

# Temporal Alignment in Distributed Observability Systems

A Structural Account of Time Semantics, Ordering, and Cross-Layer Inference

**Alexey A. Nekludoff**

AstraVerge Research

E-mail: [an@astraverge.org](mailto:an@astraverge.org)

ORCID: 0009-0002-7724-5762

17 February 2026

## Abstract

Distributed observability systems integrate measurements from heterogeneous instrumentation pipelines with distinct sampling intervals, aggregation windows, buffering behavior, and timestamp semantics. Despite this heterogeneity, operational indicators are routinely aggregated and interpreted as if they shared a common temporal structure.

This paper argues that temporal alignment is a prerequisite for legitimate cross-layer inference. When heterogeneous time domains are implicitly aligned, numerical indicators may appear stable and comparable while encoding non-equivalent temporal structures. In such cases, global orderability is assumed rather than explicitly constructed.

We introduce a minimal formal framework for local time domains, alignment operators, and non-commutative temporal transformations including sampling, windowing, and aggregation. Using this framework, we identify recurring failure modes in distributed observability practice, such as false stability, phantom outages, and structurally unstable SLO and SLA evaluations.

The analysis situates temporal alignment within the broader requirements for coherent inter-local inference, as articulated in Coherent Observational Epistemology (COE), and treats temporal misalignment as a structural, rather than incidental, constraint on what can be meaningfully inferred from distributed observability data.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Contributions . . . . .	1
<b>2</b>	<b>Observability as a Temporal Process</b>	<b>1</b>
2.1	Observational Streams and Local Time Domains . . . . .	1
2.2	Sampling, Windowing, and Aggregation . . . . .	2
2.3	Temporal Provenance . . . . .	2
<b>3</b>	<b>The Temporal Misalignment Problem</b>	<b>3</b>
3.1	Non-Common Time Domains . . . . .	3
3.2	Jitter, Drift, Buffering, and Ingestion Delay . . . . .	3
3.3	Non-Isomorphic Temporal Structures . . . . .	3
<b>4</b>	<b>Alignment Operators and Their Structural Cost</b>	<b>4</b>
4.1	Alignment as Constructed Compatibility . . . . .	4
4.2	Classes of Alignment Operators . . . . .	4
4.3	Alignment Cost and Information Loss . . . . .	4
4.4	Alignment Cost and Distortion . . . . .	5
<b>5</b>	<b>Temporal Misalignment and the Illusion of Reliability</b>	<b>5</b>
5.1	SLO/SLA as Temporally Constructed Assertions . . . . .	5
5.2	False Confidence Through Window Compatibility . . . . .	6
5.3	SLA Disputes as Alignment Disputes . . . . .	6
5.4	Alignment-Aware Reliability . . . . .	6
<b>6</b>	<b>A Minimal Formal Example</b>	<b>6</b>
6.1	Equal Aggregates, Non-Equivalent Meaning . . . . .	6
<b>7</b>	<b>Failure Modes in Distributed Observability</b>	<b>7</b>
7.1	False Stability . . . . .	7
7.2	Phantom Outages . . . . .	7
7.3	SLO/SLA Distortion . . . . .	7
7.4	Cross-Team and Cross-Product Incomparability . . . . .	8
<b>8</b>	<b>Temporal Alignment and COE</b>	<b>8</b>
8.1	GOP1 as a Global Orderability Requirement . . . . .	8
8.2	Transformational Coherence and Pipeline Equivalence . . . . .	8
<b>9</b>	<b>Toward Alignment-Aware Observability Architectures</b>	<b>9</b>
9.1	Design Principles . . . . .	9
9.2	Temporal Observability Contract . . . . .	9
<b>10</b>	<b>Related Work</b>	<b>10</b>
<b>11</b>	<b>Conclusion</b>	<b>10</b>

# 1 Introduction

Distributed observability systems are built from multiple instrumentation pipelines, each designed to capture a particular aspect of system behavior. These pipelines often differ in sampling rhythm, aggregation strategy, window semantics, and ingestion latency.

