is not event recording. Governance is transformation qualification. 2. Capability expansion must not silently become authority expansion.

Keywords: AI governance, boundary engineering, recursive governance, evidentiary computing, accountability boundaries, governance vectors, transformation qualification, failure modes

Glossary of Core Terms

The following definitions establish the precise meaning of terms used throughout this paper. These are operational definitions -- bounded to their governance context and not intended as general linguistic or philosophical claims.

TERM	OPERATIONAL DEFINITION
Accountability Boundary	A crossing point between two governance layers at which decision authority, semantic state, and responsibility attribution must be explicitly qualified and transmitted.
Accountability Survivability	The property whereby responsibility for a decision remains attributable, verifiable, and semantically coherent across all boundary crossings from decision formation to consequence propagation.
Boundary Connector	A system component that performs controlled semantic translation between adjacent governance layers, enforcing lawful transformation without introducing governance claims not present upstream.
Causal Persistence (Cp)	An observational signal that qualifies whether a subsequent governance state preserves sufficient causal inertia from its prior degradation path. Values: present, attenuated, absent, inconclusive.
Crystallization	The transition from an active stabilization process to a confirmed, externally anchored governance state. Crossing-sufficient -- not epistemically true.
Evidentiary Chronology	A timestamped, independently reconstructable record of the governance stabilization process, generated during boundary crossing rather than after it.
Governance Vector $G^n(t)$	A structured multi-dimensional representation of accountability state at layer n, crossing time t. Components: Decision, Authority, Intervention, Threshold, Continuity, Evidentiary.
Governance-Relevant Semantics	The set of observable accountability properties that must survive a boundary crossing for responsibility to remain attributable. Excludes inferred intent, subjective meaning, and NLP-style semantic content.
Recursive Boundary Alignment	Iterative observation-refinement cycles applied at each boundary crossing to progressively stabilize governance state before confirmation and transmission.
Semantic Custody	The measurable preservation of governance-relevant meaning across accountability boundaries. Requires: emission of a structured governance vector, transmission with structure preserved, and reception with degradation detection capacity.
Semantic Divergence (Δs)	A measurable distance between the governance-relevant properties of a state emitted by one layer and received by another. When $\Delta s > \varepsilon$, semantic custody failure is declared.
Split Confirmation	A mechanism that partitions governance vector components into confirmed, degraded, non-transferable, and unverifiable signals at crossing time, preventing binary crossing decisions.

Synthetic Coherence	A failure mode in which a governance state appears stable without having paid the expected dynamic cost of recovery from prior degradation. Detected when Cp < 0.40 and stability trend = improving.
Transformation Qualification	The act of determining whether a state transition between governance layers is lawful -- preserving accountability-relevant semantics without inflation, hallucination, or silent authority expansion.

1. Introduction

1.1 The Failure of Static Governance

The dominant paradigm in AI governance today is snapshot-based. Systems emit logs. Governance engines capture events. Compliance frameworks request reports. This model assumes that accountability is recoverable -- that given sufficient data, the chain of responsibility can always be reconstructed after the fact. This assumption is increasingly untenable.

When a decision crosses an accountability boundary, the semantic content of that decision -- its authority, intervention conditions, threshold applicability, continuity constraints -- is routinely stripped, translated without preservation guarantees, or lost entirely. Logs exist. Events are recorded. Accountability collapses anyway.

1.2 The Accountability Survivability Problem

We introduce Accountability Survivability as the central challenge this paper addresses:

> *Accountability survivability is the property whereby responsibility for a decision remains attributable, verifiable, and semantically coherent across all boundary crossings from decision formation to consequence propagation.*

Auditability asks: can we reconstruct what happened? Survivability asks: does responsibility remain intact while it is happening? In agentic AI systems, the window between decision emission and consequence propagation is often too short for post-hoc reconstruction to be effective.

1.3 Motivation: Recursive State Propagation in AI Systems

Recursive multi-agent system architectures (cf. RecursiveMAS, 2026) demonstrate that agents can pass not only outputs but compressed semantic state representations between processing layers -- carrying context, trajectory, priority,

and coherence signals beyond discrete output tokens. The governance implication is direct: a governance layer that receives only a final decision token, stripped of the semantic state that produced it, cannot perform meaningful accountability qualification.

This paper proposes that governance layers should emit and receive structured semantic representations -- governance vectors -- that preserve accountability-relevant meaning across boundaries.

1b. Why Current Audit Logs Fail: A Mathematical Argument

Before introducing RSG, we establish formally why the traditional audit log model is structurally insufficient. This is not a philosophical critique -- it is a mathematical one.

The Audit Log Model

Traditional governance architectures model accountability as the accumulation of discrete events:

$$A = \Sigma(e_i) \quad \text{where } e_i = \{\text{type, timestamp, payload}\}$$

Accountability = sum of discrete events. Each event is independent, point-in-time, and carries no semantic relationship to adjacent events.

This model has one critical mathematical property: it is additive over time, but not transformational. Events accumulate. They do not evolve. The semantic state that produced event e_i is not recoverable from e_i alone.

The RSG State Transformation Model

RSG replaces discrete event accumulation with state transformation qualification:

$$G(t_{n+1}) = T(G(t_n), \Delta s, C_p)$$

The governance state at the next crossing is a function of: current state $G(t_n)$, semantic divergence Δs, and causal persistence C_p

This has a fundamentally different mathematical property: it is transformational over boundary crossings. The state at each crossing depends on the trajectory that produced it, not merely the current snapshot. Semantic loss is measurable. Drift is computable. Recovery authenticity is verifiable.

The practical consequence is direct: in the audit log model, governance failure at a boundary crossing leaves no recoverable trace unless explicitly logged. In the RSG model, any deviation from expected state transformation is mathematically detectable through divergence Δs exceeding threshold ε.

```
Traditional: failure undetected until post-hoc reconstruction

RSG: failure detectable as Δs > ε at crossing time
```

RSG detects governance degradation at the boundary, not after the consequence has propagated

---

2. Architectural Overview

Before introducing the formal framework, we present three canonical architectural diagrams that orient the RSG model visually. These diagrams establish the spatial and sequential structure of the framework.

