

# Availability Without a Model: On the Semantic Fragmentation of Operational Metrics

**Alexey A. Nekludoff**

AstraVerge Research

E-mail: [an@astraverge.org](mailto:an@astraverge.org)  
ORCID: 0009-0002-7724-5762

12 February 2026

## **Abstract**

Availability is widely used as a primary managerial indicator in operational and distributed information systems. Despite its central role, the term often lacks a formally unified definition. In practice, availability indicators originate from heterogeneous observational layers, including infrastructure state, service behavior, and business-level functionality. These metrics are internally coherent within their respective domains, yet are frequently presented under a shared managerial label without an explicit cross-layer model.

This paper analyzes the structural consequences of such usage. We demonstrate that availability in many operational contexts functions as a layer-relative projection rather than as a formally derived system property. A minimal formalization is introduced to clarify the distinction between infrastructure state space, service behavior space, and business outcome space, and to show that scalar availability indicators may represent semantically divergent constructions.

The absence of an explicit mapping between observational layers produces semantic fragmentation: numerical stability is preserved, while conceptual comparability is not guaranteed.

The Unified Availability Model (UAM) is presented as a domain-specific example of model-based availability, demonstrating that cross-layer formalization is implementable. More generally, the analysis situates operational availability within the framework of Coherent Observational Epistemology (COE), treating monitoring systems as observational localities whose integration requires declared structural compatibility.

The paper argues that availability without a model is not merely a simplification, but a specific case of inter-local inference performed without explicit coherence construction.

**Keywords:** availability; operational metrics; monitoring systems; layered observability; semantic fragmentation; cross-layer aggregation; model-based availability; Unified Availability Model (UAM); Coherent Observational Epistemology (COE); Global Observational Postulates (GOP); SLO; SLA; enterprise IT systems.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Layered Origin of Operational Metrics</b>	<b>1</b>
2.1	Infrastructure Layer . . . . .	1
2.2	Service Layer . . . . .	2
2.3	Business or Functional Layer . . . . .	2
2.4	Layer Independence and Metric Autonomy . . . . .	2
2.5	Temporal Semantics and Observational Rhythm . . . . .	3
<b>3</b>	<b>Semantic Divergence Under a Unified Label</b>	<b>3</b>
3.1	Temporal Non-Alignment and the Failure of Global Orderability . . . . .	4
3.2	Illustrative Operational Scenarios . . . . .	4
3.3	A Minimal Formal Example of Informational Non-Isomorphism . . . . .	5
<b>4</b>	<b>A Minimal Formalization Attempt</b>	<b>5</b>
<b>5</b>	<b>Managerial Consequences of Model-Free Availability</b>	<b>6</b>
5.1	Architectural Consequences for Monitoring Systems . . . . .	6
5.2	Stability Without Semantic Transparency . . . . .	7
5.3	Incomparability Across Systems . . . . .	7
5.4	Decision-Making Under Implicit Assumptions . . . . .	7
5.5	Information Compression and Loss . . . . .	7
5.6	Availability as Convention . . . . .	7
<b>6</b>	<b>Toward Model-Based Availability</b>	<b>8</b>
6.1	From Projection to Relation . . . . .	8
6.2	Necessary and Sufficient Conditions . . . . .	8
6.3	Transparency and Comparability . . . . .	8
6.4	Scope and Limits . . . . .	8
<b>7</b>	<b>Relation to the Unified Availability Model</b>	<b>9</b>
7.1	Scope of UAM Within the Monitoring Ecosystem . . . . .	10
<b>8</b>	<b>Availability as a Special Case of Coherent Observational Epistemology</b>	<b>10</b>
8.1	Monitoring Systems as Observational Localities . . . . .	10
8.2	Semantic Fragmentation as Coherence Failure . . . . .	10
8.3	UAM as a Constrained Coherence Construction . . . . .	11
8.4	Availability and the Global Observational Postulates . . . . .	11
8.5	Monitoring Dashboards as Partial Observational Pipelines . . . . .	12
<b>9</b>	<b>Limitations and Scope Clarification</b>	<b>12</b>
<b>10</b>	<b>Conclusion</b>	<b>12</b>

# 1 Introduction

Availability occupies a central position in contemporary operational management. It functions as a primary indicator of system health, a contractual benchmark in service-level agreements, and a high-level signal for executive decision-making. Dashboards, reports, and incident reviews frequently converge on availability as the summary representation of system performance.