In practice, metrics produced by such pipelines are routinely aggregated, compared, and used as the basis for operational and managerial decisions. Temporal inconsistencies between streams are commonly treated as secondary implementation details, to be mitigated through resampling or smoothing.

This paper argues that such inconsistencies constitute a structural condition rather than a technical inconvenience. Temporal structure determines which aspects of system behavior can be observed, compared, and inferred. When heterogeneous temporal domains are implicitly aligned, numerical indicators may appear stable and comparable while referring to incompatible behavioral histories.

The central claim of this work is that temporal alignment is a prerequisite for legitimate cross-layer inference. Absent explicit construction of shared temporal structure and declared equivalence conditions, global indicators operate beyond their epistemic justification.

The analysis treats temporal misalignment as a first-order problem of distributed observability, with concrete implications for reliability assessment, incident analysis, and service-level governance. The discussion is grounded in operational practice while framed within the general requirements for coherent inter-local inference as articulated in Coherent Observational Epistemology (COE).

## 1.1 Contributions

The paper makes the following contributions:

- It introduces a minimal formalization of local time domains and temporal structure in observability pipelines.
- It develops a taxonomy of temporal alignment operators and characterizes the structural costs and distortions they introduce.
- It identifies recurring failure modes in distributed observability systems attributable to implicit temporal misalignment.
- It interprets temporal alignment as a concrete prerequisite for Global Orderability (GOP1) and Transformational Coherence within the framework of COE [1].

## 2 Observability as a Temporal Process

### 2.1 Observational Streams and Local Time Domains

Let  $T$  denote an idealized global temporal domain representing the ordering of events in the observed system. In practice, observability does not operate directly on  $T$ . Each measurement stream accesses the system through a locality-specific temporal interface.

Formally, each stream operates on its own local time domain  $T_i \subseteq T$  and produces observations as a function

$$X_i : T_i \rightarrow \mathbb{R}.$$

The definition of  $T_i$  is not merely a technical detail. It encodes how the system is observed. Local time domains may differ with respect to:

- sampling interval (e.g., fixed-period scraping),

- event-driven emission (e.g., transactional logs),
- buffering and delayed delivery,
- timestamp assignment rules (event time vs ingestion time),
- clock synchronization and drift.

As a result, two streams observing the same underlying system may induce distinct temporal orderings, even when their timestamps appear comparable.

## 2.2 Sampling, Windowing, and Aggregation

Raw observations are rarely consumed directly. Instead, they pass through a sequence of temporal transformations that define the effective semantics of the resulting metric.

We distinguish the core classes of transformations commonly applied to observational streams:

- sampling and resampling operators  $S$ ,
- window projection operators  $W_\Delta$ ,
- aggregation operators  $\mathcal{A}$  (mean, max, quantiles, ratios),
- downsampling and retention operators  $D$ .

These operators do not merely reduce data volume; they actively reshape temporal structure. In particular, they define which variations are preserved, which are suppressed, and which are rendered invisible.

*Remark 1.* In general, these operators are not commutative. For example,

$$\mathcal{A} \circ W_\Delta \circ S \neq S \circ \mathcal{A} \circ W_\Delta.$$

This non-commutativity implies that two pipelines constructed from the same operators, but applied in different orders, may yield metrics with distinct temporal semantics. Such metrics cannot be assumed to be equivalent, even if they share a name and unit.

This property is central to distributed observability: the meaning of a metric is inseparable from the order in which temporal transformations are applied.

## 2.3 Temporal Provenance

Operational dashboards often present metric values as if they were directly comparable across systems, services, or layers. However, a displayed scalar value typically represents the output of a multi-stage temporal pipeline.

We use the term *temporal provenance* to denote the ordered sequence of transformations that map raw observations into a reported metric. This includes: sampling rhythm, window definitions, aggregation operators, and retention policies.