Figure 1. Static Governance vs Recursive Semantic Governance

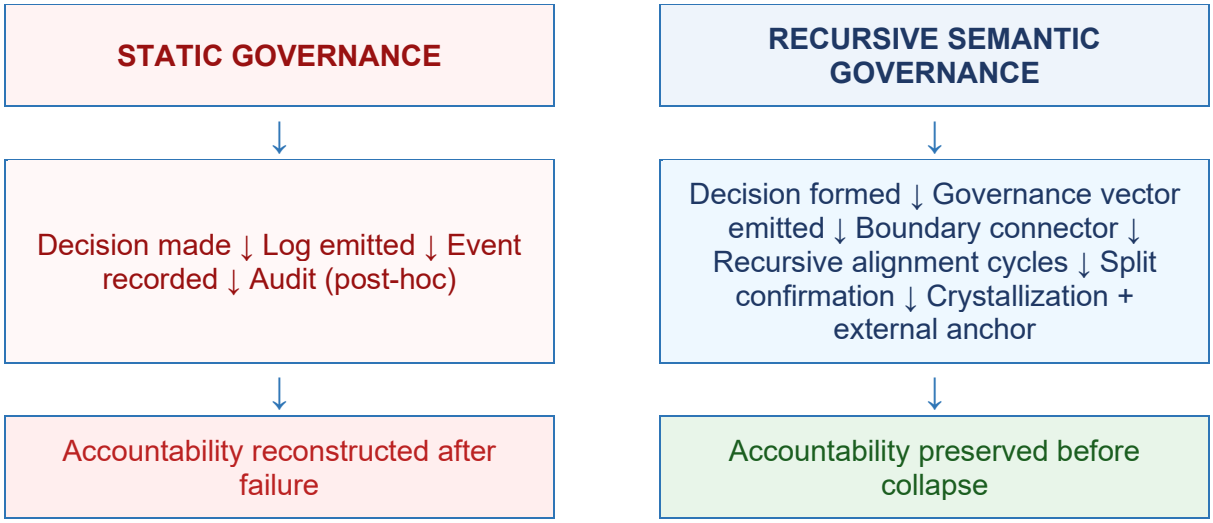
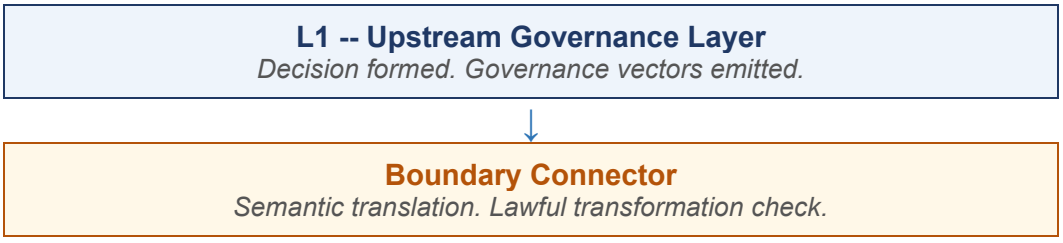


Figure 1 -- In static governance (left), accountability is reconstructed post-hoc from discrete events. In RSG (right), semantic state is actively propagated and stabilized at every boundary crossing.

Figure 2. Boundary Crossing Lifecycle



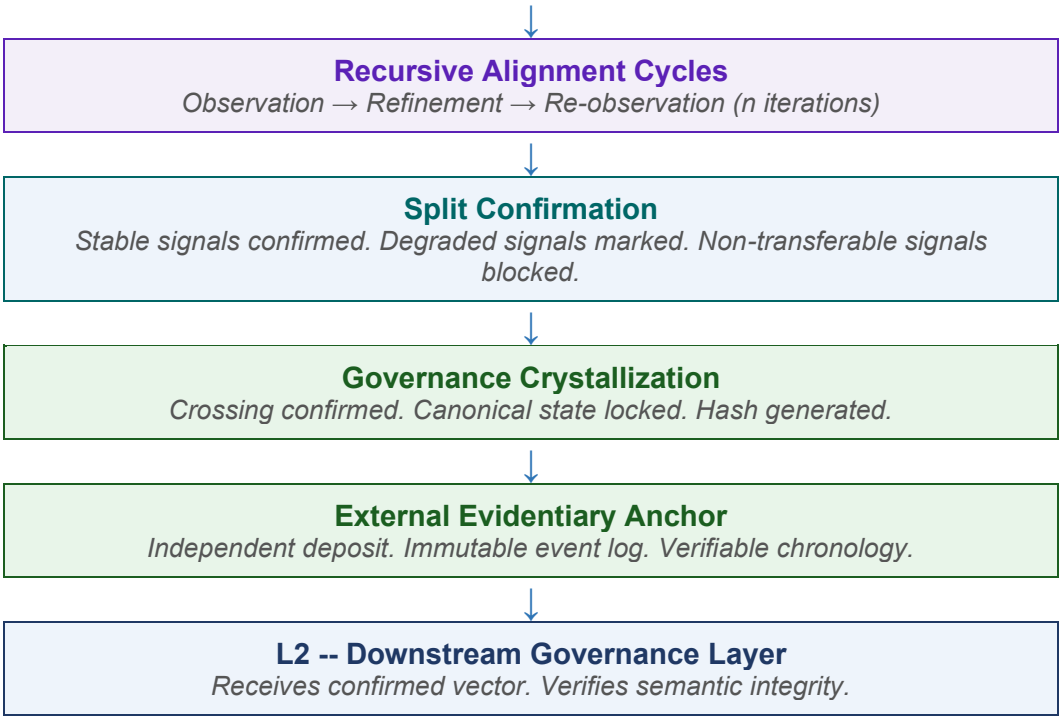


Figure 2 -- The boundary crossing lifecycle replaces single-pass event emission with a governed stabilization sequence. Each stage is independently auditable.

Figure 3. Governance Vector Propagation at Boundary Crossing

VECTOR TYPE	STATE AT CROSSING	STATUS
Decision Vector	`classification: stable threshold: defined`	✓ CONFIRMED
Authority Vector	`attribution: attributed conflict: none`	✓ CONFIRMED
Intervention Vector	`modification: minor stability: coherent`	✓ CONFIRMED
Continuity Vector	`trend: improving causal: attenuated`	⚠ DEGRADED
Threshold Vector	`attribution: pending resolution: open`	⚠ DEGRADED
Evidentiary Vector	`visibility: partial unresolved: [sig_01]`	□ PARTIAL

Figure 3 -- Governance vectors carry different confirmation states across a boundary. Confirmed vectors are transmitted fully. Degraded vectors are transmitted with explicit markers. Blocked vectors (not shown) are rejected with boundary rejection records.

3. Formal Notation and Primitives

We introduce minimal formal notation sufficient to ground the framework computationally. The notation is deliberately lightweight -- not a full formal calculus, but sufficient to express the core governance relationships precisely.

Mathematical Regime Declaration

RSG operates within a specific formal paradigm that readers should hold in mind throughout this section:

- Operational semantics: governance properties are defined by their observable behavior at boundary crossings, not by their internal representation. A governance vector is valid if it produces the expected accountability behavior under the defined operations.
- Discrete dynamic systems: governance state evolves through discrete boundary crossing events, not continuous time. The transition function T qualifies state changes at each crossing.
- Observability metrics: the measurability requirements for Δs and C_p are grounded in observability theory -- what the governance layer can legitimately observe, not what is internally true about the system being governed.