Despite its operational importance, availability often lacks a formally unified model of definition. In practice, the term is applied to metrics derived from heterogeneous monitoring layers, each capturing different aspects of system behavior. Infrastructure monitoring systems report resource states such as CPU load, memory utilization, or host liveness. Application-level monitoring systems capture request latency, error rates, and response codes. In some contexts, synthetic transactions or business-process probes are used to estimate end-to-end functionality.

These metrics are internally coherent within their respective layers. However, they are not necessarily semantically equivalent. A host may be technically alive while the service fails to execute business operations. Conversely, transient infrastructure degradation may not immediately affect observable service behavior. Nevertheless, indicators derived from these distinct observational domains are frequently labeled under the same managerial term: “availability.”

The absence of an explicit model relating infrastructure state, service behavior, and business functionality results in what may be termed semantic fragmentation. Availability becomes a label applied to different measurement constructs rather than the output of a unified computational framework. This fragmentation does not typically produce visible inconsistency at the dashboard level. Instead, it generates stable numerical signals whose semantic scope remains implicit.

The objective of this paper is not to critique specific tools or implementations. Rather, it aims to examine the structural conditions under which availability indicators emerge in real-world monitoring environments. By analyzing the layered origin of operational metrics and their managerial interpretation, we argue that availability in many organizational contexts functions without an explicit underlying model. This condition produces semantic divergence while preserving numerical stability, thereby shaping decision-making through formally consistent yet conceptually heterogeneous indicators.

## 2 Layered Origin of Operational Metrics

Operational monitoring environments typically emerge through incremental architectural evolution. As systems grow in complexity, different observational layers are instrumented independently, often by distinct tools and teams. The resulting monitoring landscape is therefore not monolithic but stratified.

At minimum, three observational layers can be distinguished.

### 2.1 Infrastructure Layer

The infrastructure layer captures the state of computational resources. Typical metrics include processor utilization, memory consumption, disk saturation, network reachability, and host liveness. These measurements describe the availability of resources required for computation. They reflect the operational condition of the environment in which services execute.

Infrastructure metrics answer questions such as:

- Is the host reachable?
- Are system resources within acceptable thresholds?
- Is the operating system responsive?

These indicators are necessary preconditions for service execution. However, they do not directly describe the success or failure of service-level operations.

## **2.2 Service Layer**

The service layer captures observable behavior of applications. Typical metrics include request latency, throughput, error rates, response status codes, and transaction durations. These measurements reflect how the system behaves under load and how it responds to client interactions.

Service-level metrics answer questions such as:

- Are requests processed successfully?
- What proportion of responses result in errors?
- How long does a transaction take to complete?

Unlike infrastructure metrics, these indicators describe externally observable behavior rather than internal resource conditions. They may degrade even when infrastructure remains technically operational.

## **2.3 Business or Functional Layer**

In some environments, an additional layer attempts to approximate business-level availability. Synthetic transactions, scenario-based probes, or end-to-end validation routines are used to determine whether key user-facing operations can be completed successfully.

These metrics answer questions such as:

- Can a user complete a critical workflow?
- Is a financial transaction successfully committed?
- Does an authentication sequence fully execute?

This layer most closely corresponds to managerial expectations of availability. However, it is also the most complex and least universally implemented.

## **2.4 Layer Independence and Metric Autonomy**

Each observational layer is internally coherent. Infrastructure metrics form a consistent model of resource health. Service metrics form a consistent model of behavioral performance. Business-layer metrics, where implemented, form a consistent model of functional execution.

Yet these layers are not reducible to one another. Infrastructure sufficiency does not guarantee service correctness. Service responsiveness does not guarantee business-level success. Conversely, localized infrastructure degradation may not immediately manifest in service-level failure.

Despite this structural independence, organizational dashboards frequently apply a common managerial label—“availability”—to indicators derived from any of these layers. The layered origin of the metrics remains implicit, while the terminology suggests conceptual unity.