Without explicit declaration of temporal provenance, comparisons implicitly assume that distinct metrics share a common temporal structure. Such assumptions may hold locally, but they do not hold by default across heterogeneous pipelines.

Temporal provenance is therefore a prerequisite for justified comparison. Absent this information, numerical similarity may conceal structural divergence in what is actually being observed.

### 3 The Temporal Misalignment Problem

#### 3.1 Non-Common Time Domains

Layered observability systems rarely operate over a single, globally constructed temporal domain. Instead, each observational layer induces its own effective time base.

For a typical layered architecture one encounters distinct local domains:

$$T_I \subseteq T, \quad T_S \subseteq T, \quad T_B \subseteq T,$$

corresponding, for example, to infrastructure-level monitoring, service-level telemetry, and business-level event observation.

These domains differ not only in resolution, but in how time is operationally defined. Infrastructure metrics often rely on periodic sampling. Service telemetry may be aggregated over fixed windows. Business observations are frequently event-driven and sparse.

As a result, the inclusion relations above do not imply temporal compatibility. Even when all domains are embedded into a shared wall-clock, their induced orderings may not admit a consistent global alignment.

Absent an explicit construction of a common ordering space, cross-layer aggregation implicitly assumes temporal equivalence that has not been verified.

#### 3.2 Jitter, Drift, Buffering, and Ingestion Delay

Temporal misalignment is further amplified by runtime effects commonly treated as secondary implementation details.

Sampling jitter introduces variability in effective observation times. Clock drift and synchronization error distort relative ordering. Buffered transmission and batch ingestion decouple event occurrence time from observation availability.

Crucially, these effects do not merely add noise. They modify the temporal structure of observational streams.

A buffered metric stream may smooth short-lived failures. A delayed ingestion pipeline may reorder events across windows. A resampled stream may suppress burst behavior entirely.

From a structural perspective, these mechanisms act as implicit temporal transformations. Treating them as negligible noise obscures their role in shaping the observable system behavior.

#### 3.3 Non-Isomorphic Temporal Structures

Let  $\Sigma_T(X)$  denote the temporal information structure induced by an observational stream  $X$ . This structure includes: sampling rhythm, window hierarchy, aggregation strategy, and the ordering effects of buffering and ingestion.

Two indicators may be numerically identical while inducing distinct temporal structures. That is, for indicators  $X$  and  $Y$ , it may hold that:

$$X = Y, \quad \text{while} \quad \Sigma_T(X) \neq \Sigma_T(Y).$$

Numerical equality therefore does not imply temporal isomorphism.

**Definition 1.** Two indicators  $X$  and  $Y$  are temporally isomorphic if their temporal information structures  $ET(X)$  and  $ET(Y)$  are equivalent under a declared class of temporal abstractions.

Indicators that share a value may encode fundamentally different patterns of system behavior over time.

This distinction is essential for cross-layer inference. Without explicit alignment, aggregation operates on values while discarding the temporal structure that gives those values operational meaning.

## 4 Alignment Operators and Their Structural Cost

### 4.1 Alignment as Constructed Compatibility

Temporal alignment is often informally described as “bringing metrics onto a common timeline”. This description is misleading. Alignment does not reveal an underlying shared order; it constructs one.

Formally, alignment introduces a mapping

$$\alpha : \bigcup_i T_i \rightarrow T^*,$$

where  $T^*$  is a constructed temporal space chosen to support a particular class of inferences.

The choice of  $\alpha$  is therefore not neutral. It determines which temporal distinctions are preserved, which are collapsed, and which become unobservable.

Alignment is not a corrective operation. It is a modeling decision.