This regime is distinct from formal logic (which would require complete axiomatization), probability theory (which would require distributional assumptions), and linear algebra (which would impose geometric constraints inappropriate for governance semantics). Appendix A provides sketch definitions for readers requiring more formal grounding.

3.1 Governance Vector Notation

Let L_n denote a governance layer. The governance vector emitted by L_n at boundary crossing time t is:

$$G^n(t) = \langle D^n, A^n, I^n, T^n, C^n, E^n \rangle$$

where D = Decision, A = Authority, I = Intervention, T = Threshold, C = Continuity, E = Evidentiary

Each component is a structured object, not a scalar. The vector captures the full accountability-relevant state at crossing time.

3.2 Semantic Divergence

The semantic divergence between adjacent governance layers L_n and L_{n+1} is:

$$\Delta s(L_n, L_{n+1}) = d(G^n(t\_emit), G^{n}_{+1}(t\_receive))$$

where d is a governance-semantics distance function, t_emit is emission time, t_receive is reception time

When Δs exceeds a threshold ε, semantic custody failure is declared and the boundary crossing is rejected or marked with degradation markers.

$$\Delta s > \varepsilon \quad \Rightarrow \quad \text{SEMANTIC CUSTODY FAILURE}$$

The threshold ε is layer-pair-specific and reflects the maximum tolerable semantic loss for that crossing

3.3 Causal Persistence Score

The causal persistence signal C_p is computed as a function of three observable dimensions:

$$C_p = f(S_t, D_s, A_t)$$

S_t = stability trend | D_s = drift state | A_t = alignment coherence

The four canonical causal persistence states map to scoring ranges:

- present: $C_p \geq 0.75$ -- causal inertia sufficiently preserved
- attenuated: $0.40 \leq C_p < 0.75$ -- causal inertia partially preserved
- absent: $C_p < 0.40$ -- causal inertia insufficient; possible synthetic coherence
- inconclusive: observability surface insufficient to compute C_p

The absent state is the most governance-significant. A governance state that exhibits absence of causal persistence after apparent recovery is a candidate for synthetic coherence -- a state that appears stable without having paid the expected dynamic cost of recovery.

$$C_p < 0.40 \wedge \text{trend} = \text{improving} \quad \Rightarrow \quad \text{SYNTHETIC COHERENCE CANDIDATE}$$

This condition triggers escalation review regardless of surface stability indicators

3.4 Governance Stabilization Score

The stabilization score S at buffer close time represents sufficiency for boundary crossing, not truth:

$$S(t\_close) = w_1 \cdot C_p + w_2 \cdot trend\_score + w_3 \cdot (1 - unresolved\_ratio)$$

Weights w_1, w_2, w_3 are calibrated empirically. $S \geq \theta$ required for stable verdict

Critically: $S \geq \theta$ qualifies the vector for crossing -- it does not certify the underlying decision as correct. Conflating the two produces governance inflation.

---

4. Governance as Semantic State Propagation

4.1 Semantic Custody

Definition: Governance-Relevant Semantics

Throughout this paper, ‘semantic’ refers specifically to governance-relevant semantics -- a constrained, operational definition distinct from linguistic or distributional semantics in NLP:

- Governance-relevant semantics is the set of accountability properties that must survive a boundary crossing for responsibility to remain attributable and verifiable.
- These properties are observable, not interpreted: they include authority attribution status, threshold satisfaction state, continuity signal values, and unresolved signal lists -- not inferred intent, subjective meaning, or natural language interpretation.
- Semantic divergence Δs measures the distance between these observable accountability properties across a boundary crossing -- not linguistic similarity or embedding distance.

This definition is intentionally narrow and operationalizable. RSG does not deal in subjective meaning. It deals in bounded, measurable accountability state.

4.2 Semantic Custody

Semantic custody -- the preservation of governance-relevant meaning across accountability boundaries -- has three necessary conditions:

- Emission: the sending layer produces a structured governance vector at crossing time
- Transmission: the vector is transmitted with structure preserved and verification possible
- Reception: the receiving layer can interpret the vector and detect semantic degradation

The absence of any condition constitutes semantic custody failure. Semantic custody is not achieved by logging -- it is achieved by transmitting structured semantic representations.

4.2 Lawful Semantic Transformation

Not all semantic transformations between governance layers are lawful. A lawful transformation satisfies four constraints:

- Preservation: accountability-relevant semantics in the upstream representation remain recoverable downstream
- Attribution: the transformation is attributable to an identifiable connector component
- Verifiability: the transformation produces artifacts allowing independent verification
- Non-expansion: the transformation introduces no governance claims not present upstream

Non-expansion is critical. A boundary connector that adds authority claims, widens threshold applicability, or strengthens continuity assertions beyond what the

upstream state supports is performing governance inflation -- a failure mode discussed in Section 7.

---

5. Governance Vectors

Each component of the governance vector $G^n(t) = \langle D^n, A^n, I^n, T^n, C^n, E^n \rangle$ encodes a distinct accountability dimension.

5.1 Decision and Authority Vectors

Decision vectors encode classification status, threshold conditions, and closure state. Authority vectors encode who holds legitimate responsibility, attribution confidence, delegation depth, and conflict signals. Together they answer: what was decided, and who owns it.

5.2 Intervention and Threshold Vectors

Intervention vectors encode how the decision was modified before crossing, including modification intensity and intervention stability. Threshold vectors encode what normative constraints were applied and whether those constraints are attributable.

5.3 Continuity Vector

The continuity vector is the most governance-significant component. It encodes:

- Stability trend (improving, degrading, oscillating, static)
- Causal persistence signal C_p with computed score
- Semantic drift measurement Δs against prior crossing
- Continuity state classification (coherent, partially coherent, fragmented, unverifiable)

> *An authentic degradation leaves inertia. If inertia disappears too rapidly, coherence may have been synthetically reconstructed rather than genuinely maintained.*

5.4 Evidentiary Vector

The evidentiary vector encodes verifiability properties: canonical hash of the governance state, independence declaration, observability surface (declared_complete, partial, insufficient), and unresolved signals at crossing time.

5.5 Machine-Shape Considerations

Governance vectors as defined here are structured objects suitable for direct JSON serialization. Future extensions may address:

- Embedding compatibility: governance vectors could be projected into continuous embedding spaces, enabling similarity-based semantic divergence computation rather than field-by-field comparison.
- Graph-structured representations: for multi-agent governance chains, vectors could be expressed as directed graphs where nodes represent governance layers and edges carry semantic custody records.
- Recursive composability: vectors from multiple upstream layers could be composed into aggregate governance representations using defined composition operators.
- Partial-state propagation: not all vector components need to be fully populated at every crossing. Sparse vector representations allow partial-state propagation with explicit incompleteness markers.