This layered heterogeneity constitutes the structural foundation of semantic fragmentation. Availability, as presented in operational dashboards, is therefore not inherently a single construct. It is an emergent label attached to measurements originating from distinct and partially independent observational domains.

## 2.5 Temporal Semantics and Observational Rhythm

An additional source of semantic fragmentation lies in temporal structure.

Different monitoring systems may operate under distinct temporal models. Some rely on pull-based time-series databases with fixed scrape intervals. Others employ agent-based reporting with event-driven or buffered transmission. Sampling frequency, timestamp precision, aggregation windows, and interpolation strategies may vary significantly.

These differences are not merely technical. They define the ordering structure of observational sequences.

If availability is computed over heterogeneous temporal grids, two indicators may refer to different effective time domains: one reflecting minute-level volatility, another reflecting five-minute aggregated windows.

Without explicit temporal alignment, comparisons assume ordering compatibility that has not been formally established.

In COE terms, this corresponds to a potential violation of Global Orderability (GOP1), where alignment of observational rhythms is presupposed rather than constructed.

## 3 Semantic Divergence Under a Unified Label

**Definition 1.** Semantic fragmentation is the condition in which multiple observational functions produce scalar indicators that are numerically comparable but lack a declared structural mapping ensuring conceptual equivalence.

The existence of multiple observational layers does not, by itself, generate inconsistency. Each layer provides a valid and internally coherent description of system state within its own domain. The difficulty arises when indicators derived from these distinct domains are presented under a single managerial label: “availability.”

In operational practice, availability functions as a unifying concept. It appears in dashboards, reports, and service-level discussions as though it denotes a stable and well-defined property of a system. However, as shown in the previous section, the underlying measurements may originate from infrastructure conditions, service behavior, or business-level execution. The semantic content of the indicator therefore depends on its layer of origin.

This produces a form of semantic divergence. Two indicators may both be labeled “availability,” yet refer to fundamentally different constructs: one describing resource sufficiency, another describing application responsiveness, and a third describing functional completion of a business process.

The divergence remains largely invisible at the level of visualization. Numerical stability and graphical consistency create the appearance of conceptual unity. A green status light derived from infrastructure liveness and a green status light derived from successful transaction execution appear indistinguishable in managerial interfaces. The shared label suppresses the distinction between necessary conditions and sufficient conditions for service functionality.

This phenomenon can be described as semantic compression. Distinct observational meanings are compressed into a single managerial term. The compression does not falsify the underlying metrics. Each metric remains accurate within its own domain. What changes is the interpretative frame: the aggregation boundary implicitly asserts equivalence where only partial dependence exists.

The consequences are structural rather than accidental. Decision-making processes rely on the apparent stability of availability indicators. Yet without an explicit model linking infrastructure state, service behavior, and functional execution, the term “availability” operates as a convention rather than as the output of a formally defined relation.

Semantic divergence under a unified label therefore represents not an implementation flaw, but a modeling gap. Availability is treated as a property of the system, while in practice it is a layer-dependent measurement

whose meaning shifts across contexts. The absence of explicit cross-layer formalization permits this shift to remain unnoticed, thereby stabilizing managerial interpretation while preserving underlying heterogeneity.

In real-world environments, different subsystems often rely on distinct monitoring infrastructures with heterogeneous storage models, collection mechanisms, and temporal semantics. Time-series databases optimized for service metrics may coexist with agent-based infrastructure monitoring systems. While each environment ensures local coherence, no guarantee of cross-system structural alignment is implied.

### 3.1 Temporal Non-Alignment and the Failure of Global Orderability

Let  $T$  denote a global temporal domain. In practice, observational layers operate over distinct temporal subdomains:

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

Infrastructure monitoring may sample at fixed one-minute intervals. Service telemetry may operate at ten-second scrape intervals. Business events may be recorded asynchronously as discrete occurrences.

Formally:

$$O_I : \mathcal{I}(T_I) \rightarrow \mathbb{R}^n, \quad O_S : \mathcal{S}(T_S) \rightarrow \mathbb{R}^m, \quad O_B : \mathcal{B}(T_B) \rightarrow \mathbb{R}^k.$$

Unless an explicit alignment operator

$$\alpha : T_I \cup T_S \cup T_B \rightarrow T^*$$

is defined, these observational sequences do not admit guaranteed embedding into a shared ordering space.

