Observability for AI systems: tracing LLM calls, tokens and drift

26 October 2026 · 5 min read · blogs.iictp.org/observability-for-ai-systems

A language-model application fails in ways ordinary monitoring does not see: the service is up and fast, yet the answers are poor. Observability for AI adds visibility of what went in, what came out and what it cost.

TL;DR The short version

- An AI application can be up and fast yet give poor answers, which ordinary monitoring does not see.
- Trace each model call (model, settings, latency, tokens) and each retrieval step.
- Measure cost and quality signals such as feedback, retries and refusals, and protect the personal data prompts may contain.

Read online, listen to it, or change the text size: https://blogs.iictp.org/observability-for-ai-systems

What to trace

- Each model call: model, settings, latency and token counts.
- Retrieval steps: what was fetched and how relevant it was.
- Tool calls and their results.
- The full chain for one user request, as a single trace.

What to measure

- Cost per request and per feature.
- Quality signals: feedback, retries, refusals, evaluation scores.
- Drift: quality changing after a model, prompt or data change.

Privacy

Prompts and outputs can contain personal data. Decide what to store, mask sensitive fields and limit access and retention. OpenTelemetry has emerging conventions for generative-AI telemetry; check their current status before depending on them.

Related credential: Clean Software Telemetry Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. OpenTelemetry. Semantic conventions for generative AI. opentelemetry.io/docs/specs/semconv/gen-ai/.

Published by IICTP, an independent technical credential body, on 26 October 2026. The current version of this post is at https://blogs.iictp.org/observability-for-ai-systems. More guides: blogs.iictp.org.