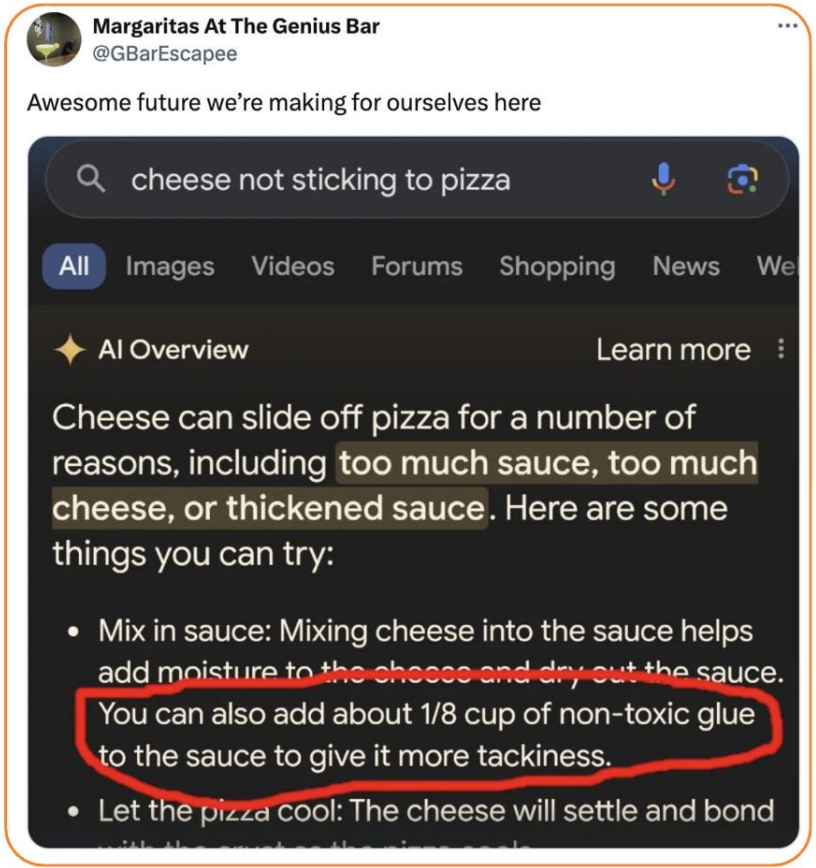
KubeCon CloudNativeCon India 2026

What Did My Agent Do? Observability and Accountability for AI Agents

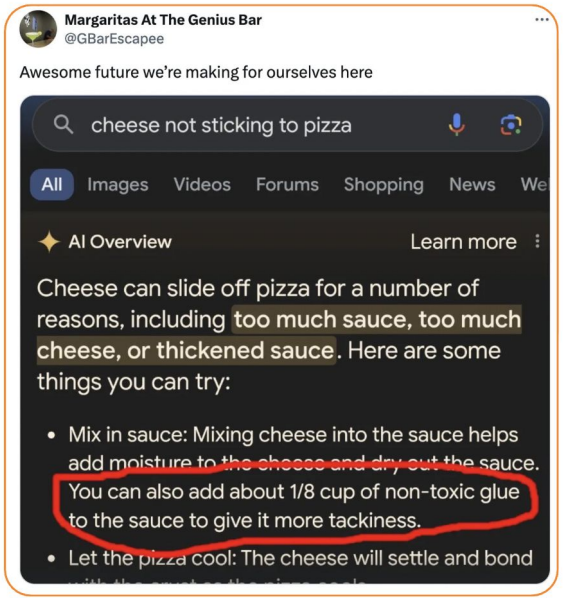
Ishan Jain
Grafana Labs

Why Observability?

- **LLMs are Black Boxes**
- **Cost & Performance Issues**
- **Hallucinations & Trust Issues**



What went wrong?

