

Observability for Gen AI: Monitoring LLM Applications with OpenTelemetry and Splunk

OBS1107



Forward- looking statements

This presentation may contain forward-looking statements regarding future events, plans or the expected financial performance of our company, including our expectations regarding our products, technology, strategy, customers, markets, acquisitions and investments. These statements reflect management's current expectations, estimates and assumptions based on the information currently available to us. These forward-looking statements are not guarantees of future performance and involve significant risks, uncertainties and other factors that may cause our actual results, performance or achievements to be materially different from results, performance or achievements expressed or implied by the forward-looking statements contained in this presentation.

For additional information about factors that could cause actual results to differ materially from those described in the forward-looking statements made in this presentation, please refer to our periodic reports and other filings with the SEC, including the risk factors identified in our most recent quarterly reports on Form 10-Q and annual reports on Form 10-K, copies of which may be obtained by visiting the Splunk Investor Relations website at www.investors.splunk.com or the SEC's website at www.sec.gov. The forward-looking statements made in this presentation are made as of the time and date of this presentation. If reviewed after the initial presentation, even if made available by us, on our website or otherwise, it may not contain current or accurate information. We disclaim any obligation to update or revise any forward-looking statement based on new information, future events or otherwise, except as required by applicable law.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. We undertake no obligation either to develop the features or functionalities described, in beta or in preview (used interchangeably), or to include any such feature or functionality in a future release.

Splunk, Splunk> and Turn Data Into Doing are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names or trademarks belong to their respective owners.

© 2025 Splunk LLC. All rights reserved.

splunk>

.conf25

Observability for Gen AI: Monitoring LLM Applications with OpenTelemetry and Splunk



**Derek
Mitchell**

Staff Observability Strategist | Splunk



**Sarah
Ware**

Sr. Observability Strategist | Splunk

Learning Objectives

#1

Learn the key concepts used to build LLM applications such as embeddings, vector databases, and Retrieval Augmented Generation (RAG).

#2

Learn how OpenTelemetry and Splunk Observability Cloud empower you to capture critical metrics, logs, and traces from your LLM applications.

#3

Learn how to troubleshoot LLM application issues and optimize performance using Splunk Observability Cloud.

Agenda

Key Concepts & Conceptual Solution

Implementation & Instrumentation

Optimization with Splunk Observability Cloud

Key Concepts & Conceptual Solution

What problem are we trying to solve?

Given a textbook of several hundred pages, we want the ability to provide answers to specific questions using the content of the textbook.

*"What are the four layers of soil?"
according to my science textbook*

Why can't we just use ChatGPT?

What are the four layers of soil?

The four main layers of soil, often referred to as **soil horizons**, are:

1. O Horizon (Organic Layer):

- Contains decomposed leaves, plants, and other organic matter.
- Dark in color and rich in nutrients.

2. A Horizon (Topsoil):

- A mix of minerals and organic material.
- Crucial for plant growth and home to many organisms like worms and insects.

3. B Horizon (Subsoil):

- Contains minerals leached down from the topsoil (like iron, clay, and aluminum).
- Less organic material than the A horizon and denser.

4. C Horizon (Parent Material):

- Made of partially broken-down rock.
- Little to no organic material and is the source of the soil above.

How can we solve the problem?

“What are the four layers of soil?”



Textbook



LLM



“The four layers of soil are topsoil, subsoil, parent rock, and bedrock. Topsoil is the upper layer, containing humus and nutrients...”

How can we solve the problem?

“What are the four layers of soil?”



Relevant portion(s)
of Textbook



LLM



“The four layers of soil are topsoil, subsoil, parent rock, and bedrock. Topsoil is the upper layer, containing humus and nutrients...”

How can we solve the problem?

“What are the four layers of soil?”



Textbook Pages



Embeddings Model



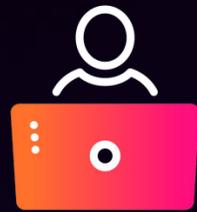
[Page 1, 8471, 491, 401,
8711, 19844, 401, 15915,
816...],
...
[Page 132, 3923, 527,
279, 3116, 13931, 315,
17614, 30...],
...

Embeddings



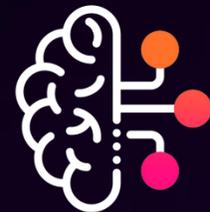
Vector Database

Bringing it all together with Retrieval-Augmented Generation (RAG)



User Prompt

“What are the four layers of soil?”