Without verified temporal alignment, global aggregation assumes order compatibility that has not been formally established.

This condition directly corresponds to a violation of Global Orderability (GOP1) in COE terminology.

### 3.2 Illustrative Operational Scenarios

To clarify the structural argument, consider two simplified but realistic scenarios.

**Scenario A: Infrastructure-Available, Functionally-Unavailable.** A distributed service runs on multiple hosts. All infrastructure metrics indicate normal operation: CPU usage remains within threshold, memory consumption is stable, network interfaces are reachable, and host liveness checks return positive results. The infrastructure-derived availability indicator therefore reports 100%.

However, due to an internal application-layer misconfiguration, all user transactions fail at the business-logic level. From the perspective of functional execution, the service is unavailable.

Both indicators are internally correct. The divergence arises because they refer to different observational domains.

**Scenario B: Equal Percentages, Different Constructions.** Two products report 99.95% availability. Product A computes availability based on the proportion of failed business transactions relative to total requests. Product B computes availability as the proportion of reachable hosts over a fixed monitoring interval.

The numerical values coincide, yet the semantic constructions differ. Without an explicit model specifying equivalence conditions, the indicators are numerically comparable but structurally non-identical.

### 3.3 A Minimal Formal Example of Informational Non-Isomorphism

Consider three availability indicators:

- $A_I$  computed over minute-level infrastructure liveness,
- $A_S$  computed over ten-second service error ratios,
- $A_B$  computed over discrete business transaction events.

Suppose that over a 30-day interval:

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

Despite numerical equality, the informational structures differ.

$A_I$  may represent 43 isolated one-minute host failures.  $A_S$  may represent multiple short error bursts aggregated within ten-second windows.  $A_B$  may represent a small number of high-impact failed transactions.

Let  $\Sigma(X)$  denote the informational structure of indicator  $X$ . Then:

$$\Sigma(A_I) \neq \Sigma(A_S) \neq \Sigma(A_B),$$

even when:

$$A_I = A_S = A_B.$$

Numerical equivalence does not imply structural isomorphism.

## 4 A Minimal Formalization Attempt

The preceding sections argued that availability is frequently applied to measurements originating from distinct observational layers. We now introduce a minimal formal structure to clarify the relationships between these layers.

Let us define three state spaces:

- $\mathcal{I}$  — the infrastructure state space,
- $\mathcal{S}$  — the service behavior state space,
- $\mathcal{B}$  — the business or functional state space.

An element  $i \in \mathcal{I}$  may represent a vector of resource conditions (CPU, memory, network reachability, host liveness). An element  $s \in \mathcal{S}$  may represent observable service behavior (latency distributions, error ratios, throughput). An element  $b \in \mathcal{B}$  may represent functional outcomes (success or failure of critical workflows).

Operational metrics define partial observation functions:

$$O_I : \mathcal{I} \rightarrow \mathbb{R}^n, \quad O_S : \mathcal{S} \rightarrow \mathbb{R}^m, \quad O_B : \mathcal{B} \rightarrow \mathbb{R}^k.$$

Each observation function is internally coherent within its layer. However, no canonical mapping between these state spaces is guaranteed.

In practice, availability is often introduced as a scalar indicator:

$$A = f(O_X(\cdot)),$$

where  $X \in \{I, S, B\}$  denotes the layer of origin, and  $f$  is an aggregation or thresholding function.

The critical point is that  $A$  may be computed from different domains without explicit formal linkage between  $\mathcal{I}$ ,  $\mathcal{S}$ , and  $\mathcal{B}$ . The same symbol  $A$  is therefore used for distinct constructions:

$$A_I = f_I(O_I(i)), \quad A_S = f_S(O_S(s)), \quad A_B = f_B(O_B(b)).$$

Nothing in the general operational practice ensures that

$$A_I = A_S = A_B,$$

nor that any of these implies the others.

