The Observability Ambiguity Problem:
Multi-Level Observability for Operational AI Systems

Deepinder Sidhu
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
sidhu@umbc.edu

Abstract

Observability has emerged as a central concept in agentic artificial intelligence, AI governance, cybersecurity, and enterprise operations. Vendors, analysts, and practitioners increasingly claim that operational AI systems require observability. Despite the importance of the term, observability is often used without specifying the system, state space, or abstraction level being observed. As a result, two platforms may make identical observability claims while exposing fundamentally different aspects of system behavior. We refer to this as the *Observability Ambiguity Problem*. This paper argues that observability is not a singular capability. Rather, observability is a relationship between an observer and a state space. Operational AI systems contain multiple state spaces, including agent state, workflow state, mission state, and enterprise state. Consequently, meaningful observability claims must identify what is being observed. We introduce a multi-level observability framework for operational AI systems and distinguish agent observability, workflow observability, mission observability, and enterprise observability. We also introduce the distinction between logical workflows and physical workflows and describe workflow conformance as the relationship between intended behavior and actual execution evidence. The paper's central message is simple: if vendors claim observability, they should specify exactly what they observe.

Keywords: Observability, Operational AI, Agentic AI, AI Governance, Mission Assurance, Workflow Conformance, Enterprise Observability, Operational AI Engineering

1 Introduction

Observability has become one of the most frequently invoked requirements for modern AI systems. The term appears in discussions of AI governance, agentic AI, cybersecurity operations, model monitoring, runtime inspection, compliance, and trust management. As AI systems move from passive analytical tools to active participants in workflows, organizations must understand what these systems are doing, why they are doing it, and how their actions affect operational outcomes.

However, the term *observability* is often used as though it refers to a single property. A vendor may claim that its platform provides observability. An analyst may recommend that agentic AI requires observability. A government program may state that AI-enabled systems must be observable. Yet such statements are incomplete unless they specify what is being observed.

This paper defines the ambiguity created by unspecified observability claims as the *Observability Ambiguity Problem*. The purpose is not to argue against observability; it is to argue for precision. If vendors claim observability, they should specify exactly what they observe.

1.1 Contributions

This paper makes three contributions. First, it defines the Observability Ambiguity Problem as a problem that arises when observability is claimed without specifying the system, state space, or abstraction level to which the claim applies. Second, it introduces a multi-level observability framework for operational AI systems spanning agent, workflow, mission, and enterprise state spaces. Third, it introduces workflow conformance as a way to relate logical workflow specifications to observed physical execution evidence.

2 Related Work

Observability originated in control theory, where it describes the ability to infer the internal state of a dynamical system from external measurements. In software engineering and distributed systems, observability has evolved to encompass logs, metrics, traces, and runtime telemetry that enable operators to understand system behavior. More recently, observability has become a central concept in cloud computing, AIOps, cybersecurity operations, and AI-enabled systems [4, 6, 1].

Contemporary AI governance frameworks, including the NIST AI Risk Management Framework, emphasize monitoring, transparency, accountability, and continuous oversight of AI systems [5]. Industry guidance similarly identifies observability as a critical requirement for trustworthy AI operations [2, 3]. However, existing literature generally treats observability as a singular capability without explicitly distinguishing the multiple state spaces present in operational AI systems.

This paper focuses on a different problem. Rather than asking whether a system is observable, it asks what state space is being observed. We argue that operational AI systems contain multiple distinct state spaces—including agent, workflow, mission, and enterprise state spaces—and that observability claims become ambiguous when the observed state space is not explicitly identified. This ambiguity forms the basis of the Observability Ambiguity Problem introduced in this work.

The primary contribution of this paper is the identification and formalization of the Observability Ambiguity Problem. We argue that observability is not a singular system property but a relationship between an observer and a specific state space. Because operational AI systems contain multiple state spaces operating at different levels of abstraction, claims of observability are incomplete unless the observed state space is explicitly identified. Building on this observation, we introduce a multi-level observability framework for operational AI systems and define distinctions between agent observability, workflow observability, mission observability, and enterprise observability.

3 The Observability Ambiguity Problem

Definition 1 (Observability Ambiguity Problem).

> *The Observability Ambiguity Problem arises when a system is claimed to be observable without specifying the system, state space, or abstraction level to which the claim applies.*

The statement "our platform provides observability" is incomplete. A meaningful claim should answer: What system is being observed? What state space is observable? At what abstraction level is observability provided? What remains hidden? Can state be reconstructed from the available observations?

For agentic AI systems, at least four state spaces commonly arise:

$$\{S_A,\ S_W,\ S_M,\ S_E\}$$

where

$$
\begin{aligned}
S_A &: \text{ Agent State} \\
S_W &: \text{ Workflow State} \\
S_M &: \text{ Mission State} \\
S_E &: \text{ Enterprise State}
\end{aligned}
$$

These state spaces are conceptual abstractions used to distinguish different levels of operational reasoning and are not intended to imply a specific mathematical representation.

4 Observability and State Spaces

We use the framing:

$$\text{Observability} = f(\text{Observer}, \text{System}, \text{State Space}, \text{Abstraction Level}).$$

Observability is not a free-standing property. It is a relationship. To evaluate an observability claim, one must know the state space being observed.