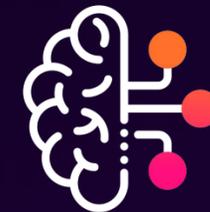
Embeddings Model



Vector Database



Prompt + Related Docs



LLM

“The four layers of soil are topsoil, subsoil...”

Implementation & Instrumentation

Loading the Textbook PDF

```
from langchain_community.document_loaders import PyPDFLoader

# Load the textbook PDF
loader = PyPDFLoader(
    file_path,
    mode="page",
)

textbook_pages = loader.load()
```

Calculate Embeddings and Store in the Vector Database

```
from langchain_openai import OpenAIEmbeddings
from langchain_chroma import Chroma

embeddings_model = OpenAIEmbeddings(model="text-embedding-3-large")

db = Chroma.from_documents(
    textbook_pages,
    embedding=embeddings_model,
    persist_directory="../chroma_db"
)
```

Initialize the Vector DB and LLM

```
from langchain.chat_models import init_chat_model
from langchain_chroma import Chroma

embeddings_model = OpenAIEmbeddings(model="text-embedding-3-large")

vector_store = Chroma(
    persist_directory="./chroma_db",
    embedding_function=embeddings_model
)

llm = init_chat_model("gpt-4o-mini", model_provider="openai")
```

Initialize the Vector DB and LLM

```
from langchain_core.prompts import ChatPromptTemplate
```

```
prompt = ChatPromptTemplate.from_template("You are an assistant for question-answering tasks. Use  
"  
    + "the following pieces of retrieved context to answer the question. If you don't know the  
answer, "  
    + "just say that you don't know. Use three sentences maximum and keep the answer concise.\n"  
    + "Question: {question}\n"  
    + "Context: {context}\n"  
    + "Answer:")
```

Define functions to retrieve related documents and generate a response using the LLM

```
class State(TypedDict):
    question: str
    context: List[Document]
    answer: str

def retrieve(state: State):
    retrieved_docs = vector_store.similarity_search(state["question"])
    return {"context": retrieved_docs}

def generate(state: State):
    docs_content = "\n\n".join(doc.page_content for doc in state["context"])
    messages = prompt.invoke({"question": state["question"], "context": docs_content})
    response = llm.invoke(messages)
    return {"answer": response.content}
```

Build a graph and invoke it

```
from langgraph.graph import START, StateGraph

graph_builder = StateGraph(State).add_sequence([retrieve, generate])
graph_builder.add_edge(START, "retrieve")
graph = graph_builder.compile()

response = graph.invoke({"question": "What are the four layers of soil?"})
```

The four layers of soil are topsoil, subsoil, parent rock, and bedrock. Topsoil is the upper layer, containing humus and nutrients; subsoil is beneath it with some humus and minerals...

**We have our
application deployed
but how do we get
insight into its
performance?**

Observability in 2 Steps

1. Install the [Splunk Distribution of the OpenTelemetry Collector](#)
2. Instrument the Python application with the [Splunk Distribution of OpenTelemetry Python](#)



Install the Splunk Distribution of the OpenTelemetry Collector

```
curl -sSL https://dl.signalfx.com/splunk-otel-collector.sh > /tmp/splunk-otel-collector.sh;  
sudo sh /tmp/splunk-otel-collector.sh \  
--realm $SPLUNK_REALM \  
--memory $SPLUNK_MEMORY_TOTAL_MIB \  
-- $SPLUNK_ACCESS_TOKEN
```

Instrument the Python application with the Splunk Distribution of OpenTelemetry Python

