# Governance Chain Boundary Note

*EFA and EVIDE as Temporally Ordered, Non-Overlapping Layers*

| | |
|---|---|
| **Version** | 1.0 |
| **Date** | May 27, 2026 |
| **Authors** | Prepared by Emanuel Celano referencing EFA (Charles R. Rupp) and EVIDE |
| **Reviewed and approved by** | Charles R. Rupp, DBA (EFA Framework) |
| **Status** | Joint Architectural Note -- Public |

## Purpose

This note maps the Ethical Functionality without Agency (EFA) framework and EVIDE as adjacent, non-overlapping governance layers operating at distinct temporal boundaries within the same AI consequence lifecycle. It also positions both architectures within the Three-Plane Execution Model (Rupp, 2026) as a shared reference structure.

It does not claim authorship over either architecture. It makes the sequencing relationship operationally legible for institutional actors, auditors, and implementers who need to understand how distinct governance layers interact across the full consequence lifecycle -- without collapsing them into a single control surface.

## The Problem This Note Addresses

Most current AI governance discourse treats governance as a single, unified surface. In practice, at least three structurally distinct problems must be solved at three distinct moments:

- What is observable inside the system before the boundary?
- Is the boundary crossing justified at the exact moment of consequence formation?
- Is the responsibility-bearing closure state independently reconstructable after authorization?

These are not the same problem. Conflating them produces architectures that are monolithic, fragile under audit, and difficult to contest or replace without systemic collapse.

## Relationship to the Three-Plane Execution Model

The Three-Plane Execution Model (Rupp, 2026) proposes a vendor-agnostic reference architecture that separates execution-boundary governance into three distinct, interoperable planes: Signal Formatting (Plane 1), Adjudication Logic (Plane 2), and Mechanical Enforcement (Plane 3). The model describes governance up to and including enforcement, providing a shared architectural language for evaluating any governance solution.

The three governance layers described in this note map directly onto that model -- and extend it with a post-Plane 3 layer that the model does not yet address:

- **Layer 1 (Runtime Observability)** maps to Plane 1 -- Signal Formatting. Produces internal trace events, continuity signals, responsibility object coherence indicators, and attribution consistency records before the signal reaches the adjudication boundary.
- **Layer 2 (EFA -- Boundary Governance)** maps to Plane 2 and Plane 3 -- Adjudication Logic and Mechanical Enforcement. JRT/MECHA adjudicate justified reliance; REL enforces the ALLOW / ESCALATE / PAUSE / BLOCK decision.
- **Layer 3 (EVIDE -- Post-Authorization Closure)** operates as a post-Plane 3 layer -- entering only after ALLOW has been enforced and a governance decision record exists. It anchors the finalized responsibility-bearing closure state in an independently verifiable, externally reconstructable form.

The result is that the full governance sequence -- from runtime observability through adjudication, enforcement, and post-authorization evidentiary anchoring -- has distinct, non-overlapping architectural candidates for each plane, and one layer beyond the three-plane model that the current reference architecture does not yet address.

## The Three-Layer Sequence

### Layer 1 -- Runtime Observability (Pre-Gate)

**Function:** Produces internal visibility of the system's generation state before boundary crossing. Generates trace events, continuity signals, responsibility object coherence indicators, and attribution consistency records.

**Temporal position:** Operates inside the system, before the execution boundary.

**What it does NOT do:** It does not decide whether consequence may bind. It does not cross the boundary. It does not produce an externally anchorable closure object.

**Output:** Structured governance signals and observability data that arrive at the boundary gate.

### Layer 2 -- Boundary Governance (At the Gate)

**Framework:** EFA -- Ethical Functionality without Agency (Charles R. Rupp)

**Core components:** Justified Reliance Threshold (JRT), MECHA conditions, Runtime Enforcement Layer (REL), Dimensional-Collapse Gate (DCG)

**Function:** Receives governance-relevant signals at the execution boundary. Evaluates whether the full human-AI decision system is justified in permitting consequence to bind at that exact moment, through five conjunctive MECHA conditions:

- **M -- Machine State** (is the machine stable enough to recommend?)
- **E -- Evidence State** (is the evidence sufficient?)
- **C -- Commitment Class** (what is the consequence severity?)
- **H -- Human State** (is the human sovereign enough to concur?)

- **A -- Authority State** (is authority valid at this crossing?)

REL enforces the gate and returns one of four canonical states: ALLOW / ESCALATE / PAUSE / BLOCK.

**Temporal position:** Operates at the boundary. This is the decisive adjudication point.

**What it does NOT do:** It does not produce the downstream evidentiary anchoring. It does not monitor the system after the crossing. Before REL returns ALLOW, no finalized closure state exists to anchor.

**Output:** A governed boundary crossing decision. Only ALLOW produces a finalized responsibility state that can be independently anchored downstream.