5 Multi-Level Observability

Operational AI systems are layered systems. They involve models, agents, workflows, missions, and enterprises. Consequently, observability must also be layered.

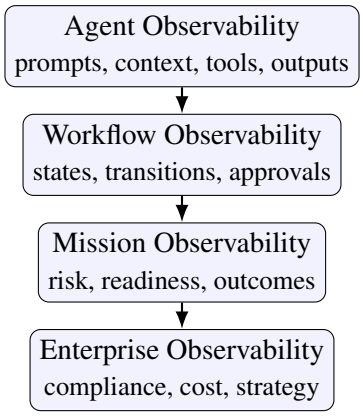

Figure 1: Multi-level observability in operational AI systems. Each layer observes a different state space and answers different questions.

5.1 Agent Observability

Agent observability concerns prompts, retrieved context, tool calls, intermediate actions, outputs, policy checks, and safety events. It answers: What did the agent do? What information did it use? What tool did it call? Was a policy violated?

5.2 Workflow Observability

Workflow observability concerns workflow state, state transitions, agent interactions, messages, dependencies, human approvals, retries, failures, and branch decisions. It answers: What is the workflow doing? Which state is active? Where is execution blocked? Was human approval required and obtained? This is system-level observability.

5.3 Mission Observability

Mission observability concerns mission objectives, mission risk, readiness, dependency impacts, operational performance, and mission outcomes. It answers: Did the workflow improve mission readiness? Did risk decrease? Did the mission succeed?

5.4 Enterprise Observability

Enterprise observability concerns compliance posture, cost, workforce impact, strategic objectives, enterprise risk, and policy adherence. It answers: Did the organization benefit? Did compliance improve? Did enterprise risk decrease?

6 Logical and Physical Workflows

A logical workflow describes intended behavior. It is a specification, for example:

$$\text{Knowledge Agent} \rightarrow \text{Reasoning Agent} \rightarrow \text{Human Approval} \rightarrow \text{Execution Agent}.$$

A physical workflow describes actual execution evidence: messages, logs, traces, API calls, packets, tool invocations, timestamps, and resource usage. The physical workflow is analogous to a packet capture in networking. It is evidence of execution, not the logical protocol itself.

$$\text{Logical Flow} \neq \text{Physical Flow}.$$

7 Workflow Conformance

Workflow conformance asks whether physical execution conforms to the logical workflow specification. Suppose the logical workflow requires

$$K \rightarrow R \rightarrow H \rightarrow E$$

where K is knowledge gathering, R is reasoning, H is human approval, and E is execution. If the observed physical workflow is

$$K \rightarrow R \rightarrow E,$$

then the human approval step has been bypassed. This is a workflow conformance failure. This notion is analogous to protocol conformance testing.

8 Evaluating Observability Claims

When a platform claims observability, ask:

1. What system is being observed?

2. What state space is observable?

3. What abstraction level is observable?

4. What physical evidence is collected?

5. Can logical state be reconstructed?

6. Can workflow conformance be verified?

7. What state remains hidden?

8. What is inferred rather than directly observed?

9. What mission or enterprise outcomes are observable?

This checklist transforms observability from a marketing term into an engineering property.

9 Observability About Observability

Observability claims themselves should be auditable. If a platform claims observability, an evaluator should be able to determine the scope and limits of that observability: which state spaces are observable, which measurements are available, which inferences are required, which states remain hidden, and what level of reconstruction is possible. In this sense, observability claims should themselves be observable.

10 Implications for Trustworthy Operational AI

Trustworthy operational AI asks not only whether an agent behaved correctly, but whether the overall mission workflow remained governed, observable, compliant, and effective.

$$\text{Agent Correctness} \not\Rightarrow \text{Mission Success}$$

$$\text{Agent Observability} \not\Rightarrow \text{Mission Observability}$$

Therefore, agent observability alone is insufficient. Operational AI systems require multi-level observability.

11 Conclusion

This paper introduced the Observability Ambiguity Problem. The ambiguity arises when observability is claimed without specifying the system, state space, or abstraction level being observed. Observability is not singular. Operational AI systems contain multiple state spaces, including agent state, workflow state, mission state, and enterprise state. Each gives rise to a distinct form of observability. The central message is straightforward: if vendors claim observability, they should specify exactly what they observe.

The contribution of this paper is not a new observability mechanism, but a new way of reasoning about observability claims in operational AI systems. By explicitly identifying state spaces, organizations can compare observability platforms, evaluate governance claims, and reason about assurance at multiple levels of abstraction.

References

[1] Betsy Beyer, Chris Jones, Jennifer Petoff, and Niall Murphy. *Site Reliability Engineering*. O'Reilly, 2016.

[2] Gartner. Ai governance requires more than policies, 2025.

[3] Gartner. Market guide for ai trust, risk and security management, 2025.

[4] Rudolf Kalman. On the general theory of control systems. *Proceedings First International Congress of Automatic Control*, 1960.

[5] National Institute of Standards and Technology. Ai risk management framework (ai rmf 1.0), 2023.

[6] OpenTelemetry Project. Opentelemetry documentation, 2024.