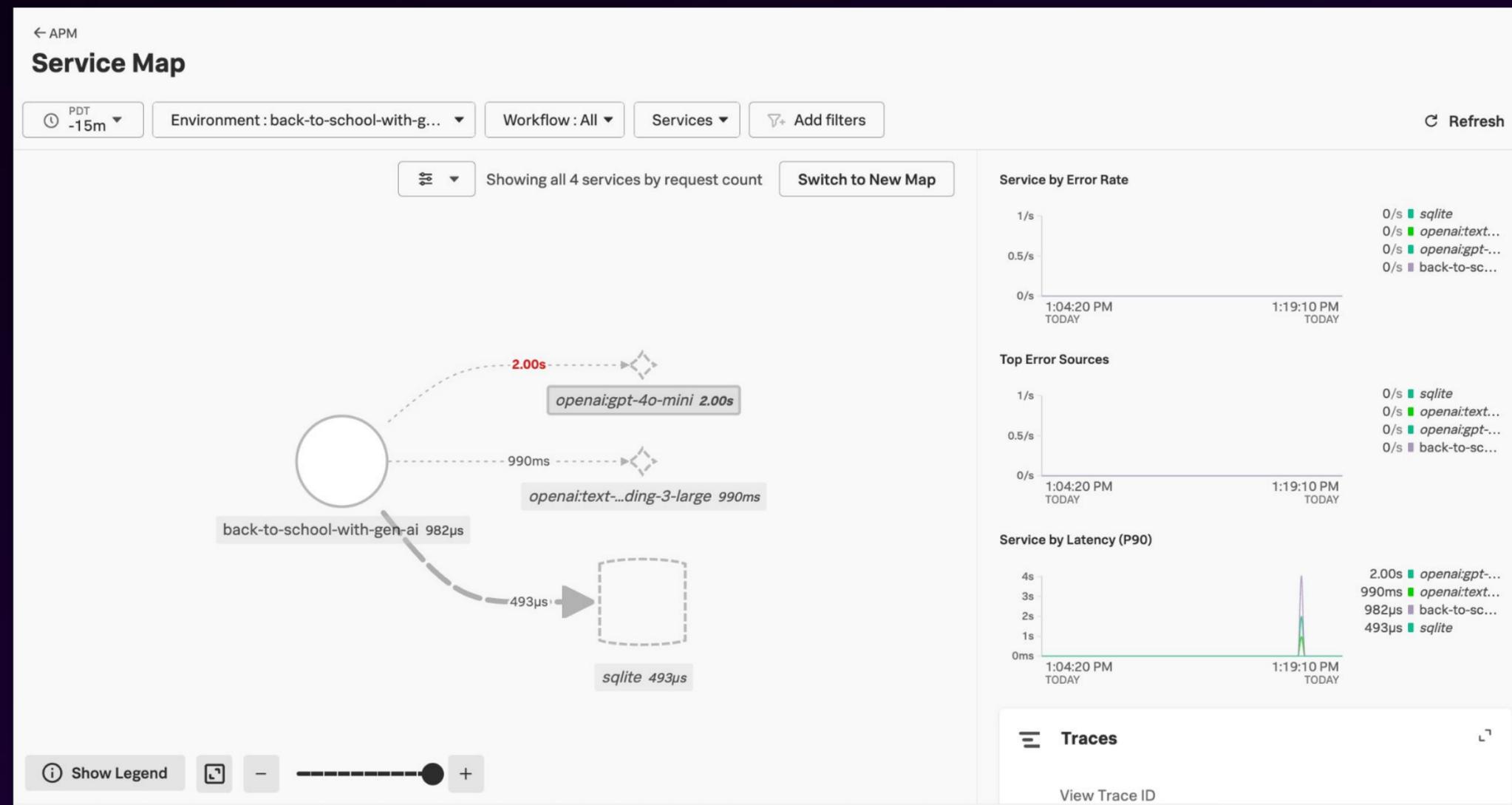
```
# install the package
pip install "splunk-opentelemetry[all]"

# install instrumentation for packages used by the app
opentelemetry-bootstrap -a install

# tell OpenTelemetry how to report data
export OTEL_SERVICE_NAME=back-to-school-with-gen-ai
export OTEL_RESOURCE_ATTRIBUTES='deployment.environment=test'

# start the application with instrumentation
opentelemetry-instrument python app.py
```