At most, partial dependency relations may exist. For example, one may postulate a structural dependence:

$$i \in \mathcal{I} \text{ is a necessary condition for } s \in \mathcal{S},$$

but necessity does not imply sufficiency. Infrastructure viability does not entail correct service behavior, and correct service behavior does not entail successful completion of business-level workflows under all conditions.

Without an explicit cross-layer mapping

$$\Phi : \mathcal{I} \times \mathcal{S} \times \mathcal{B} \rightarrow \{0, 1\},$$

availability remains layer-relative. The indicator  $A$  becomes a projection of a higher-dimensional state space onto a scalar value without a formally defined aggregation model.

This minimal formalization clarifies the structural issue: availability is frequently treated as a property of the system, while in fact it is a layer-dependent functional over incomplete observational domains.

In the absence of an explicit inter-layer model, the equality of availability indicators across contexts is assumed rather than derived.

## 5 Managerial Consequences of Model-Free Availability

The absence of an explicit cross-layer model does not typically produce immediate operational failure. On the contrary, availability indicators often appear stable, consistent, and actionable. This apparent stability explains their persistence.

However, model-free availability has structural consequences for decision-making.

### 5.1 Architectural Consequences for Monitoring Systems

Semantic fragmentation has concrete architectural implications:

- **SLO/SLA instability:** Objectives may be defined over layer-relative indicators without formal equivalence conditions.
- **False stability signals:** Aggregated availability may remain high while layer-specific degradation accumulates.
- **Distorted Root Cause Analysis (RCA):** Layer ambiguity complicates causal tracing across infrastructure, service, and business domains.

- **Cross-team incomparability:** Products reporting identical availability values may rely on fundamentally different constructions.

These effects do not arise from measurement error, but from structural misalignment between observational layers.

## 5.2 Stability Without Semantic Transparency

Scalar indicators derived from layer-relative observations produce numerically stable signals. Dashboards display clear values, thresholds trigger alerts, and service-level summaries appear coherent.

Yet the semantic origin of the indicator remains implicit. When  $A_I$ ,  $A_S$ , or  $A_B$  are presented simply as “availability,” the decision context assumes equivalence across products and systems, even when the underlying constructions differ.

This creates stability at the interface level while preserving heterogeneity at the structural level.

## 5.3 Incomparability Across Systems

If availability is computed from infrastructure metrics in one system and from service-level metrics in another, comparisons become formally underdetermined.

Two systems may both report 99.9% availability, yet refer to distinct observational domains. Without a shared model linking  $\mathcal{I}$ ,  $\mathcal{S}$ , and  $\mathcal{B}$ , such comparisons lack semantic grounding.

The indicator retains numerical comparability but loses conceptual comparability.

## 5.4 Decision-Making Under Implicit Assumptions

In the absence of explicit formalization, managers implicitly assume that availability represents a unified system property. Resource allocation, prioritization, and escalation decisions are therefore guided by indicators whose construction is not transparently defined.

The issue is not inaccuracy of measurement, but indeterminacy of interpretation. An availability value may be correct relative to its layer, yet incomplete relative to system functionality.

## 5.5 Information Compression and Loss

Scalar availability indicators compress multidimensional state spaces into a single value. Compression is necessary for managerial usability. However, without a declared model of aggregation, this compression may obscure which layer contributes to observed degradation.

The result is not misinformation, but structured information loss: distinct causal domains are merged into an undifferentiated signal.

## 5.6 Availability as Convention

When availability lacks explicit cross-layer definition, it functions as a managerial convention. Its authority derives from repetition and institutional embedding, rather than from formal derivation.

Such conventions can remain operationally effective, but they limit analytical clarity. The system appears measurable, yet the meaning of what is measured remains partially implicit.

## 6 Toward Model-Based Availability

The preceding analysis suggests that the central issue is not measurement accuracy, nor tool selection, nor implementation detail. It is the absence of an explicit model linking observational layers.

If availability is to function as a coherent managerial indicator, it must be defined as the result of a declared relation between infrastructure state, service behavior, and business functionality. In other words, availability must be computed from an articulated model, rather than inferred from layer-local projections.

