



FORMAL RECORD 1

Three Primitives Canonical Logic Sequence

Authors: Pyrate Ruby Passell, Stacy Gildenston
Affiliation: 3primitives.io
Version: v2.3 | September 2026
License: CC BY 4.0 | https://3primitives.io/formal_records/

v2.3 states the three primitives as numbered definitions in §0; they are made explicit here, not altered, and the closure statement in §4 now refers to them directly. A provenance note has been added and the suggested citation now carries the date of first publication. No proof, result, or logical content has changed.

Scope and Provenance

Scope: Formal definition, logic audit, and structural closure

Provenance. In the last two days of November 2025, at a hackathon run by MLAI and sponsored by the Office of the Australian eSafety Commissioner, Pyrate Ruby Passell built a working API safety system that exhibited the structure this record formalises. The primitives were uncovered from that build, not derived from theory. Stacy Gildenston spent the following three weeks establishing what the build had demonstrated, closing the three governance primitives on 24 December 2025. This record was first published on 28 December 2025.

The formulation was developed without reference to any existing governance framework. A continuing search of the governance, alignment and formal-methods literature has not located an equivalent statement of these primitives; any prior equivalent formulation would be received as correction.

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.

0. The Three Primitives

Definition 1 (Authority $\neq$ Outcome). Legitimacy is not a property of the action produced. Two system states may yield the same action while differing in whether that action is authorised. Authority is therefore not recoverable from the outcome.

Definition 2 (Permission $\neq$ Decision). The permission function g is distinct from the decision function f . g authorises execution; f realises the action. No system may proceed from $E(x) = 1$ to $f(x)$ without an explicit g .



Definition 3 (Declared Authority). Permission requires a declaration $\delta(x) = (p, au, c)$ supplied externally, prior to execution, and immutable for that event. δ is not derivable from system state.

The three primitives are properties of legitimate action. They are distinguished from the declaration δ , which is the object crossing the gate (FR06), and from the boundary conditions they force at the point of exit (FR12).

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 au selects action $a \in A(x)$.

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

Declaration Tuple (δ)

$\delta(x) = (p, au, c)$, where p is Purpose, au 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 ($R(x)$)

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

4. Final Control Sequence for Structural Legitimacy

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

Closure Statement

The three primitives defined in §0 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 is formally closed.

Attribution and Licence

This document is the intellectual property of Stacy Gildenston and Pyrate Ruby Passell, held under 3primitives.io. Published under Creative Commons Attribution 4.0 International (CC BY 4.0). You are free to share, adapt, and build upon this work for any purpose, including commercially, provided you give appropriate credit to 3primitives.io and Stacy Gildenston and Pyrate Ruby Passell as originators.

Suggested citation:

Passell, P.R. & Gildenston, S. (2025/2026). Three Primitives — Canonical Logic Sequence. Three Primitives Framework, Formal Record 1, v2.3. Melbourne, Australia. First published 28 December 2025. 3primitives.io. CC BY 4.0.