View data in Splunk Observability Cloud



Optimization with Splunk Observability Cloud

Demo

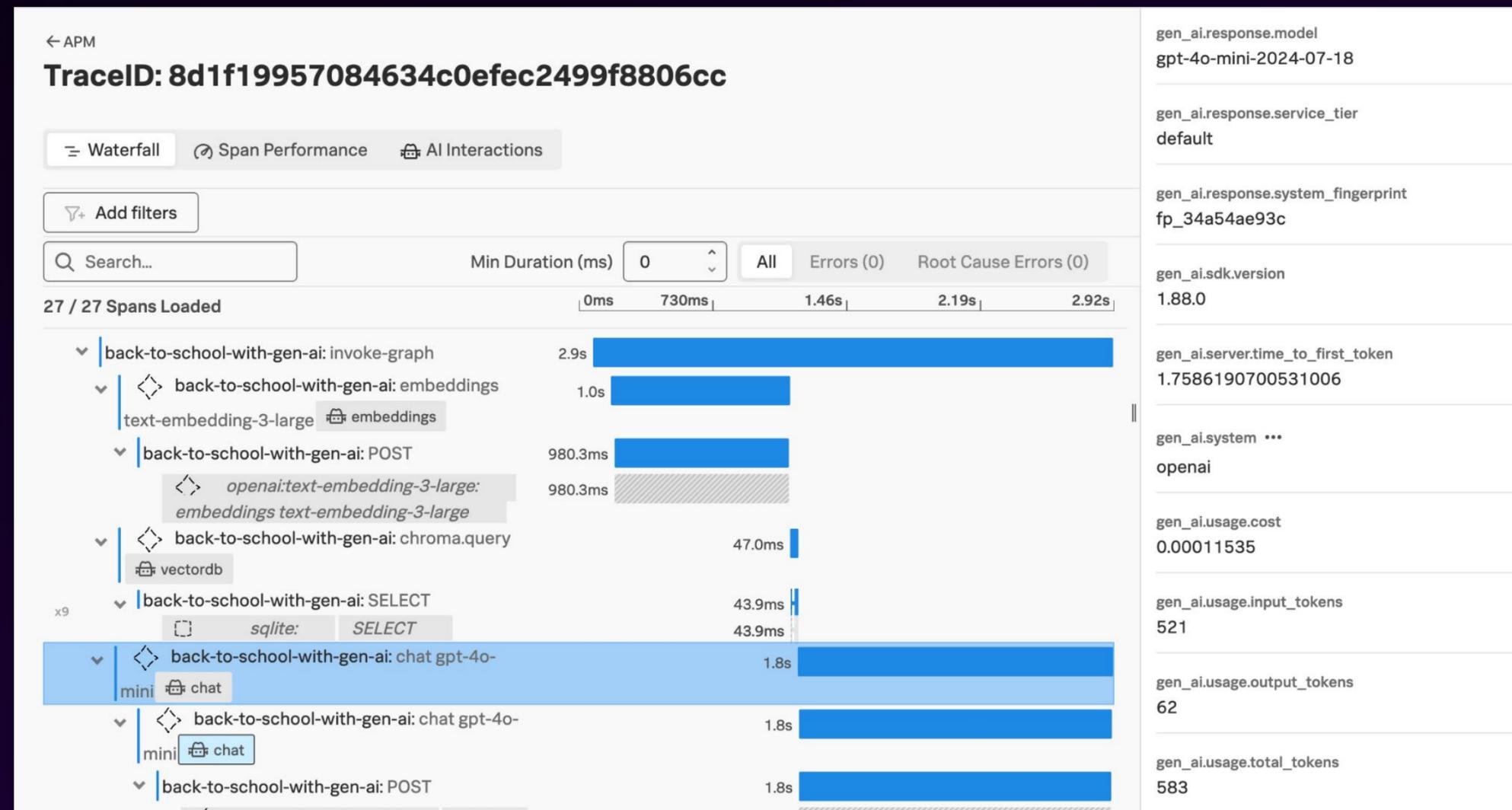
Similarity Search Implementation

```
def retrieve(state: State):  
  
    retrieved_docs = vector_store.similarity_search(  
        query = state["question"]  
    )  
  
    return {"context": retrieved_docs}
```

Similarity Search Implementation

```
def retrieve(state: State):  
  
    retrieved_docs = vector_store.similarity_search(  
        query = state["question"],  
        k = 2  
    )  
  
    return {"context": retrieved_docs}
```