### 4.2 Classes of Alignment Operators

Common alignment strategies include:

- **Resampling to a common grid**, which enforces uniform spacing at the cost of temporal resolution.
- **Interpolation or carry-forward strategies**, which fabricate values to fill alignment gaps.
- **Window projection**, which maps observations into a hierarchy of fixed-duration intervals.
- **Event anchoring**, which aligns metrics relative to discrete reference events.
- **Logical-time alignment**, which replaces physical time with causal or sequence-based ordering.

Each strategy introduces a characteristic distortion. No alignment operator preserves all aspects of the original temporal structures.

### 4.3 Alignment Cost and Information Loss

Let  $\Sigma_T(X)$  denote the temporal information structure of an observational stream  $X$ . For an aligned stream  $X^* = \alpha(X)$ , in general:

$$\Sigma_T(X^*) \subsetneq \Sigma_T(X).$$

Alignment is therefore a lossy transformation.

Burst behavior may be smoothed. Short-lived failures may be absorbed into windows. Event orderings may be weakened or erased.

The practical implication is critical: alignment trades epistemic fidelity for cross-stream comparability.

This trade-off must be explicit. Absent explicit acknowledgment, alignment artifacts may be misinterpreted as properties of the observed system.

## 4.4 Alignment Cost and Distortion

Temporal alignment is not a neutral normalization step. It is an explicitly lossy transformation.

Any alignment operator  $\alpha$  imposes a chosen notion of temporal compatibility, and in doing so necessarily discards information present in the original observational streams.

Typical forms of distortion introduced by alignment include:

- **Aliasing**, where high-frequency behavior is folded into lower-frequency representations.
- **Loss of burst information**, where short-lived but operationally significant events disappear within aggregation windows.
- **Artificial smoothness**, where variability is reduced not because the system is stable, but because alignment suppresses it.
- **Boundary effects**, where events near window edges are reassigned or diluted depending on alignment choice.

Let  $X$  be an observational stream with temporal structure  $\Sigma_T(X)$ , and let  $X^* = \alpha(X)$  be its aligned representation. In general, the following relation holds:

$$\Sigma_T(X^*) \subsetneq \Sigma_T(X).$$

Alignment therefore constitutes a trade-off: it increases cross-stream comparability at the cost of epistemic fidelity.

This trade-off is unavoidable in distributed observability. However, when it is implicit, distortions introduced by alignment may be misinterpreted as properties of the observed system itself.

An aligned indicator that appears stable may owe its stability to the alignment operator, rather than to the underlying system behavior.

Alignment does not discover system behavior. It reshapes it for the purpose of inference.

## 5 Temporal Misalignment and the Illusion of Reliability

### 5.1 SLO/SLA as Temporally Constructed Assertions

Service-level objectives and agreements are frequently treated as factual properties of a system: “the service meets its availability target”.

In reality, an SLO or SLA is a statement about behavior over a constructed temporal domain. It presupposes:

- a declared observation window,
- a defined aggregation strategy,
- an implicit alignment of heterogeneous inputs.

When temporal alignment is implicit or inconsistent, the resulting SLO assertion lacks a stable referent.

## 5.2 False Confidence Through Window Compatibility

Consider two indicators used in an SLO computation: one derived from minute-level infrastructure sampling, the other from event-level service errors. If these streams are aligned through fixed windows without structural justification, the resulting SLO value may appear stable while concealing temporal inconsistency.

Short-lived but frequent failures may remain below window thresholds. High-impact events may be diluted by long aggregation periods.

The SLO is satisfied, yet user experience is degraded.

This condition produces false confidence: numerical compliance without temporal coherence.

## 5.3 SLA Disputes as Alignment Disputes

Many operational SLA disputes reduce, upon inspection, to disagreements about temporal construction:

- which time base was authoritative,
- which windows were applied,
- which alignment assumptions were implicit.

Such disputes cannot be resolved by examining scalar indicators alone. They require reconstruction of the temporal provenance of measurements.

Temporal misalignment thus transforms a technical monitoring issue into a contractual and organizational risk.