### Layer 3 -- Post-Authorization Evidentiary Closure (Post-Gate)

**Framework:** EVIDE -- External Evidentiary Deposit (Emanuel Celano)

**Specification:** evide-intake-schema v2.x, evidentiary_profile v1.1

**Function:** Enters only after ALLOW or authorized escalation closure. Receives the finalized boundary state -- not the runtime trace. Anchors the responsibility-bearing closure object in an independently verifiable, externally reconstructable form.

The closure object preserves:

- who held authority at the decision boundary
- whether human intervention occurred and under what authority
- the closure state of the decision at crossing-time
- the upstream classification structure active at closure
- boundary_readiness and runtime-condition stability at crossing
- cryptographic integrity and independent anchoring

**Temporal position:** Operates after the boundary crossing has been authorized.

**What it does NOT do:** EVIDE does not monitor runtime continuously. It does not evaluate whether the crossing was justified (that is EFA's function). It does not absorb runtime observability -- doing so would destroy the evidentiary independence that is its primary institutional value.

**Output:** A portable, independently verifiable closure object capable of surviving audits, disputes, system migrations, and cross-organizational review without reconstructive dependency on the source environment.

## The Complete Governance Sequence

| | |
|---|---|
| **AI generates** | Probabilistic output is produced by the generative system. |
| **L1 -- Runtime Observability** | Internal continuity signals, trace events, and attribution consistency records are produced inside the system boundary. |

| | |
|---|---|
| **L2 -- EFA / Dimensional-Collapse Gate** | MSS evaluates machine-state stability. JRT / MECHA evaluate justified reliance. REL returns: ALLOW / ESCALATE / PAUSE / BLOCK. |
| **L3 -- EVIDE (only if ALLOW)** | EVIDE anchors the finalized responsibility-bearing closure state independently, portably, and reconstructably. |
| **Future audit / dispute / review** | No reconstructive dependency on the originating system. Each layer is independently contestable. |

## Why Non-Overlap Matters Institutionally

The three layers do not overlap because they operate at distinct temporal moments of the same sequence. Each layer can be independently challenged, replaced, upgraded, or audited without collapsing the integrity of the others.

This is not an incidental design choice. It is the architectural property that makes layered AI governance resilient to:

- jurisdictional variation
- technical evolution
- institutional contestation
- vendor or platform replacement

A monolithic architecture that collapses runtime observability, admissibility adjudication, authority attribution, and evidentiary persistence into a single control surface cannot satisfy these requirements. Temporally ordered, functionally distinct governance layers can.

## Key Boundary Distinctions

| Dimension | Layer 1 -- Pre-Gate | Layer 2 -- EFA (At Gate) | Layer 3 -- EVIDE (Post-Gate) |
|---|---|---|---|
| **Temporal position** | Before boundary | At boundary | After boundary |
| **Primary question** | What is observable? | Is reliance justified? | Is closure independently anchorable? |
| **Produces** | Internal signals | ALLOW / ESCALATE / PAUSE / BLOCK decision | Portable closure object |
| **Crosses boundary?** | No | Yes -- governs the crossing | Enters after crossing |
| **Depends on prior layer?** | No | Receives signals from L1 | Enters only after ALLOW |
| **Replaces other layers?** | No | No | No |

## Attribution and Relationship

EFA and EVIDE were developed independently, by different researchers, working on adjacent boundaries of the same structural problem. The convergence was not coordinated by design.

- **EFA (Charles R. Rupp)** governs whether consequence may bind at the execution boundary.
- **EVIDE (Emanuel Celano)** governs whether the finalized responsibility state remains independently reconstructable after authorization has occurred.

These are not competing architectures. They are temporally ordered solutions to different moments of the governance chain.

## References


Rupp, Charles R.. *The Three-Plane Execution Model: A Reference Architecture for Structural AI Governance.* Version 1.0, May 2026. DOI: 10.5281/zenodo.20403788

Rupp, Charles R.. *The Dimensional-Collapse Gate: Governance at the Moment of Binding Consequence.* EFA Technical Note, Version 1.2, May 2026. DOI: 10.5281/zenodo.20284325

Rupp, Charles R.. *Machine State Signal (MSS): Governing the Governor's Instrument.* Version 1.2, May 2026. DOI: 10.5281/zenodo.19992746

Rupp, Charles R.. *Justified Reliance Threshold (JRT): Governing the Moment of Consequence.* Version 1.1, May 2026. DOI: 10.5281/zenodo.20018067

Rupp, Charles R.. *Runtime Enforcement Layer (REL): Non-Bypassable Execution Control for EFA.* Version 1.2, May 2026. DOI: 10.5281/zenodo.20115642

Rupp, Charles R.. *The Ethical Functionality without Agency (EFA) Canon: Complete Module Index.* Version 1.1, May 2026. DOI: 10.5281/zenodo.20324644

Celano, Emanuel.. *EVIDE -- External Evidentiary Deposit: Evidentiary registry for digital content and AI decisions.* Version 2.0, 2026.

Published at: https://app.certifywebcontent.com/

Celano, Emanuel.. *EVIDE JSON Schema and API Documentation.* Version 2.x, 2026.

Published at: https://app.certifywebcontent.com/docs/evide-intake-schema/