Trace with Reduced Document Count



Observability for AI Product Roadmap

AI Infrastructure Monitoring

Out of the box dashboards and real time detection for common AI components

Agentic Conversations

Visualization of agent interactions, tool calls, and end-to-end behavior

Quality Evaluations

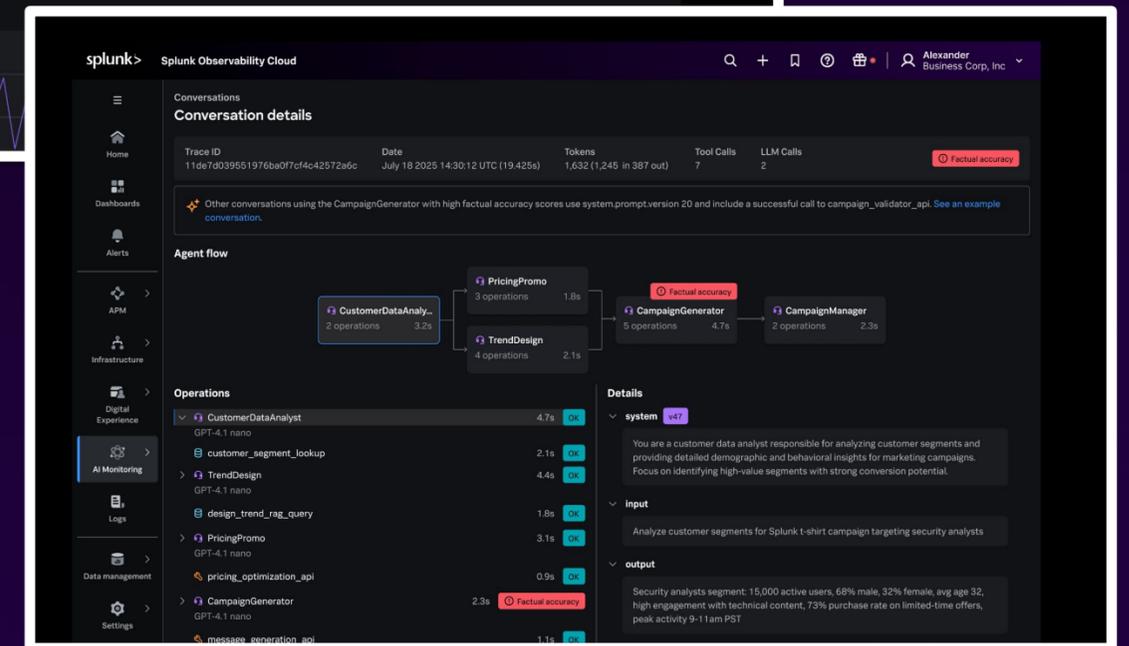
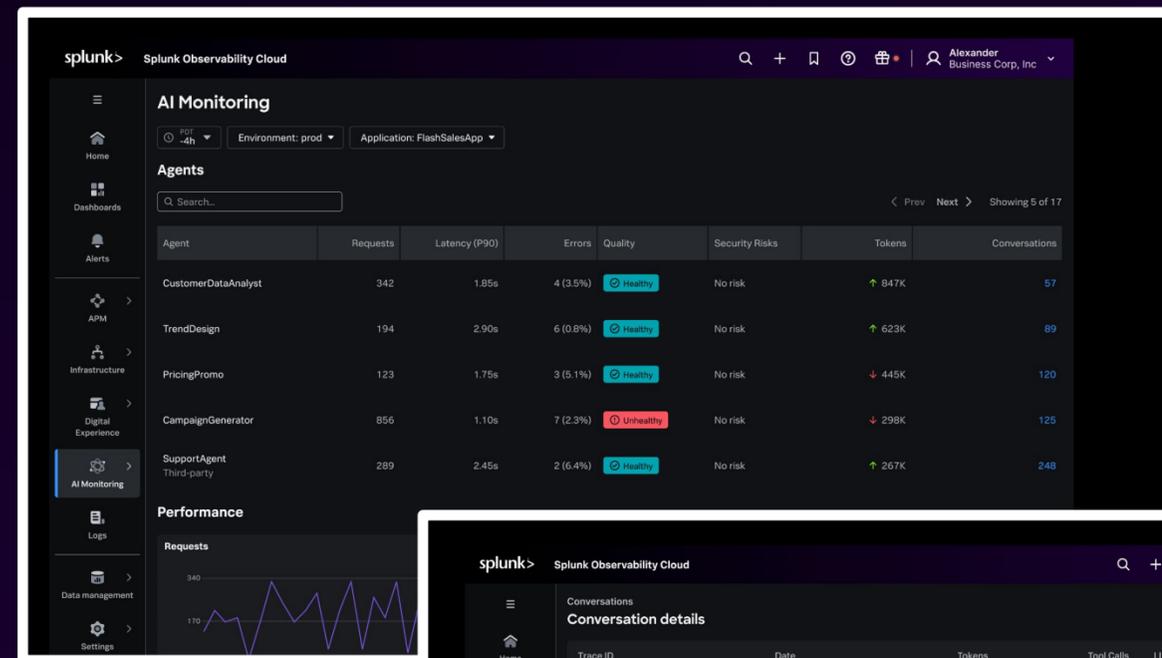
Measure and troubleshoot the quality of agent / LLM decisions and responses

Security

Detect and protect against LLM risks such as prompt injections, DDoS, PII leakage, etc.