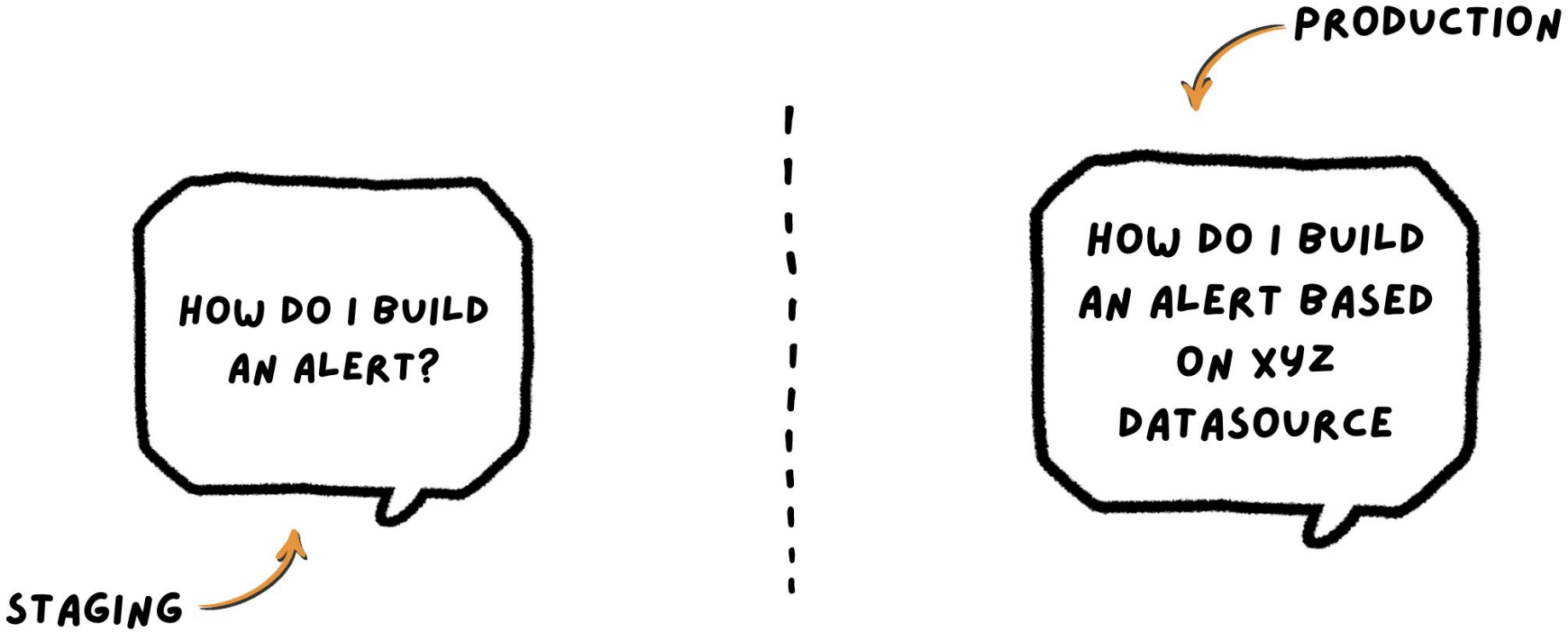


- Wrong Context?
- LLM hallucinated?
- Bad chunking?
- Bad system prompt?