### 6.1 From Projection to Relation

In the model-free setting, availability is a projection:

$$A_X = f_X(O_X(\cdot)),$$

where  $X$  denotes a single observational layer. Such projections are valid within their domain, but they do not capture cross-layer dependence.

A model-based approach instead defines availability as a relation:

$$A = \Psi(i, s, b),$$

where  $i \in \mathcal{I}$ ,  $s \in \mathcal{S}$ ,  $b \in \mathcal{B}$ , and  $\Psi$  explicitly encodes dependency structure.

The essential shift is conceptual: availability becomes a function over a structured state space, rather than a scalar extracted from a single measurement domain.

### 6.2 Necessary and Sufficient Conditions

A model-based definition clarifies the distinction between necessary and sufficient conditions.

Infrastructure viability may be necessary for service functionality, but not sufficient. Service responsiveness may be necessary for business completion, but not sufficient. Explicit modeling forces these relations to be stated, rather than assumed.

This prevents semantic collapse by preserving the layered structure within the computation itself.

### 6.3 Transparency and Comparability

When availability is derived from an explicit inter-layer model, its meaning becomes portable across systems. Two availability indicators become comparable only if they are derived from formally equivalent constructions.

Model-based availability therefore restores conceptual comparability to numerical comparability. The scalar value regains interpretative grounding.

### 6.4 Scope and Limits

The purpose of advocating a model-based approach is not to increase complexity unnecessarily. Operational environments require compression and abstraction. However, compression should follow formal aggregation rules, not precede them.

Availability cannot be eliminated as a managerial concept. It is too deeply embedded in operational practice. What can be reconsidered is its construction: from a conventionally applied label to a formally derived indicator.

A system with model-based availability does not eliminate uncertainty. It makes the source and structure of that uncertainty explicit.

## 7 Relation to the Unified Availability Model

The analysis presented in this paper identifies a structural gap: availability is frequently treated as a managerial label applied to heterogeneous operational metrics without an explicit cross-layer formalization.

The Unified Availability Model (UAM) [2] provides an example of how such formalization can be constructed.

UAM explicitly distinguishes:

- raw metrics  $M_{ij}$ ,
- normalized metrics  $N_{ij} \in [0, 1]$ ,
- system-level availability  $A_i$ ,
- contour-level availability  $A_{\text{contour}}$ .

Unlike model-free approaches, UAM requires:

1. declared normalization rules for each metric class,
2. explicit weight coefficients,
3. transparent aggregation formulas,
4. hierarchical coherence constraints between levels.

In the terminology introduced earlier in this paper, UAM defines an explicit mapping:

$$\Phi : \mathcal{I} \times \mathcal{S} \times \mathcal{B} \rightarrow [0, 1],$$

thereby replacing layer-relative projections with a formally constructed availability relation.

Importantly, UAM does not assume homogeneity of metric spaces. Instead, it acknowledges heterogeneity as a structural property of enterprise systems and resolves it through normalization operators and weighted composition.

This formalization eliminates semantic fragmentation in two ways:

- Availability is no longer a label applied independently at different observational layers.
- Cross-system comparability is restored through declared construction rules rather than implicit convention.

The significance of UAM in the context of this paper is not merely practical. It demonstrates that the absence of a model is not inevitable. Availability can be defined as a formally derived indicator over heterogeneous state spaces, provided that normalization and aggregation rules are made explicit.

Thus, the transition from model-free availability to model-based availability is not a theoretical abstraction, but an implementable architectural shift.

## 7.1 Scope of UAM Within the Monitoring Ecosystem

UAM does not attempt to replace monitoring systems. It operates at the aggregation layer.

Infrastructure collectors, service telemetry systems, and business probes remain independent observational localities. UAM introduces normalization and weighting operators that act upon already stabilized metric sequences.

The model is therefore applicable to heterogeneous information systems where:

- metrics can be normalized into bounded intervals,
- hierarchical relationships between components are declared,
- aggregation rules can be formally specified.

UAM does not eliminate the need for local coherence. It presupposes it. Its contribution lies in constructing an explicit cross-layer aggregation relation.

Thus, UAM should be understood as a structured intermediate layer between raw observability and managerial representation.