These extensions are not required for RSG v1 and are noted here to establish architectural compatibility with recursive semantic propagation systems.

---

6. Recursive Boundary Alignment

6.1 Beyond Verification: Recursive Semantic Stabilization

Recursive Boundary Alignment is not merely verification -- it is recursive semantic stabilization. The distinction is fundamental.

Verification asks: does the governance state meet crossing conditions? Stabilization asks: how does the governance state transform through successive observation cycles to approach crossing conditions? The recursive process does not simply check a state -- it actively shapes it toward stability.

This makes RSG architecturally analogous to iterative refinement in recursive reasoning systems: each cycle receives feedback from the previous observation and adjusts the emitted governance state accordingly. The number of cycles, the trajectory of adjustments, and the convergence path are all part of the governance record.

6.2 Recursive Verification Cycles

Formally, the stabilization process for a boundary crossing from L_n to L_{n+1} proceeds as:

1. Emission: L_n emits $G^n(t_0)$
2. Observation: connector observes $G^n(t_0)$, computes divergence Δs, detects anomalies

3. Refinement: L_n refines vector based on observation feedback, emits $G^n(t_1)$
4. Re-observation: connector re-evaluates. If $\Delta s \leq \varepsilon$, proceed to confirmation. Else, repeat.
5. Confirmation: crossing conditions met. Split confirmation applied.
6. Transfer: confirmed vector transmitted. Evidentiary record generated.

6.3 Split Confirmation Propagation

Under split confirmation, governance vectors are partitioned at crossing time into four categories:

- Stable signals (✓): confirmed and transmitted with full evidentiary weight
- Degraded signals (⚠): transmitted with explicit degradation markers and reduced epistemic weight
- Non-transferable signals (×): blocked; boundary rejection record generated
- Unverifiable signals (□): transmitted with explicit uncertainty declarations

This prevents binary crossing decisions and allows partial transmissions with explicit qualification. A governance layer that blocks a non-transferable signal is performing correct governance, not failure -- provided the block is recorded.

Critical clarification on blocked signals: a blocked signal (×) does not mean the system silently proceeds without that governance component. A blocked signal generates a boundary_rejection_record that must interrupt the execution chain at the Execution Boundary. The downstream layer cannot act on a partially confirmed vector as if it were complete -- the block must either trigger human review, halt progression to a safe state, or explicitly escalate. A system that proceeds past a blocked signal without declaring the gap is producing a false crystallization failure mode (Section 8.7), not a correct partial transmission.

6.4 Governance Stabilization Dynamics

Three dimensions characterize stabilization dynamics:

- Stability trend: is the state becoming more or less stable over iterations? (improving, degrading, oscillating, static)
- Convergence type: does stabilization result from semantic resolution or timeout expiration? A timeout-stabilized state carries fundamentally weaker evidentiary weight than a convergence-stabilized state.
- Causal persistence: does the state at confirmation preserve sufficient causal inertia from prior states?

---

7. Recursive Evidentiary Governance

7.1 The Evidentiary Chronology

Each recursive boundary alignment cycle generates verifiable artifacts. Together, these constitute an evidentiary chronology -- a timestamped, independently reconstructable record of the governance stabilization process.

An evidentiary chronology differs from a standard audit log in four critical respects: it is generated during the crossing (not after), it captures the trajectory of stabilization (not only the final state), it includes explicit degradation and unresolved signal records, and it is anchored externally and independently verifiable.

7.2 Governance Crystallization

> *Governance crystallization is the moment at which a governance vector achieves sufficient stabilization to be transmitted across a boundary, anchored externally, and treated as the authoritative record of the crossing state.*

Crystallization is not truth certification. A crystallized governance state is crossing-sufficient, not necessarily correct. This distinction is mechanically enforced in the EVIDE ESB through the semantic precision: buffer_verdict = stable means sufficiently stabilized for crossing -- not absolute epistemic truth.

7.3 Non-Claims Discipline

Every governance layer must maintain explicit non-claims declarations. The minimal non-claims set includes:

- Does not determine truth or correctness -- qualifies observable governance properties only
- Does not certify decision correctness -- a stable governance record does not imply a correct decision
- Does not interpret intent -- encodes observable state, not inferred purpose
- Does not determine legal admissibility -- admissibility is a function of legal systems
- Does not grant execution permission -- stabilization is not authorization

---

7b. Boundary Taxonomy

RSG operates across multiple boundary types. Different boundary types have different semantic preservation requirements, crossing qualification criteria, and failure mode profiles. We introduce a preliminary boundary ontology:

- Semantic boundary: the crossing at which governance-relevant meaning transitions between representation formats or vocabularies. Primary risk: semantic divergence Δs through unlawful transformation.
- Authority boundary: the crossing at which decision responsibility transitions between attributable agents. Primary risk: authority hallucination -- responsibility inferred rather than transmitted.
- Evidentiary boundary: the crossing at which governance state is anchored externally. Primary risk: false crystallization -- anchoring before crossing conditions are met.
- Execution boundary: the crossing at which a governance-qualified decision triggers consequence propagation. Primary risk: execution before stabilization -- action before governance closure.
- Continuity boundary: the crossing at which continuity of accountability state must be verified across a temporal gap or system discontinuity. Primary risk: continuity collapse -- undetected state fragmentation.
- Admissibility boundary: the crossing at which governance records enter a legal or compliance framework. Primary risk: governance inflation -- evidentiary records interpreted as stronger than their qualified scope.

Each boundary type requires a specialized governance vector configuration. A complete RSG implementation specifies crossing qualification criteria independently for each boundary type rather than applying uniform criteria to all crossings.

> *Not all boundaries are equivalent. A semantic boundary failure is recoverable through re-transmission. An execution boundary failure is not -- the consequence has propagated.*

---

8. Governance Failure Modes

A major contribution of RSG is the identification and formal characterization of governance failure modes that existing architectures cannot detect. We catalog eight primary failure modes.

8.1 Semantic Inflation

Definition: a governance layer makes stronger accountability claims than the upstream state supports. The receiving layer believes it has received a fully authorized, verified governance state when the upstream layer only emitted a provisional one.

Detection: compare upstream $G^n(t\_emit)$ authority confidence with downstream $G^{n+1}(t\_receive)$ authority confidence. Any positive delta is semantic inflation.

8.2 Authority Hallucination

Definition: a governance layer attributes decision authority to an entity that did not exercise it. This typically occurs when boundary connectors infer authority from context rather than receiving explicit authority vectors.