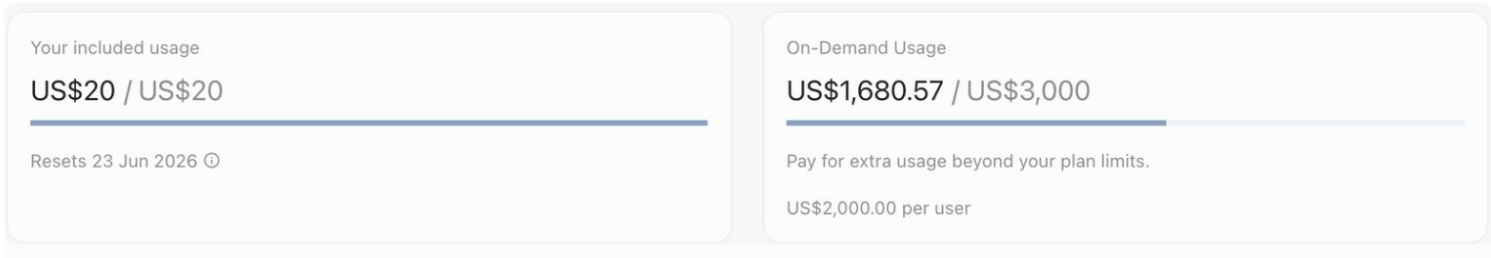
→ WE NEED TO SEE THE TRACE

You don't know what your agent will do until your users use it

LLMs are stochastic

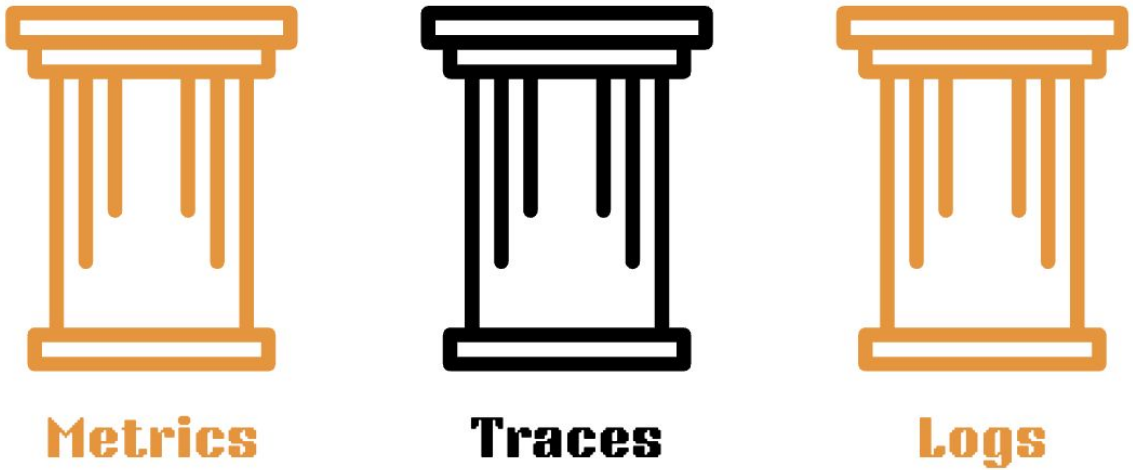


LLMs cost a lot



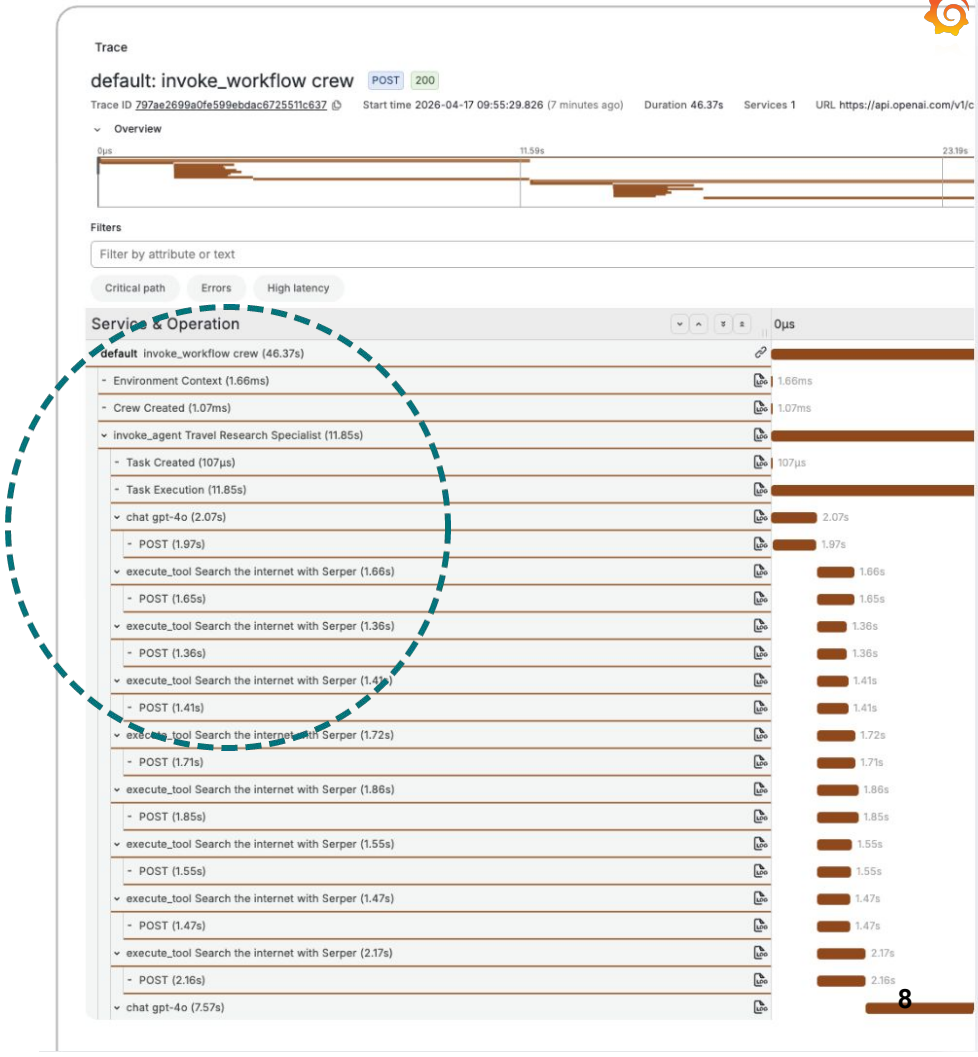
- **Which steps cost the most?**
- **How much does Opus 4.8 cost?**
- **Did my last change increase my cost?**
- **Which users cost the most?**