## 8 Availability as a Special Case of Coherent Observational Epistemology

The preceding sections framed availability as a layer-dependent indicator whose meaning depends on the structure of observational aggregation. This problem is not unique to operational monitoring. It belongs to a broader class of epistemic situations in which heterogeneous observational localities are integrated without explicit structural alignment.

Coherent Observational Epistemology (COE) [1] provides a general methodological framework for analyzing such situations.

### 8.1 Monitoring Systems as Observational Localities

In COE terminology, each monitoring subsystem constitutes an observational locality:

- Infrastructure monitoring systems generate sequences within  $\mathcal{I}$ .
- Service-level monitoring systems generate sequences within  $\mathcal{S}$ .
- Business or synthetic probes generate sequences within  $\mathcal{B}$ .

Each locality maintains its own fixation rules, ordering structures, transformation pipelines, and operational semantics.

From the COE perspective, the problem identified in this paper corresponds to the absence of formally established inter-local coherence.

### 8.2 Semantic Fragmentation as Coherence Failure

The semantic divergence of availability indicators can be interpreted through COE axioms:

- Lack of explicit ordering compatibility (cf. Structural Axiom S1).
- Absence of declared transformation mapping between layers (S2).
- Semantic non-commensurability of layer-specific constructs (S3).

- Aggregation without preservation of structural information (S5).

In this light, model-free availability corresponds to a partial observational pipeline in which Global Alignment and Global Ordering are assumed rather than constructed.

The scalar availability indicator functions as a supra-local signal without explicit verification of the structural conditions required for coherent integration.

### 8.3 UAM as a Constrained Coherence Construction

The Unified Availability Model (UAM), discussed in the previous section, can therefore be interpreted as a domain-specific instantiation of COE principles.

UAM explicitly:

- normalizes heterogeneous metrics,
- defines aggregation operators,
- preserves hierarchical structure,
- documents transformation rules.

In COE terms, UAM attempts to construct a legitimate global observational structure by satisfying minimal coherence constraints within the operational domain.

Thus:

- COE provides the epistemological architecture,
- this paper diagnoses a specific coherence failure within operational monitoring,
- UAM provides a constrained engineering realization of model-based availability.

Availability without a model is therefore a concrete manifestation of a more general phenomenon: inter-local inference performed without verified structural compatibility.

### 8.4 Availability and the Global Observational Postulates

The structural gap identified in this paper can be made more explicit by interpreting availability construction through the lens of the Global Observational Postulates (GOP) introduced in COE [1].

Under GOP1 (Global Orderability), cross-local inference requires that observational sequences admit embedding into a shared ordering space. In operational monitoring environments, infrastructure metrics, service-level metrics, and business probes often evolve independently. Dashboards aggregate their outputs without verifying whether their orderings admit a formally constructed supra-local alignment.

Under GOP2 (Transformational Coherence), locality-specific transformations must admit a shared factorization. In model-free availability, normalization and thresholding are typically performed within each monitoring subsystem, yet no explicit cross-layer transformation mapping is declared.

Under GOP3 (Semantic Liftability), operational definitions must admit non-contradictory translation into a shared semantic space. When infrastructure liveness and service responsiveness are both labeled “availability,” semantic liftability is presumed rather than demonstrated.

Under GOP5 (Integrative Sufficiency), no locality-specific structural information essential to the phenomenon should be discarded in the construction of a global indicator. Scalar availability projections frequently compress layer-distinct information into a single value, thereby satisfying usability requirements while leaving structural preservation unspecified.

From this perspective, model-free availability corresponds to a situation in which a supra-local indicator is constructed without explicit verification of GOP1–GOP5.

## 8.5 Monitoring Dashboards as Partial Observational Pipelines

COE describes the formation of cross-local evidence as a structured observational pipeline:

Locality → Fixation → Local Ordering → Global Alignment → Global Ordering → Integrability → Inference.

In many operational monitoring environments, the early stages of this pipeline are satisfied locally: each subsystem fixes, orders, and stabilizes its own metrics.

However, the stages of Global Alignment and Global Ordering are often implicit. Dashboards present aggregated indicators without a formally constructed alignment space between infrastructure, service, and business-level observations.