## 5.4 Alignment-Aware Reliability

Reliability claims are meaningful only relative to declared temporal structure.

An alignment-aware observability architecture does not eliminate uncertainty. It makes the sources of temporal distortion explicit and bounds their impact on inference.

Without such explicit construction, SLOs and SLAs function as indicators of compliance, not as guarantees of behavior.

# 6 A Minimal Formal Example

## 6.1 Equal Aggregates, Non-Equivalent Meaning

Consider three availability-like indicators derived from the same underlying system, but constructed under distinct temporal semantics:

- $A_I$ , computed from minute-level infrastructure liveness checks,
- $A_S$ , computed from ten-second service error windows,
- $A_B$ , computed from discrete business transaction outcomes.

Each indicator aggregates observations over a shared reporting interval, for example one month. Assume that all three yield the same scalar value:

$$A_I = A_S = A_B = 0.999.$$

Despite numerical equality, the informational content of these indicators differs fundamentally.

$A_I$  may represent a small number of isolated one-minute host failures, distributed uniformly over time.  $A_S$  may represent frequent short error bursts, each absorbed within ten-second windows.  $A_B$  may represent a few high-impact transaction failures, each corresponding to a discrete business event.

Let  $\Sigma_T(X)$  denote the temporal information structure induced by indicator  $X$ , including sampling rhythm, windowing strategy, and aggregation semantics.

Then, even under numerical equality, it holds that:

$$\Sigma_T(A_I) \neq \Sigma_T(A_S) \neq \Sigma_T(A_B).$$

The indicators are therefore not temporally isomorphic. They encode different patterns of system behavior, with different operational and experiential implications.

This example demonstrates that numerical comparability is insufficient for global inference. Without explicit temporal alignment and declared equivalence conditions, aggregated indicators may coincide in value while referring to non-equivalent behavioral histories.

Equal aggregates do not imply equal meaning.

In the absence of declared temporal equivalence, treating such indicators as interchangeable constitutes an underdetermined inference.

## 7 Failure Modes in Distributed Observability

### 7.1 False Stability

False stability arises when high-level indicators remain within acceptable bounds while the underlying system exhibits short-lived but operationally significant degradation.

This effect is typically introduced by window-based aggregation. Burst failures, latency spikes, or transient error storms may be absorbed into aggregation windows without materially affecting the reported metric.

As a result, the indicator suggests stability, not because the system is stable, but because temporal alignment suppresses variability.

False stability is particularly dangerous in capacity planning and reliability assessment, as it delays corrective action until degradation becomes persistent enough to survive aggregation.

### 7.2 Phantom Outages

Phantom outages occur when misaligned temporal pipelines produce contradictory indicators for the same period.

For example, infrastructure-level metrics may report host unavailability during a window in which service-level telemetry remains nominal, or vice versa. Ingestion delays, buffering, and differing window boundaries may cause events to be assigned to different reporting intervals.

From the operator's perspective, the system appears to both fail and succeed simultaneously.

Phantom outages are not caused by system behavior, but by incompatible temporal constructions. They complicate incident response by obscuring the distinction between real failures and alignment artifacts.

### 7.3 SLO/SLA Distortion

Service-level objectives and agreements are evaluated over constructed temporal domains. When the time bases and window semantics of contributing indicators are misaligned, SLO compliance becomes structurally unstable.

An objective may be met under one temporal interpretation and violated under another, even when derived from the same underlying events.

This distortion is not an accounting error. It is a consequence of implicit alignment assumptions. Without explicit temporal provenance, SLO and SLA assertions lack a stable referent and may provide a false sense of contractual compliance.

## 7.4 Cross-Team and Cross-Product Incomparability

In large organizations, availability and reliability indicators are often used for cross-team benchmarking and cross-product governance.

When identical percentages are derived from indicators with different temporal semantics, comparisons become misleading. Teams may appear more or less reliable based not on system behavior, but on differences in sampling rhythm, window definitions, or aggregation pipelines.