Pillars of Observability



What is a Trace?

172981728931hj23b1hg31y287398198



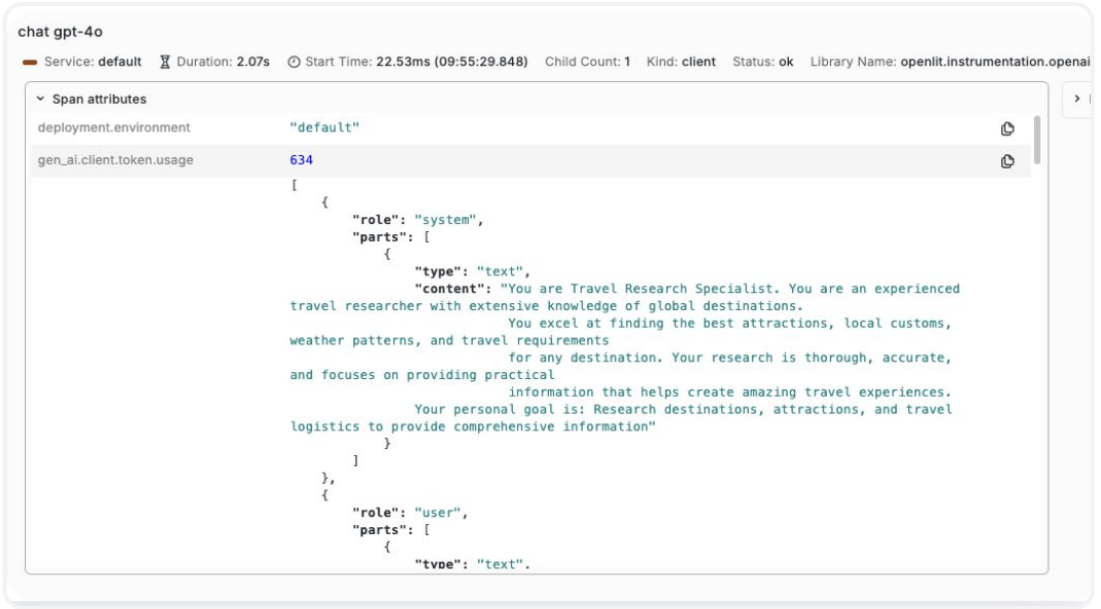
What is a span?

Core Definition

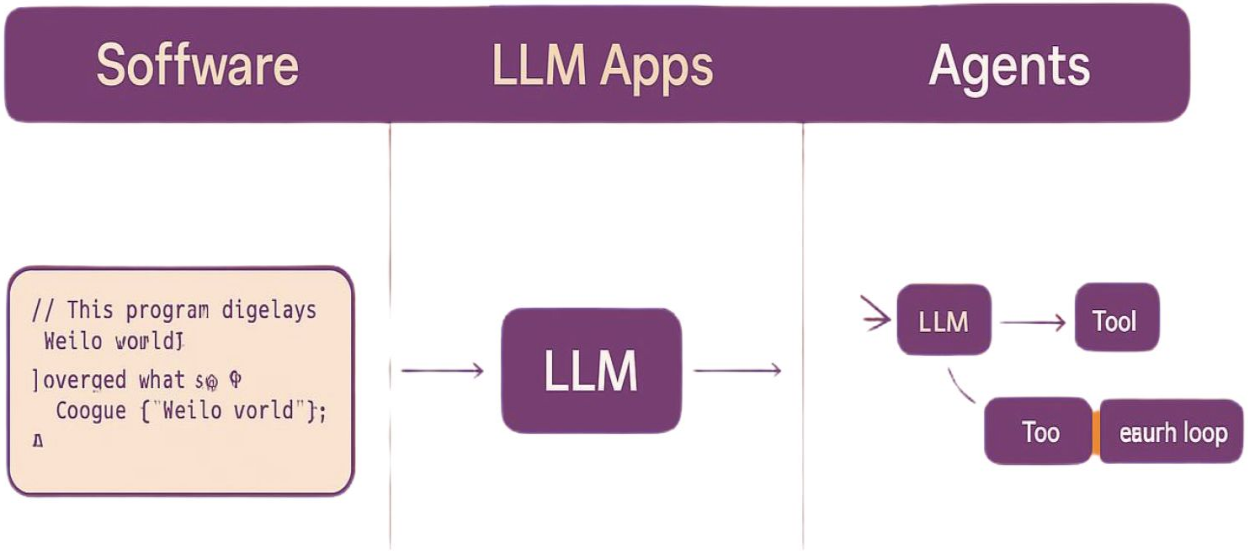
A span represents a **single operation** within a trace.