Detection: authority vectors must carry explicit attribution provenance, not inferred attribution.

8.3 Synthetic Coherence

Definition: a governance state appears stable and coherent without having undergone the expected dynamic cost of recovery from prior degradation. The continuity vector shows improving stability trend while causal persistence score is absent or inconclusive.

Detection: $C_p < 0.40 \wedge$ trend = improving triggers synthetic coherence alert.

8.4 Stabilization by Timeout

Definition: a buffer closure occurs because the observation window expired, not because the governance state achieved genuine semantic convergence. The crystallized state inherits the full evidentiary weight of convergence-based closure without warranting it.

Detection: closure_trigger = timeout must carry explicit epistemic weight reduction in evidentiary records.

8.5 Continuity Collapse

Definition: the continuity vector degrades to fragmented or unverifiable state but the governance record does not reflect this because continuity signals are not transmitted across boundaries.

Detection: continuity vectors must be included in every boundary crossing; absence of continuity data is itself a governance signal.

8.6 Recursive Drift Amplification

Definition: small semantic errors introduced at early boundary crossings are amplified through successive crossings. Each layer inherits the degraded state as baseline and adds further drift. By the final layer, the governance state has departed significantly from the original without any single crossing failing individual validation.

> *Local admissibility does not necessarily preserve global governability. A system can remain operational, individually coherent, and fully logged while the cumulative legitimacy of the trajectory quietly degrades across layers.*

This formulation captures the failure mode precisely. Each layer may honestly conclude it received a governance-valid state -- but what it actually received may

already contain attenuated continuity inherited from earlier crossings. The architecture accumulates locally valid transitions while producing globally degraded accountability.

Detection: cross-layer Δs must be computed against the L1 baseline, not only the immediately prior layer.

8.7 False Crystallization

Definition: a governance state is crystallized and externally anchored despite not meeting crossing conditions. This occurs when confirmation logic is bypassed -- for example, when timeout triggers override unresolved signal requirements.

Detection: crystallization records must include explicit confirmation path documentation (convergence vs. timeout vs. override).

8.8 Governance Deadlock

Definition: two or more governance layers require each other's confirmation before proceeding, creating a circular dependency that prevents any boundary crossing. This failure mode is specific to recursive architectures with bidirectional confirmation requirements.

Detection: timeout escalation with explicit deadlock classification prevents indefinite blocking.

8b. Recursive Drift Amplification: A Diagram

Recursive drift amplification (failure mode 8.6) deserves special visual treatment because it is the most counter-intuitive failure mode: no single layer fails, yet the final governance state is corrupted. Figure 4 illustrates the dynamics.

Figure 4. Recursive Drift Amplification

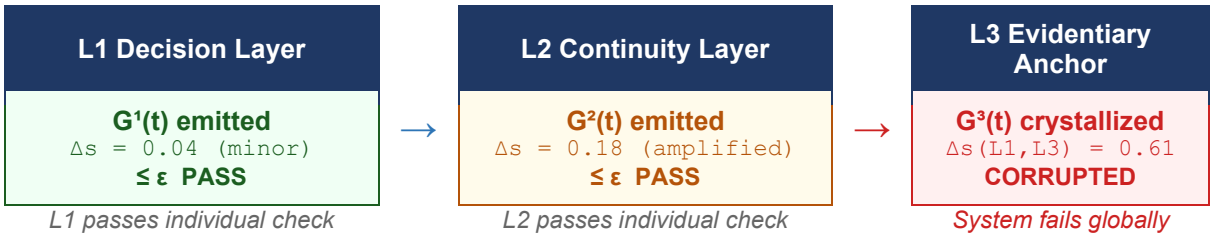


Figure 4 -- Recursive drift amplification. Each individual boundary crossing passes its local check (Δs ≤ ε per-layer). However, the cumulative cross-layer divergence Δs(L1, L3) = 0.61 far exceeds the acceptable threshold. No single layer fails. The system fails. RSG requires Δs computed against the L1 baseline, not only the immediately prior layer.

---

9. End-to-End Walkthrough: AI Insurance Claim Processing

We present a full walkthrough of the RSG framework applied to an AI-assisted insurance claim processing system. This scenario involves three governance layers and three boundary crossings.

9.1 Scenario Setup

An AI system processes an insurance claim involving property damage. The processing pipeline involves:

- L1: AI Decision Engine -- assesses claim validity and proposes settlement amount
- L2: Continuity Governance Layer -- qualifies whether L1's decision authority and continuity remain attributable
- L3: Evidentiary Anchor (EVIDE) -- crystallizes and externally anchors the governance record

The claim involves ambiguous damage assessment: the AI proposes a settlement, but two signals remain unresolved -- a subcontractor liability question and a policy exclusion boundary condition.

9.2 L1 Governance Vector Emission

L1 emits the following governance vector at crossing time t_0:

> $G^1(t_0)$ = ⟨ D: provisional, A: attributed, I: moderate, T: partial, C: attenuated, E: partial ⟩

Notable: classification_status = provisional (not stable), threshold attribution is partial (policy exclusion unresolved), causal persistence C_p = 0.52 (attenuated).

9.3 Recursive Alignment at L1→L2 Boundary

The boundary connector observes $G^1(t_0)$ and initiates recursive alignment:

- Cycle 1: connector detects unresolved_signals = [subcontractor_liability, policy_exclusion_boundary]. Flags threshold vector as non-transferable.
- Cycle 2: L1 resolves subcontractor_liability through additional policy lookup. Emits $G^1(t_1)$ with one signal resolved.
- Cycle 3: policy_exclusion_boundary remains unresolved. Connector confirms split: decision + authority vectors CONFIRMED; threshold vector DEGRADED; policy_exclusion_boundary signal BLOCKED.

Split confirmation result:

- ✓ Decision vector: transmitted (provisional classification preserved, not inflated to stable)

— ✓ Authority vector: transmitted (attribution confirmed)
— ⚠ Threshold vector: transmitted with degradation marker (one signal unresolved)
— ✕ policy_exclusion_boundary: blocked; boundary rejection record generated

9.4 L2 Continuity Qualification

L2 receives the split-confirmed vector and qualifies continuity:

> $C_p = f(S_t=\text{improving}, D_s=\text{partial}, A_t=\text{coherent}) = 0.61 \rightarrow \text{attenuated}$

L2 issues its governance qualification:

— Decision authority: attributable to L1 AI engine (declared)
— Continuity boundary: partially coherent -- one signal remains open
— Handoff integrity: verifiable within declared scope
— Reconstructability: confirmed for resolved signals; not confirmed for blocked signal

L2 non-claims maintained: does not determine claim correctness, does not determine policy interpretation, does not grant payment authorization.