Such incomparability undermines governance mechanisms, distorts incentives, and erodes trust in shared metrics.

The root cause is not measurement inconsistency, but the absence of declared temporal equivalence across observability pipelines.

These failure modes are not independent. They emerge from the same root cause: implicit temporal alignment in heterogeneous observability pipelines.

# 8 Temporal Alignment and COE

## 8.1 GOP1 as a Global Orderability Requirement

In Coherent Observational Epistemology (COE), Global Orderability (GOP1) states that inter-local inference is legitimate only if observational sequences can be embedded into a shared ordering structure [1].

Temporal alignment provides a concrete operational interpretation of this requirement within distributed observability systems. Distinct monitoring pipelines generate observational streams over local time domains  $T_i$ , each inducing its own effective ordering. Absent an explicit construction of a shared temporal space  $T^*$ , these orderings remain incomparable.

From this perspective, temporal misalignment corresponds to a direct violation of GOP1. Cross-layer aggregation performed without verified temporal alignment assumes global orderability rather than constructing it.

Temporal alignment is therefore not an optimization, but a prerequisite for justified global inference.

## 8.2 Transformational Coherence and Pipeline Equivalence

COE further requires that transformations applied within observational localities admit coherent integration. This requirement is formalized as Transformational Coherence.

In distributed observability, metric pipelines apply sequences of temporal transformations: sampling, windowing, aggregation, and downsampling. Two pipelines may produce numerically comparable indicators while applying non-equivalent transformation chains.

Without declared pipeline equivalence, cross-local comparison becomes underdetermined. The resulting inference relies on implicit assumptions about the commutativity and compatibility of transformations that may not hold in practice.

Temporal alignment must therefore be accompanied by explicit declaration of transformation structure. Only when pipeline equivalence is established can aggregated indicators be treated as observations of the same system property.

In COE terms, alignment without transformational coherence produces apparent integration without epistemic justification.

## 9 Toward Alignment-Aware Observability Architectures

### 9.1 Design Principles

An alignment-aware observability architecture does not attempt to eliminate temporal heterogeneity. Instead, it treats temporal structure as a first-class concern.

The following principles characterize such an architecture.

- **Explicit time base declaration.** Each indicator must declare the temporal domain over which it is defined, including effective sampling rhythm and timestamp semantics (event time, observation time, or ingestion time).
- **Declared window hierarchy.** Aggregation windows must be explicitly specified as a hierarchy (e.g., 10 s, 1 min, 5 min, 1 h), rather than implicitly encoded in dashboards or queries. Window choice determines which behaviors are observable.
- **Transformation provenance tracking.** Indicators should carry metadata describing the sequence of temporal transformations applied to raw observations: sampling, windowing, aggregation, and downsampling. Without this provenance, numerical comparison lacks interpretative grounding.
- **Explicit alignment operators.** Cross-stream comparison must specify the alignment operator used to construct a shared temporal space. The operator's distortion characteristics and information loss must be acknowledged.
- **Declared equivalence conditions.** Indicators may be compared or aggregated only when their temporal structures are declared equivalent for the intended inference task.

These principles shift observability from implicit normalization to explicit temporal modeling.

### 9.2 Temporal Observability Contract

To operationalize these principles, we propose the notion of a *Temporal Observability Contract*.

A temporal observability contract specifies:

- the indicator's time domain and sampling semantics,
- the applied window hierarchy,
- the alignment mapping to a shared temporal space,
- the sequence of temporal transformations,
- acceptable bounds on distortion and information loss.

Such a contract does not constrain implementation details. Instead, it constrains interpretation.

By making temporal assumptions explicit, the contract enables:

- justified cross-layer aggregation,
- reproducible SLO/SLA evaluation,
- meaningful comparison across teams and products,
- traceable reasoning in incident analysis.