Attributes

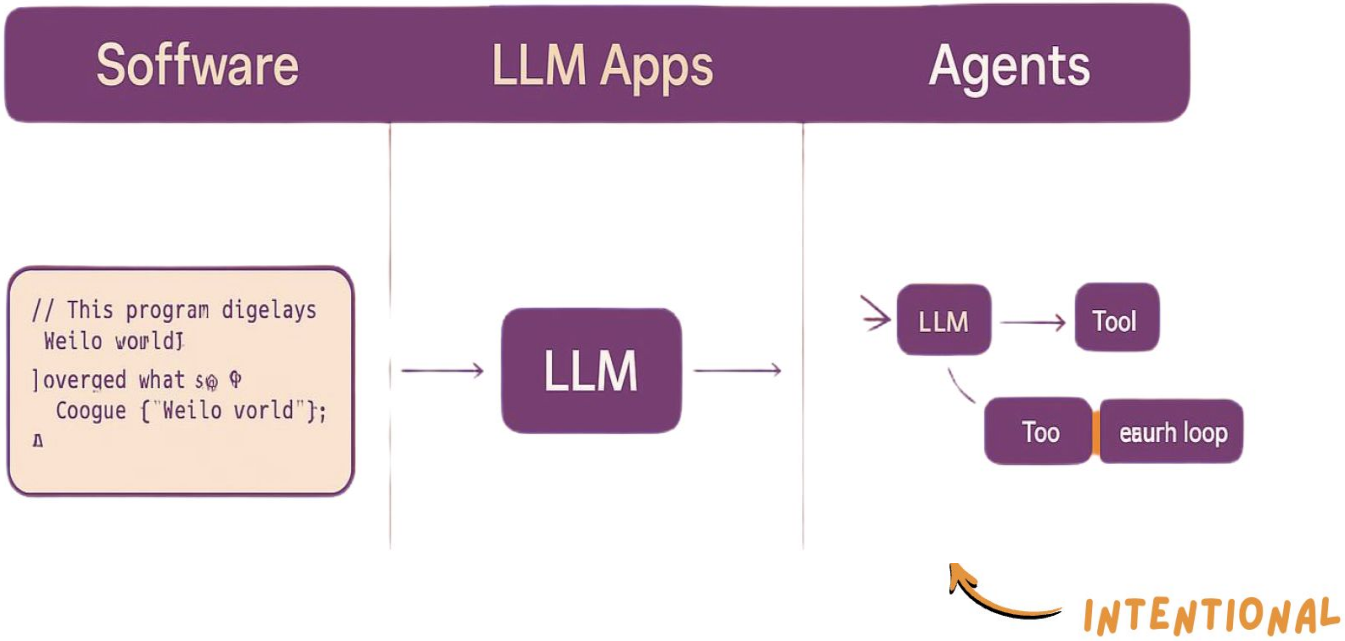
Key-value pairs providing metadata about the operation.