The availability indicator thus functions as if global alignment had been established, while in practice only local coherence is guaranteed.

This condition does not imply methodological error. Rather, it indicates that the observational pipeline remains partial. Global inference is performed on the basis of layer-relative projections, without explicit verification of structural compatibility across localities.

In this sense, availability without a model is structurally analogous to inter-local inference performed prior to confirmed global observational construction.

## 9 Limitations and Scope Clarification

The argument developed in this paper does not advocate the replacement of layered monitoring architectures with a single scalar indicator. Complex systems require stratified observability, and domain-specific metrics remain indispensable for diagnosis, control, and engineering intervention.

The critique presented here concerns a specific condition: the use of a unified managerial availability indicator without an explicit cross-layer construction model.

Operational environments may legitimately employ multiple indicators corresponding to infrastructure, service behavior, and business functionality. These indicators need not be collapsed into one value.

However, when organizational decision-making requires a single summary signal, its construction cannot be reduced to a minimal liveness probe or a layer-local threshold. A unified indicator, if introduced, must be defined through declared normalization rules, aggregation structure, and dependency relations between observational layers.

The purpose of this paper is therefore not prescriptive with respect to the number of indicators, but structural with respect to their construction. It identifies the conditions under which a scalar availability indicator becomes conceptually grounded, and distinguishes them from cases where numerical stability masks semantic indeterminacy.

Model-based availability is thus presented as a requirement only in contexts where cross-layer aggregation is intended. Where no such aggregation is performed, the critique does not apply.

## 10 Conclusion

This paper examined availability as it appears in operational practice and identified a structural gap: the frequent use of a unified managerial label in the absence of an explicit cross-layer model.

We showed that availability indicators commonly originate from heterogeneous observational layers— infrastructure state, service behavior, or business-level functionality— without formally declared relations between them. Such usage produces semantic fragmentation: numerically stable indicators whose conceptual content depends on their layer of origin.

A minimal formalization demonstrated that layer-relative projections cannot substitute for an explicit relational construction. Without a defined mapping between  $\mathcal{I}$ ,  $\mathcal{S}$ , and  $\mathcal{B}$ , availability remains a convention rather than a derived system property.

The Unified Availability Model (UAM) illustrates that model-based availability is implementable. By introducing normalization rules, declared aggregation operators, and hierarchical coherence constraints, UAM replaces implicit projection with explicit construction.

More generally, this analysis constitutes a specific instance of the broader framework of Coherent Observational Epistemology (COE). Within COE terminology, operational monitoring systems function as observational localities. Model-free availability corresponds to partial or unverified inter-local integration. Model-based availability represents a constrained construction of a global observational structure under declared compatibility conditions.

Availability without a model is therefore not merely a technical simplification. It is a particular case of inter-local inference performed without explicit coherence verification.

The present analysis may therefore be understood as a three-level structure:

- Coherent Observational Epistemology (COE) defines the general conditions for legitimate inter-local inference.
- The current paper identifies a concrete coherence failure within operational monitoring, manifested as semantic fragmentation of availability.
- The Unified Availability Model (UAM) represents a domain-specific engineering attempt to satisfy minimal coherence constraints.

Availability without a model is thus not an isolated operational imperfection, but a particular instance of inter-local inference performed without verified structural alignment.

Recognizing this condition does not invalidate existing monitoring practice. Rather, it clarifies the epistemic status of availability indicators and establishes the structural criteria under which they become conceptually comparable and methodologically transparent.

Structured industry methodologies such as SLI/SLO frameworks, error-budget policies, and RED/USE monitoring approaches provide internal coherence within specific layers. However, unless cross-layer alignment is explicitly declared, these methodologies remain layer-relative. They improve local observability, but do not by themselves guarantee semantic integration across infrastructure, service, and business domains.

## References

- [1] A. A. Nekludoff, *Coherent observational epistemology (coe): Foundational principles, secondary principles, and axiomatic system*, Zenodo, 2025. doi: 10.5281/zenodo.17632756 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, 2025. doi: 10.5281/zenodo.17721208 Accessed: Feb. 12, 2026. [Online]. Available: <https://doi.org/10.5281/zenodo.17721208>