

Execution Without Normativity

A Minimal Theory of Deterministic Execution and Observation

Lukas Pfister

December 27, 2025

Abstract

This paper presents a minimal axiomatic theory of deterministic execution and observation. The theory isolates execution structure from any notion of meaning, policy, or governance. It models atomic execution events, an append-only ordered execution stream, and observations that are explicitly non-derivational. The core guarantees are structural determinism of canonical projections, observational non-interference, and replay equivalence under fixed authoritative execution inputs. The theory does not address correctness, semantics, or domain-specific interpretations.

1 Introduction

Deterministic execution is often discussed as a property of outcomes. This view conflates execution with meaning and treats observational artifacts as if they were part of the execution structure. When observational changes are allowed to influence deterministic projections, replay and audit cease to be reliable.

This paper introduces a minimal axiomatic model that separates execution structure from observation without introducing normativity. The goal is to define what it means for an execution to be deterministic in a structural sense, independent of any domain semantics. The model is intentionally small and focuses on order, replay, and explicit treatment of observations.

We present formal foundations, axioms, and proof sketches of observational non-interference and replay equivalence. The results are purely structural and apply to any system that can produce an ordered stream of execution events.

2 Scope and Non-Goals

This paper defines a minimal theory of deterministic execution and observation. It deliberately excludes any notion of normativity, governance, or domain meaning. The model is structural and is intended to support precise replay and audit of execution structure without interpretation.

Out of scope. The following are explicitly excluded:

- policy, governance, or decision logic
- application semantics or correctness
- alignment, safety, or ethical evaluation
- post-execution filtering as a control mechanism

- distributed consensus, fault tolerance, or availability guarantees
- side-channel resistance or covert channel analysis

The theory addresses execution order and observational separation only.

3 Conceptual Foundations

We distinguish execution structure from observation and avoid assigning meaning to either. The theory uses only structural terms.

Execution. Execution is a sequence of atomic events that can be ordered without reference to wall clock time. Execution is not defined by outcome meaning, only by the order and identity of events.

Order and time. Order is derived from sequence position. Time is not used to define order. Clock time may be recorded as observation but has no role in deterministic projection.

Observation and derivation. Observations are artifacts produced by execution. They are not derivational inputs to deterministic projections. A projection that changes when observations change is not deterministic in the sense of this theory.

Structural determinism. Determinism is defined as invariance of a canonical projection under fixed authoritative execution inputs I_{exec} . It is a property of structure, not of outcome meaning.

Normativity and derivation. Execution is descriptive. Normativity is an external operator not defined in this theory. Observations are explicitly excluded from derivations that define deterministic projections.

4 Formal Model

We introduce a minimal formal model based on atomic events and an append-only execution stream.

Definition 4.1 (Atomic events). Let Δ denote the set of atomic execution events. An event is a minimal unit of execution and may carry an observation payload.

Definition 4.2 (Execution stream). Let $V = \langle e_0, e_1, \dots \rangle$ be an append-only stream of events. Each event $e_i \in \Delta$ has a position index $t(e_i) = i$. The ordering defined by t is the only execution order used in the model.

Definition 4.3 (Observation artifacts). Let O denote the set of observation artifacts associated with events. Observations are attached to events but are not inputs to deterministic projections.

Definition 4.4 (Authoritative execution inputs). Let I_{exec} denote the authoritative execution inputs required to identify and order events in V and to evaluate the canonical projection. By definition, I_{exec} excludes observation artifacts. It may include event identifiers and structural metadata needed to interpret event order, but it does not include outputs, traces, timing data, errors, or metrics.

Definition 4.5 (Canonical projection). A canonical projection is a function π that maps an execution stream to a deterministic representation, written $\pi(V; I_{\text{exec}})$. The projection must depend only on the ordered sequence of events and on fixed authoritative execution inputs. Observations are excluded from π by definition.

Properties of π . The canonical projection satisfies the following minimal properties.

- **Well-defined.** For fixed V and fixed I_{exec} , the value $\pi(V; I_{\text{exec}})$ is unique.
- **Order-respecting.** If two streams differ only by event order, their projections may differ. Order is the only temporal input.
- **Observation-free.** Changing observation artifacts alone does not change $\pi(V; I_{\text{exec}})$.
- **Effective.** There exists a finite procedure that computes $\pi(V; I_{\text{exec}})$ from its inputs.

5 Axioms

We state five axioms that characterize deterministic execution and observational separation.

A1 Atomicity of Δ . Execution is composed of atomic events. No event can be partially applied or split into smaller execution units for the purposes of ordering or replay.

A2 Append-only ordered stream V . The execution stream V is append-only and totally ordered by the index $t(e)$. Events are immutable once appended.

A3 Order is independent of wall clock time. The order induced by t is defined by stream position only. Wall clock time is observational and does not affect order.

A4 Observations are non-derivational. Observations may be recorded and may vary arbitrarily. The canonical projection $\pi(V)$ must not depend on observations. Changes in observations alone cannot change $\pi(V)$.

A5 Structural determinism of projections. For fixed authoritative execution inputs I_{exec} , the canonical projection $\pi(V; I_{\text{exec}})$ is deterministic and invariant under execution replay. Authoritative execution inputs are those required to identify and order events in V and to evaluate π ; observational artifacts are excluded.

6 Observational Non-Interference

We formalize the separation between observation and deterministic projection. Let V be an execution stream and let O be the set of observations attached to events. Two executions are observation-equivalent if they have identical ordered event sequences and differ only in observations.