Building with AI is different



Building with AI is different

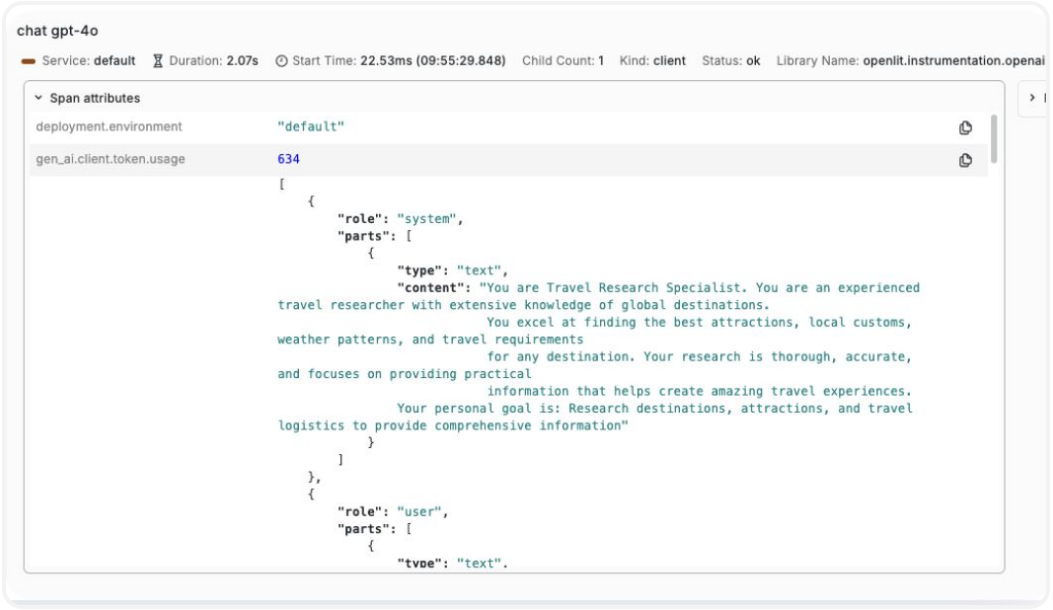


Primitives of LLM Observability

Single Turn

A single LLM request with Input and Outputs

- Captures the core interaction between the user and the model.
- Includes prompt metadata, model parameters, and response tokens.
- Essential for debugging individual model performance issues.

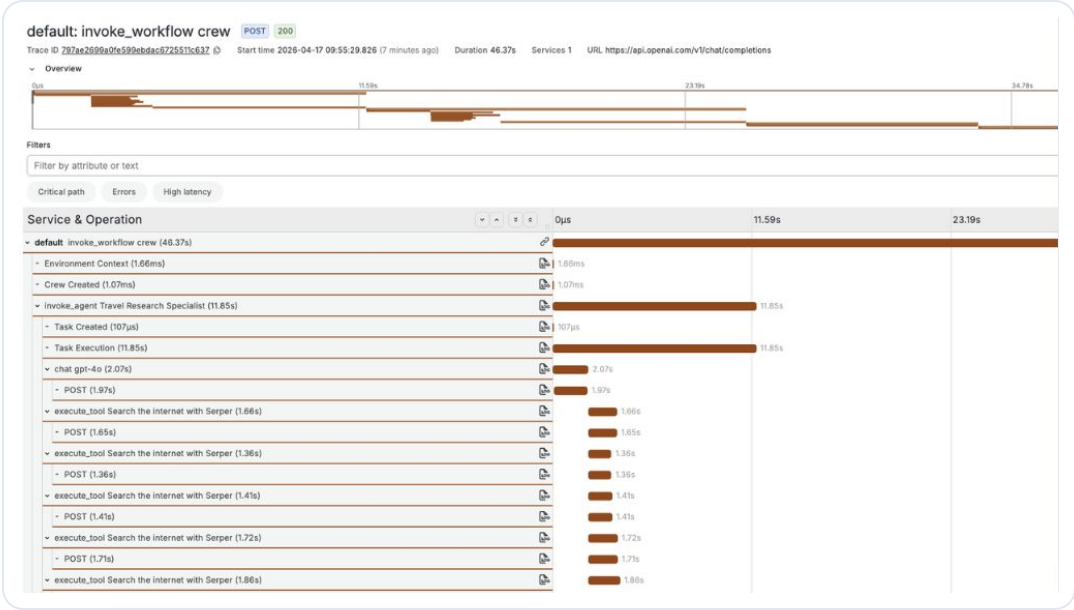


Primitives of Agent Observability

Multi Turn

A complete Agentic run showing all LLM requests and Tools usage

- Visualizes the full lifecycle of an autonomous agent's execution.
- Tracks sequential and parallel LLM calls alongside external tool invocations.
- Crucial for identifying bottlenecks and "loops" in complex agentic workflows.