9.5 EVIDE Crystallization

The stabilization score at crystallization:

> $S(t\_close) = 0.4 \cdot 0.61 + 0.3 \cdot 0.65 + 0.3 \cdot (1 - 0.125) = 0.244 + 0.195 + 0.263 = 0.70$

S = 0.70, threshold θ = 0.65. Verdict: crossing-sufficient. DEFERRED (not stable) because unresolved signals remain.

The evidentiary chronology records: 3 alignment cycles, 1 blocked signal (policy_exclusion_boundary), 1 degraded vector (threshold), stabilization_source = mixed (human_review + automated), closure_trigger = convergence (not timeout).

The EVIDE anchor generates: SHA-256 canonical hash, immutable event log with all cycle observations, independent reconstructability declaration scoped to resolved signals only.

9.6 Post-Crossing Outcome

The human claims adjudicator receives:

— A crystallized, externally anchored governance record
— An explicit unresolved signal record (policy_exclusion_boundary) requiring human determination
— A continuity qualification with explicit scope boundaries

- A plain-language governance verdict: crossing-sufficient but not fully stabilized; human review required on blocked signal

This is the correct governance outcome. The AI system proceeded as far as it could within its semantic scope. The boundary was crossed with explicit qualification. The human adjudicator receives precisely what is needed to complete the determination -- neither more nor less.

---

10. Reference Implementation: EVIDE

10.1 EVIDE as Evidentiary Boundary Layer

EVIDE (Evidentiary Deposit and Integrity for AI decisions) is an operational reference implementation of external evidentiary anchoring for governance states. EVIDE operates at the responsibility and evidentiary closure boundary -- it does not govern runtime behavior, but anchors the governance state at the moment responsibility crosses the boundary.

EVIDE v2.0 introduces the boundary_readiness structured object, capturing not just whether a boundary was crossed but what the gate could legitimately observe at crossing time. The four canonical states -- verified, verified_partial, unverifiable, candidate -- directly implement the observability surface dimension of the evidentiary vector.

10.2 Epistemic Stabilization Buffer (ESB)

The ESB is an experimental governance schema that operationalizes Recursive Boundary Alignment within the EVIDE architecture. The ESB inserts a governed observation window between intake and crystallization, tracking:

- Causal persistence signal across the window (present, attenuated, absent, inconclusive)
- Stability trend at each observation cycle
- Stabilization source with epistemic hierarchy (human review > automated convergence > timeout expiration)
- Immutable event log of all observations with timestamped snapshots

The ESB's core semantic distinction directly implements the RSG stabilization score: buffer_verdict = stable means $S \geq \theta$ -- crossing-sufficient, not epistemically true.

10.3 L2 Continuity Governance Layer

The cross-layer composition architecture tested on May 27, 2026 involved an L2 continuity governance layer composable with EVIDE's ESB. The composition verified:

- The two layers remain independently sovereign
- They are composable without merging governance authority
- The composition is externally verifiable through the EVIDE event log
- Explicit boundary: cross_layer_reference_only -- does not verify verdict, does not determine truth

The externally anchored record of this test is available at:
https://lab.certifywebcontent.com/intake/0513fb09-7a8c-4223-b2c8-21262c2d07fa

---

10b. RSG in the Current Ecosystem: Bridges to Existing Frameworks

A theoretical framework that does not connect to the tools practitioners use today risks remaining conceptual. This section explicitly positions RSG within the current AI engineering and governance ecosystem, identifying where RSG addresses gaps that existing frameworks leave open.

10b.1 Model Context Protocol (MCP) and Agentic Boundaries

The Model Context Protocol (MCP) standardizes how AI agents exchange context, tools, and resources across system boundaries. MCP defines the structure of what is exchanged. RSG defines the governance qualification of that exchange.

In an MCP-based agentic system, every tool call is a boundary crossing. The MCP layer handles serialization, transport, and capability exposure. It does not handle:

- Whether the authority to invoke the tool survives the crossing intact
- Whether the semantic governance state that justified the invocation is preserved in the tool response
- Whether continuity of accountability is maintained when the tool result is consumed by the next agent

RSG governance vectors are directly composable with MCP payloads. A governance-aware MCP implementation would attach a governance vector $G^n(t)$ to each context exchange, enabling the receiving agent to verify semantic custody before consuming the result.

> *MCP answers: what can be exchanged? RSG answers: what governance properties must survive the exchange?*

10b.2 LangChain, LangGraph, and Multi-Step Agent Pipelines

LangChain and LangGraph enable multi-step agent pipelines where each step processes, transforms, and passes state to the next. These frameworks excel at orchestrating execution chains. They do not provide:

- Accountability attribution at each step transition
- Detection of semantic drift across chain steps
- Evidentiary anchoring of intermediate governance states
- Non-claims enforcement -- preventing a downstream step from inheriting authority claims not established upstream

RSG boundary-trained connectors map naturally to LangChain's chain composition model. Each chain link is a potential boundary connector. Applying RSG at link boundaries means that governance state is qualified at every step, not only at the final output.

Consider a concrete failure scenario: a LangChain pipeline retrieves customer data (L1), summarizes it (L2), and generates a legally-binding communication draft (L3). Standard telemetry records three successful step completions. RSG would detect that the authority to generate legal communications was not established at L1, was not transmitted to L2, and was implicitly assumed by L3 -- a textbook authority hallucination failure mode.

10b.3 AutoGen and Multi-Agent Conversation Graphs

AutoGen enables multi-agent conversation graphs where agents negotiate, delegate, and cooperate autonomously. The governance challenge is acute: in a fully autonomous agent network, there is no human in the loop to catch accountability failures at runtime.

RSG provides the missing governance substrate for AutoGen-class systems:

- Each agent-to-agent message is a boundary crossing requiring governance vector transmission
- Recursive drift amplification -- the most dangerous RSG failure mode -- is endemic to multi-agent graphs, where small authority errors compound across many agent hops
- Split confirmation propagation allows agents to partially delegate tasks while explicitly flagging unresolved governance signals rather than silently inheriting them

The key insight for AutoGen deployments: the governance failure in a multi-agent system is almost never at the final agent. It occurs at an early boundary crossing and is amplified by subsequent crossings until the system produces a consequence that no individual agent explicitly authorized.

10b.4 Relationship to Existing Governance Standards

RSG does not replace existing AI governance frameworks. It provides the boundary engineering layer that existing frameworks assume but do not specify:

- EU AI Act Art. 12-13 (transparency and traceability requirements): RSG provides the governance vector structure and evidentiary chronology that make traceability mechanically achievable rather than declaratively asserted.
- NIST AI RMF (govern, map, measure, manage): RSG's governance vectors provide measurable accountability state that can feed directly into NIST measurement frameworks.
- ISO/IEC 42001 (AI management systems): RSG boundary taxonomy maps to ISO/IEC 42001 process boundary requirements, providing operational specificity for audit boundary definitions.