Alignment-aware observability architectures do not promise perfect fidelity. They promise that any loss of fidelity is explicit, bounded, and accounted for in downstream inference.

An indicator without a temporal contract is a value without a stable referent. The concrete instantiation of temporal observability contracts in organizational settings raises additional structural questions not pursued in this work.

## 10 Related Work

**Canonical Order and Operational Global Time.** Related work has explored the construction of global temporal order as an operational and canonical structure derived from locally registered event sequences, rather than as an intrinsic or absolute time coordinate. In this line of work, global time is defined as a protocol-dependent ordering constructed from reception sequences observed by local destinations, explicitly separating inaccessible source order from operationally admissible reception order [4], [5].

While the present paper remains within the domain of observability architectures and operational metrics, the two approaches are conceptually aligned. Temporal alignment in observability systems can be viewed as a restricted, task-oriented instance of canonical order construction, where global ordering is required not for physical coordination but for justified inter-local inference.

Importantly, both perspectives reject the assumption of an a priori global time. Instead, global order—where required—is treated as a constructed artifact, subject to explicit rules, distortion bounds, and declared equivalence conditions.

## 11 Conclusion

Temporal misalignment is not merely an implementation nuisance. It is a structural constraint on what can be legitimately inferred from heterogeneous observability pipelines.

This paper has shown that distributed observability systems operate over locally constructed temporal domains, shaped by sampling rhythms, window hierarchies, buffering, and aggregation strategies. When these domains are implicitly aligned, numerical indicators may appear stable and comparable while encoding non-equivalent temporal structures.

Such conditions give rise to characteristic failure modes: false stability, phantom outages, distorted SLO and SLA evaluations, and systematic incomparability across teams and products. These effects do not result from measurement error, but from undeclared assumptions about temporal compatibility.

Temporal alignment, when required, must therefore be treated as a modeling decision with explicit costs and declared equivalence conditions. Alignment increases comparability at the expense of epistemic fidelity, and this trade-off cannot be safely hidden behind scalar indicators.

Within the framework of Coherent Observational Epistemology, temporal misalignment corresponds to a failure to satisfy Global Orderability and Transformational Coherence. Cross-layer inference performed without verified alignment assumes conditions that have not been constructed.

Distributed observability does not require a single timeline. It requires that any shared temporal structure be explicitly built, bounded, and justified. Absent this construction, numerical stability remains an unreliable proxy for understanding.

More generally, the need for explicit construction of shared temporal structure suggests that global time, where required, cannot be assumed as a given coordinate, but must be operationally reconstructed from admissible ordering relations.

Temporal alignment determines not only what is measured, but what can be known.

## References

- [1] A. A. Nekludoff. “Coherent observational epistemology (coe): Foundational principles, secondary principles, and axiomatic system,” Zenodo, Accessed: Feb. 12, 2026. [Online]. Available: <https://doi.org/10.5281/zenodo.17632756>.
- [2] A. A. Nekludoff. “Unified availability model (uam): A normalization-based framework for measuring availability across heterogeneous information systems,” Zenodo, Accessed: Feb. 12, 2026. [Online]. Available: <https://doi.org/10.5281/zenodo.17721208>.
- [3] A. A. Nekludoff. “Availability without a model: On the semantic fragmentation of operational metrics,” Zenodo, Accessed: Feb. 12, 2026. [Online]. Available: <https://doi.org/10.5281/zenodo.18618639>.
- [4] A. A. Nekludoff, “Global time as a physical theory: Canonical order, operational time, and history immutability,” Feb. 2026. doi: 10.5281/zenodo.18662399. [Online]. Available: <https://doi.org/10.5281/zenodo.18662399>.
- [5] A. A. Nekludoff, “Ontology of global time: Local observers, sources and destinations,” Feb. 2026. doi: 10.5281/zenodo.18655183. [Online]. Available: <https://doi.org/10.5281/zenodo.18655183>.