The Role of OpenTelemetry

CNCF Momentum

Second-fastest growing project in CNCF ecosystem.

Industry Standard

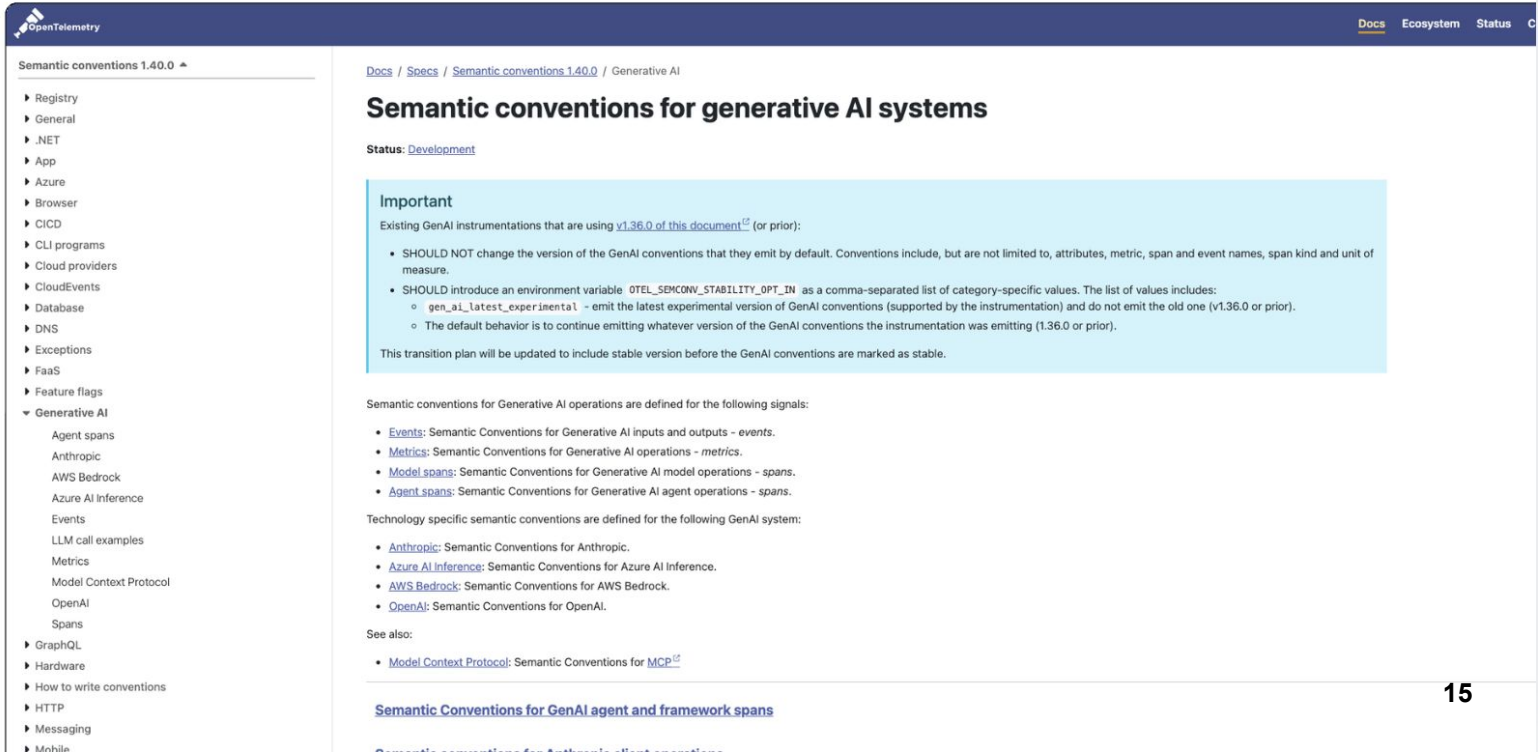
De facto standard for Observability across Traces, Metrics, and Logs.

Open & Neutral

Vendor neutral approach ensuring long-term flexibility and no lock-in.

“OTel is the USB-C of observability, one open standard for every signal.”