Definition 6.1 (Observation-equivalence). For execution streams V and V' , we write $V \sim_O V'$ if they contain the same ordered sequence of events and differ only in observation artifacts. For fixed I_{exec} , a projection π is observation-free if $V \sim_O V'$ implies $\pi(V; I_{\text{exec}}) = \pi(V'; I_{\text{exec}})$. This aligns with classical noninterference formulations in which observations do not influence permitted derivations [2, 5].

Lemma 6.2 (No observational expansion of I_{exec}). Observational changes cannot extend the authoritative execution input domain used by π . If $V \sim_O V'$, then the admissible authoritative inputs I_{exec} for evaluating π are identical for V and V' . This follows from the definition of I_{exec} and A4, which excludes observational artifacts from deterministic projections.

Lemma 6.3 (Observation-free projection). Under A4 and A5, the canonical projection π is observation-free. Therefore, changing observations does not change deterministic projections for fixed I_{exec} .

Theorem 6.4 (Observational non-interference). For all execution streams V and V' and for fixed I_{exec} , if $V \sim_O V'$, then $\pi(V; I_{\text{exec}}) = \pi(V'; I_{\text{exec}})$.

Proof sketch. By Lemma 6.2, observation changes do not alter the admissible authoritative inputs I_{exec} used by π . By A4, observations are excluded from the domain of deterministic projections. Therefore π depends only on event order and I_{exec} , which are identical under $V \sim_O V'$. Lemma 6.3 then yields $\pi(V; I_{\text{exec}}) = \pi(V'; I_{\text{exec}})$.

Example. Consider two executions with the same ordered events but different observation payloads attached to those events. Under $V \sim_O V'$, the observation changes do not enter I_{exec} and therefore do not change π . The deterministic projection is identical in both executions.

Corollary. Observation does not interfere with deterministic reconstruction. Observations can be recorded, but they do not define or alter the deterministic structure of execution.

7 Replay Equivalence

Replay is defined as reconstruction of the canonical projection from the execution stream alone.

Definition 7.1 (Replay procedure). Given $V = \langle e_0, e_1, \dots \rangle$, replay applies the projection π to the ordered stream, yielding $\pi(V; I_{\text{exec}})$. The procedure is deterministic under A2 and A5 and corresponds to state reconstruction from an ordered event trace [3].

Definition 7.2 (Replay equivalence). Two executions are replay-equivalent if they produce the same canonical projection. Formally, $V \equiv_R V'$ iff $\pi(V; I_{\text{exec}}) = \pi(V'; I_{\text{exec}})$. Under A2 through A5, any two executions with identical ordered event sequences are replay-equivalent, regardless of observation artifacts.

Lemma 7.3 (Deterministic replay). For fixed I_{exec} , replay of a stream V yields a unique projection $\pi(V; I_{\text{exec}})$. This follows from A2 and A5, which fix order and exclude observational variation from the projection.

Theorem 7.4 (Replay equivalence under fixed I_{exec}). If two executions have identical ordered event sequences and fixed I_{exec} , then they are replay-equivalent.

Proof sketch. Let V and V' be two executions with identical ordered event sequences. By A2, the order is fixed by position and by A5 the projection depends only on this order and I_{exec} . By Lemma 7.3, replay of each stream yields a unique projection, and the identical ordered structure implies the same projection. Therefore $\pi(V; I_{\text{exec}}) = \pi(V'; I_{\text{exec}})$ and $V \equiv_R V'$.

Consequence. Replay equivalence defines an execution class that ignores observation and depends only on ordered event structure and fixed authoritative execution inputs I_{exec} .

8 Implications

The model provides a minimal substrate for deterministic execution structure and observational separation. It can support systems that need replayable execution structure without introducing meaning, normativity, or governance. Event sourcing provides historical context for ordered event reconstruction, but the present model is intentionally narrower and avoids domain semantics [1]. The implications are structural rather than semantic: execution order is explicit, observation is isolated, and deterministic projections are stable under replay.

Comparison to related formalisms. Event sourcing treats events as domain-level changes and typically embeds meaning into event content, while this model treats events as structure and removes domain semantics from projection [1]. Formal specification approaches such as TLA+ model state transitions and temporal properties, whereas this model isolates order and replay without committing to state semantics [3]. Classical noninterference focuses on preventing forbidden information flow between security domains, while this model treats observations as non-derivational inputs by construction and focuses on structural determinism [2, 5].

9 Limitations

The theory does not claim correctness or meaning of execution outcomes. It does not provide any policy or governance logic. It does not address side channels, distributed consensus, or fault tolerance. It does not constrain execution behavior beyond structural ordering and observational separation. Runtime verification approaches are out of scope and are not modeled here [4].

10 Conclusion

We presented a minimal axiomatic theory of deterministic execution and observation without normativity. The theory defines atomic events, an append-only ordered stream, and observation as a non-derivational artifact. Under the axioms, canonical projections are deterministic, observation-free, and replayable. The result is a structural notion of determinism that is independent of outcome meaning or domain semantics.

References

- [1] Martin Fowler. Event sourcing. *martinfowler.com*, 2005.
- [2] Joseph A. Goguen and Jose Meseguer. Security policies and security models. In *IEEE Symposium on Security and Privacy*, 1982.

- [3] Leslie Lamport. *Specifying Systems*. Addison-Wesley, 2002.
- [4] Martin Leucker and Christian Schallhart. A brief account of runtime verification. *Journal of Logic and Algebraic Programming*, 2009.
- [5] John Rushby. Noninterference, transitivity, and channel-control security policies. *SRI International Technical Report*, 1992.