RSG is positioned as infrastructure -- not as a compliance framework or a certification scheme. It is the boundary engineering layer on which compliant, certifiable, auditable AI systems can be built.

10b.5 Relationship to Provenance and Explainability Frameworks

RSG is frequently compared to existing frameworks that address related but distinct problems. A precise positioning clarifies where RSG adds value that existing frameworks do not provide.

- W3C PROV (Provenance Framework): PROV tracks the origin and history of data entities through derivation, attribution, and generation relationships. RSG differs in scope: PROV records what happened to data; RSG qualifies whether governance-relevant meaning survived the transitions that produced it. PROV answers: where did this data come from? RSG answers: did accountability survive the journey?
- XAI (Explainable AI) Pipelines: Explainability frameworks focus on making AI decision processes interpretable to human observers -- post-hoc. RSG operates at the boundary during crossing, not after it. XAI explains a decision once made; RSG qualifies the governance state before the decision crosses a boundary. They are temporally and architecturally distinct.
- Chain-of-Thought Governance: Emerging approaches to governing multi-step reasoning chains focus on intermediate reasoning steps within a single model. RSG addresses cross-layer governance -- between distinct systems, agents, or authority domains. The two approaches are complementary: chain-of-thought governance operates intra-model; RSG operates inter-layer.
- Trust Frameworks (e.g. Zero Trust Architecture): Zero Trust verifies identity and access at every request. RSG qualifies semantic continuity at every boundary crossing. Zero Trust asks: is this entity authorized? RSG asks: does the governance

state that justifies this action remain intact? Authorization and governance continuity are related but not equivalent.

The common thread: existing frameworks address individual dimensions of accountability (origin, interpretability, authorization, reasoning). RSG addresses the propagation of accountability across boundaries -- the dimension that becomes critical when governance must survive recursive multi-layer transformations.

11. Implications for AI Governance

- EU AI Act: traceability and human oversight requirements should be interpreted as semantic continuity requirements, not merely documentation requirements.
- Agentic AI: each agent handoff is a boundary crossing; each crossing is a potential semantic custody failure. RSG provides governance at boundary resolution rather than system resolution.
- Formal verification: RSG bridges static formal verification and dynamic runtime governance, addressing the full accountability lifecycle.
- Compliance frameworks: governance vectors provide the structured data foundation for machine-verifiable compliance checking, replacing subjective audit interpretation.

11b. Computational Considerations

A technically-oriented reader will ask: what is the computational cost of recursive stabilization? RSG introduces real overhead relative to single-pass event logging. We identify four design constraints that bound this cost.

- Recursion depth constraint: alignment cycles are bounded by a maximum depth N_max. If convergence is not achieved within N_max iterations, the crossing terminates with a deferred verdict and explicit non-convergence record. N_max is a governance parameter, not a system constant -- it reflects the acceptable epistemic cost of the crossing.
- Bounded stabilization windows: the buffer window is time-bounded. A governance state that does not stabilize within the declared window triggers timeout closure with explicit epistemic weight reduction. This prevents unbounded blocking while maintaining honest evidentiary records.
- Adaptive convergence thresholds: the convergence threshold ε can be adapted based on signal stability history. Consistently stable vectors may require fewer cycles. Chronically degraded vectors may trigger escalation before N_max is reached.
- Partial vector propagation: not all governance vector components require full recursive alignment. Decision and authority vectors, once confirmed, may be

propagated without re-verification in subsequent cycles. Only actively changing components require recursive treatment.

- Probabilistic refinement termination: in high-throughput environments, probabilistic termination criteria can replace deterministic convergence checks, trading perfect semantic custody guarantees for bounded latency with quantified uncertainty.

RSG is not designed as a zero-latency governance architecture. It is designed for decisions where accountability survivability matters more than throughput -- high-stakes crossings where the cost of governance failure exceeds the cost of stabilization latency.

The Governance Overhead Threshold

A critical operational question for any RSG deployment is: when does governance overhead exceed its usefulness? The answer depends on one key relationship:

> *Governance overhead becomes counterproductive when stabilization cost exceeds the causal latency of the consequence itself. At that point, the system is no longer governing propagation -- it is documenting propagation after effective closure has already occurred.*

This implies that RSG cannot be uniformly applied across all boundary crossings. The recursion depth N_max, stabilization window, and alignment requirements should scale dynamically based on:

- Consequence severity -- high-stakes decisions warrant deeper alignment cycles
- Reversibility horizon -- irreversible consequences require earlier crystallization
- Delegation depth -- longer authority chains require stricter continuity qualification
- Authority sensitivity -- ambiguous attribution requires more recursive refinement
- Propagation speed -- fast-moving consequence chains compress the available stabilization window

A governance system that applies uniform recursion depth regardless of these parameters will eventually collapse under its own stabilization burden on low-stakes crossings, while failing to provide sufficient depth on high-stakes ones.

---

12. Limitations

- Governance vector completeness: defining complete vectors for all accountability dimensions remains open. The six types proposed are initial approximations.
- Semantic equivalence criteria: formal equivalence between upstream and downstream governance representations is not yet fully specified.

— Stable state misclassification: the system can stabilize semantically coherent but factually incorrect states. Stabilization does not imply correctness.

— Connector training: Boundary Connector Training methodology requires further formal development.

— Mathematical grounding: the scoring functions proposed are indicative. Empirical calibration of weights and thresholds requires further experimental work.

— Observability limits: systems with insufficient instrumentation cannot support full RSG implementation.

— L1 baseline integrity: Recursive Drift Amplification detection requires computing Δs against the L1 baseline. This assumes the L1 baseline is itself stable, representative, and uncompromised. In systems where the initial governance state is uncertain, contested, or evolving, the L1 anchor may be an unreliable reference point -- propagating baseline uncertainty through all subsequent cross-layer divergence calculations. RSG does not currently specify a mechanism for baseline integrity verification or dynamic baseline correction. Important clarification consistent with Non-Claims Discipline: RSG postulates procedural validity of the emission point -- not its ontological correctness. If L1 itself contains an undetected error, RSG will measure divergence from that error consistently but cannot detect the original error. This is not a deficiency of RSG specifically; it is a general property of any governance system that does not claim to verify the correctness of the decisions it governs.

— Automatic violation detection: detecting failure modes such as non-expansion violations and synthetic coherence automatically requires semantic detection capabilities that are themselves subject to error. A governance layer that misclassifies a legitimate state transition as a non-expansion violation produces false governance failures; one that fails to detect a genuine violation produces silent governance inflation. The reliability of RSG's failure detection is bounded by the reliability of the semantic classification mechanisms it depends on.