The Role of OpenTelemetry



Adding Observability to Agents

Basic

```
export OTEL_PYTHON_LOGGING_AUTO_INSTRUMENTATION_ENABLED=true
opentelemetry-instrument \
    --traces_exporter console \
    --metrics_exporter console \
    --logs_exporter console \
    --service_name dice-server \
    python my_ai_agent.py
```

Adding Observability to Agents

Adding Observability to Agents

OpenLIT

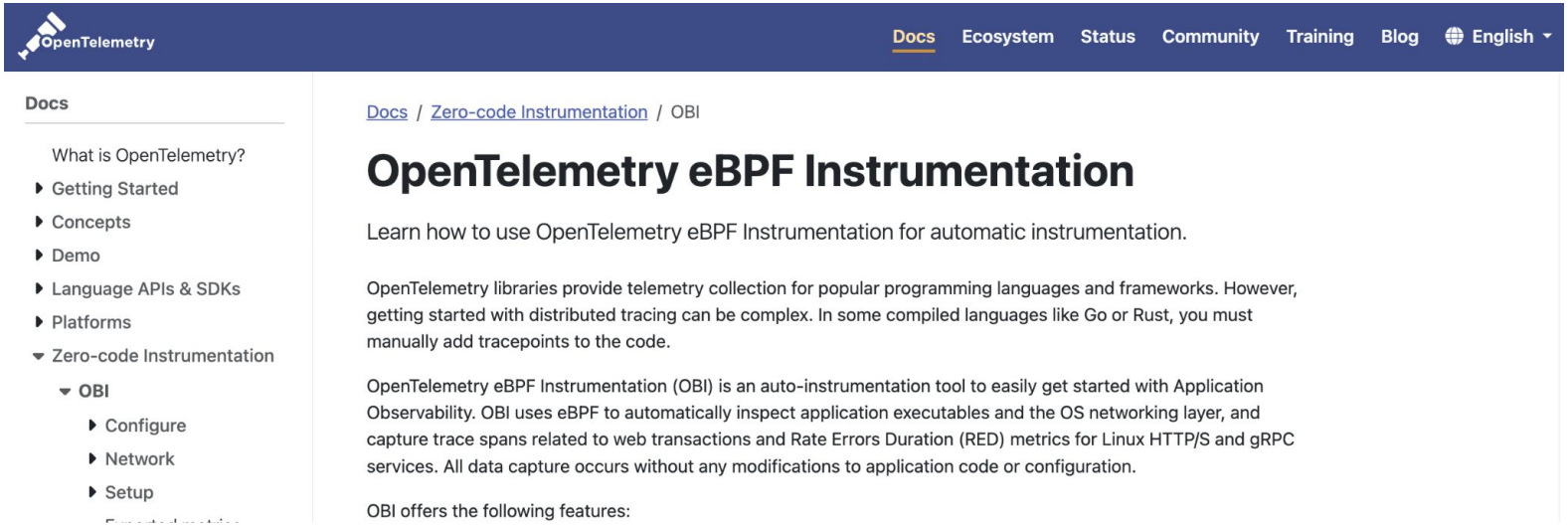
```
import openlit

openlit.init()
```

```
openlit-instrument \
  --service-name my-ai-app \
  --environment production \
  python my_ai_agent.py
```

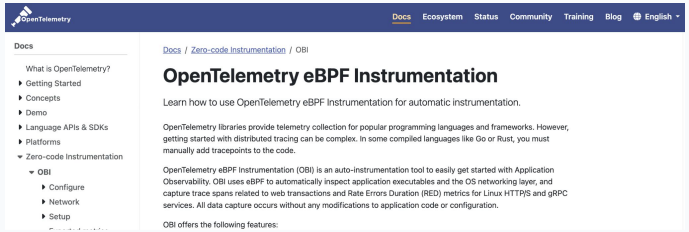
Adding Observability to Agents

Adding Observability to Agents (2026)



Adding Observability to Agents (2026)

INSTRUMENTATION



+

OpenLIT

BACKEND & VISUALIZATION

Traces

Metrics

Visualization

Demo (Dangerous)

Agent Evaluations

You are testing reasoning, not code paths

Single Turn

Did the Agent respond correctly according to the user prompt?

Multi Turn

Did the Agent maintain context across a conversation and perform well in e2e execution?

Agent Evaluations

We are testing reasoning, not code paths

Single Turn

Did the Agent respond correctly according to the user prompt?

Multi Turn

Did the Agent maintain context across a conversation and perform well in e2e execution?

Thankyou!

Linked in

Ishan Jain
Grafana Labs