

Three Primitives — Canonical Logic Sequence (Clean-room v1.2)

Pyrate Ruby Passell Stacy Gildenston

December 28, 2025

Abstract

This document defines three irreducible governance primitives—Authority \neq Outcome, Permission \neq Decision, and Declared Authority—to establish a formal framework for structural legitimacy in automated systems. Through a clean-room logic sequence, it demonstrates that while escalation detection is a computable function, the mapping from escalation to action selection is non-unique, necessitating an explicit human authority function. The sequence provides a total control flow for escalated events, integrating constraint-scoped action sets and mandatory liveness revalidation to ensure that authority remains unautomatable.

Status & Scope

- **Status:** Clean-room v1.2 — DOI readiness pass
- **Scope:** Formal definition, logic audit, and structural closure

1 Escalation Boundary and Function Separation ($g \neq f$)

Domain Definition: Let X_t be a temporally ordered stream of observable system events. Each event $x \in X_t$ is represented as a structured signal vector $f(x) \in \mathbb{R}^n$. Let $A(x)$ be the set of possible actions associated with event x .

Escalation Function (E): $E : X_t \rightarrow \{0, 1\}$, where $E(x) = 1$ indicates an event crosses a predefined escalation boundary. E must be computable, deterministic, and invariant under implementation.

Escalation Boundary: Defined as the minimal decision surface where $E(x) = 1$ and $|A(x)| > 1$, making action selection non-unique.

Axiom (Non-Equivalence): $g \neq f$, where g is the Permission Function (authorizing execution) and f is the Decision Function (realizing the action). Any system that proceeds from $E(x) = 1 \rightarrow f(x)$ without an explicit permission function g violates structural legitimacy.

2 Authority Interception and Temporal Ordering

The Null Action: Represented exclusively by $g(x) = 0$. If $g(x) = 1$, exactly one element from the action set $A(x)$ must be selected by an external authority.

Mandatory Sequence:

1. Observe $x \in X_t$ and compute $E(x)$.

2. If $E(x) = 1$: Declare $g(x)$.
3. If $g(x) = 1$: External authority a_u selects action $a \in A(x)$.

3 Declared Authority and Execution Record ($R(x)$)

Declaration Tuple (δ): $\delta(x) = (p, a_u, c)$, where p is Purpose, a_u is the live Authority holder, and c represents Constraints. $\delta(x)$ must be declared prior to execution and is immutable for that event.

Execution Record ($R(x)$): $R(x) = \langle x, \delta(x), a_{\text{selected}} \rangle$. No execution may occur without this immutable audit record.

4 Final Control Sequence for Structural Legitimacy

1. Observe x ; compute $E(x)$. If $E(x) = 0$, exit escalation path.
2. Identify live a_u ; verify Liveness $L(a_u) = 1$; declare $\delta(x)$.
3. Derive $A_{\text{legit}}(x, \delta(x))$; if empty, terminate via structural impossibility.
4. Present A_{legit} to a_u ; a_u declares Permission $g(x, a_u)$.
5. If $g = 1$: Re-verify liveness; a_u selects action; emit $R(x)$; execute.

Closure Statement

The three primitives are jointly necessary and sufficient for structural legitimacy. All branches are total, no deadlocks remain, and no implicit authority is exercised. The Canonical Logic Sequence v1.2 is formally closed.