---

13. Future Research Directions

— Formal specification of governance vector schemas and semantic equivalence criteria

— Adaptive recursive governance: systems that modify stabilization parameters based on crossing history

— Governance vector embeddings compatible with latent semantic propagation in recursive reasoning systems

— Multi-agent accountability networks: RSG applied to fully distributed agent graphs

— Cross-organizational semantic custody: governance continuity across legal and institutional boundaries

— Continuity-weighted governance memory and semantic survivability metrics

— Formal characterization of recursive drift amplification dynamics

— Automated governance deadlock detection and resolution protocols

---

14. Conclusion

This paper has argued that the dominant model of AI governance -- static, snapshot-based, event-driven, and post-hoc -- is structurally insufficient for modern AI systems operating across multiple layers, agents, and boundaries.

We have proposed Recursive Semantic Governance as an alternative, grounded in formal vector notation, three canonical architectural diagrams, eight failure mode characterizations, and a complete end-to-end walkthrough scenario. The framework is validated through an operational reference implementation -- EVIDE -- whose first cross-layer composition test was completed during the development of this paper.

> *The future of AI governance may depend less on reconstructing accountability after failure, and more on preserving the semantic survivability of responsibility before governance collapse occurs.*

The boundary is where governance succeeds or fails. RSG is a framework for governing the boundary.

Appendix A: Formal Sketch of Core Mathematical Objects

This appendix provides sketch definitions for the core mathematical objects introduced in Section 3. These are not complete proofs or axiomatizations -- they are formal boundary conditions sufficient for technically-oriented readers to assess the mathematical regime of RSG and identify paths toward full formalization.

A.1 Governance State Space

Let Γ denote the governance state space. A governance state $G \in \Gamma$ is a structured object $G = \langle D, A, I, T, C, E \rangle$ where each component is drawn from a domain-specific value space:

- D (Decision domain): finite set of classification statuses {stable, provisional, contested}, threshold satisfaction states {defined, pending, violated}, closure conditions {open, closed, deferred}
- A (Authority domain): structured object with attribution_status {attributed, declared, inferred}, delegation_depth (non-negative integer), conflict_indicators (boolean set)
- I (Intervention domain): structured object with modification_type (categorical), intensity (normalized [0,1]), stability_classification {coherent, degraded, contested}
- T (Threshold domain): structured object with constraint_set (set of applicable normative rules), attribution_status {attributed, pending, unattributable}, satisfaction_state {satisfied, partial, violated}
- C (Continuity domain): structured object with causal_persistence_score $C_p \in [0,1]$, stability_trend {improving, degrading, static, oscillating}, continuity_state {coherent, partially_coherent, fragmented, unverifiable}
- E (Evidentiary domain): structured object with canonical_hash (SHA-256 string), independence_declaration (boolean), observability_surface {declared_complete, partial, insufficient}, unresolved_signals (string array)

Required property: Γ must be closed under the transition function T -- a governance state produced by T must be a valid element of Γ.

A.2 The Semantic Divergence Function d()

The divergence function $d: \Gamma \times \Gamma \rightarrow [0,1]$ must satisfy the following minimum requirements for Δs to be a meaningful governance metric:

- Reflexivity: $d(G, G) = 0$ -- a state compared to itself has zero divergence
- Non-negativity: $d(G_1, G_2) \geq 0$ for all $G_1, G_2 \in \Gamma$
- Asymmetry permitted: d is not required to be symmetric, as upstream-to-downstream semantic loss may differ from downstream-to-upstream
- Triangular inequality NOT required: cross-layer divergence $\Delta s(L_1, L_3)$ may exceed $\Delta s(L_1, L_2) + \Delta s(L_2, L_3)$ -- this is precisely the Recursive Drift Amplification failure mode

Candidate implementations: weighted field-level comparison across vector components (current operational approach); Jensen-Shannon divergence applied to probability distributions over governance state classes; edit distance on authority constraint structures. The choice of implementation is a deployment calibration decision, not a framework invariant.

A.3 The Transition Function T()

The transition function T: Γ × [0,1] × [0,1] → Γ maps a current governance state, semantic divergence, and causal persistence to a next governance state:

$$G(t_{n+1}) = T(G(t_n), \Delta s, C_p)$$

Required properties of T:

- Monotonicity in Δs: as divergence increases, governance state quality should not improve unless explicitly explained by a stabilization cycle
- Causal sensitivity: T must be sensitive to C_p -- a state with absent causal persistence must produce a different output than one with present causal persistence, even at identical Δs values
- Non-expansion: T must not produce governance states with stronger authority claims than the input state -- authority components of $G(t_{n+1})$ are bounded by $G(t_n)$

T is not required to be continuous, differentiable, or linear. It operates on discrete governance events.

A.4 Threshold Calibration

The thresholds ε (semantic divergence) and θ (stabilization score) are deployment parameters, not framework constants. Recommended calibration approaches:

- ε: derived from the historical distribution of Δs under confirmed-stable governance conditions in the target deployment context. A conservative initial value: ε = μ + 2σ, where μ and σ are the mean and standard deviation of Δs in a stable baseline sample
- θ: linked to the consequence severity of the boundary crossing. High-consequence, low-reversibility crossings warrant θ ≥ 0.80. Routine, high-reversibility crossings may operate at θ ≥ 0.55. The relationship between θ and consequence profile is a risk management decision external to the RSG framework

Future work: systematic empirical calibration of ε and θ across deployment contexts, including sensitivity analysis to validate that the framework's governance detection properties are robust to threshold variation.

Acknowledgements

This work emerged from research conducted within the EVIDE Governance Lab during May 2026. The author thanks the EVIDE Lab research community for early feedback on the ESB architecture.

References

[1] RecursiveMAS. (2026). Recursive Multi-Agent System Architecture. GitHub: https://github.com/RecursiveMAS/RecursiveMAS

[2] Celano, E. (2026). EVIDE v2.0 Architectural Backlog and Roadmap. EVIDE Governance Lab: https://app.certifywebcontent.com/docs/evide-v2-roadmap/

[3] Celano, E. (2026). Decision Wave Compression: Technical Note v0.1. https://app.certifywebcontent.com/docs/decision-wave-compression/

[4] EVIDE Governance Lab. (2026). Epistemic Stabilization Buffer: Schema v_buffer_01. lab.certifywebcontent.com

[5] European Parliament. (2024). Artificial Intelligence Act. Official Journal of the European Union.

[6] Celano, E. (2026). H.A.S.H.E.S. Manifesto: Human Alignment for Stability, Harmony, Ethics & Survival. certifywebcontent.com