Agentic Conversation and Evaluation

- Understand the health, quality and cost of each agent running production
- Online evaluation of LLM responses to assess the effectiveness of application
- Trace all interactions between users, agents and tools in a dedicated view
- Visualize multi-agent workflows, to understand inputs, outputs and agent decisions
- Pinpoint issues at every step including incorrect tool calls, retries, latency and quality issues



Recap

Here's what we covered in today's session:

- The key concepts used to build LLM applications such as embeddings, vector databases, and Retrieval Augmented Generation (RAG).
- How OpenTelemetry and Splunk Observability Cloud empower you to capture critical metrics, logs, and traces from your LLM applications.
- How to troubleshoot LLM application issues and optimize performance using Splunk Observability Cloud.



**What questions
do you have?**

Call to Action

Splunk® Observability Cloud Trial

If you're not already using Splunk® Observability Cloud today, you can sign up for a free 14-day trial at:

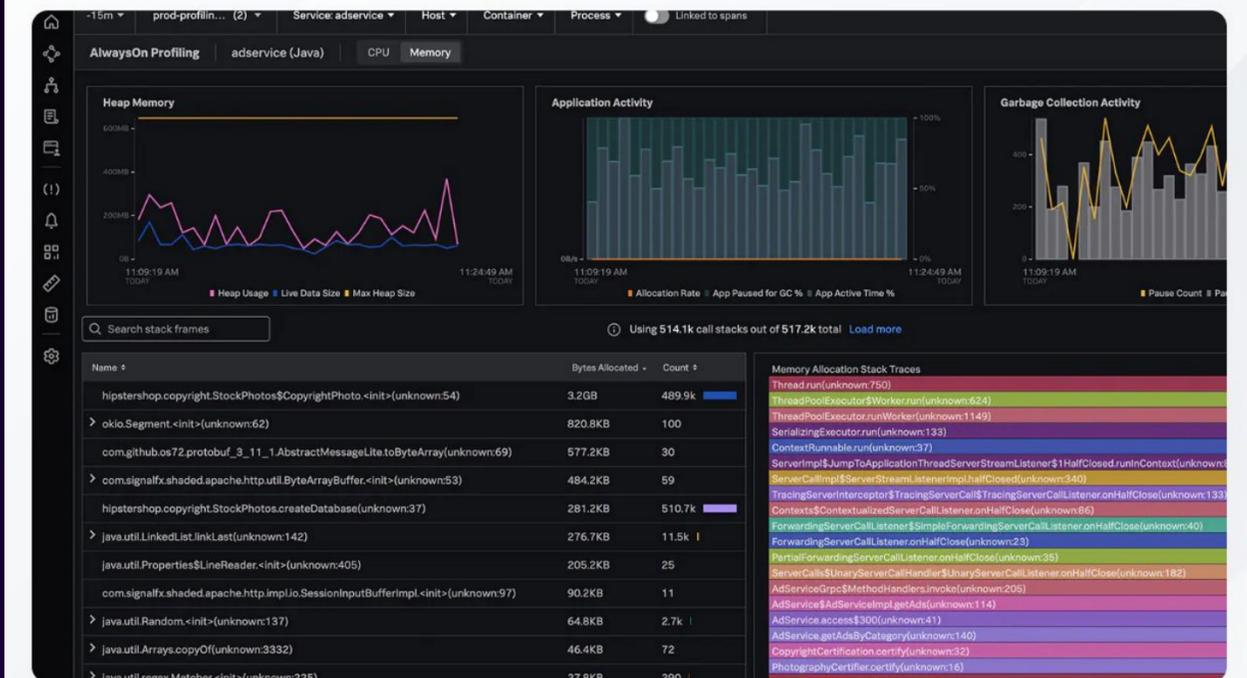
cs.co/observability-trial

Or ask your Splunk sales representative about it.

Visit voc.splunk.com to sign up for early access to new features!

Splunk Observability Cloud Free Trial

Try Splunk Observability Cloud free for 14 days. No credit card required.



Key Community Contributions



OpenTelemetry

Standardize conventions for data collection, evaluations and integrate standardized instrumentation for agentic systems

CNCF Top Project



Develop an infrastructure stack for agent collaboration, including discovery, identity, messaging, and observability

Part of the Linux Foundation

Learn about AGNTCY
<https://agntcy.org/>



Let's Connect

We'd love to hear how your Observability for AI journey is going!

Email:

dmitch@cisco.com

sarware@cisco.com

LinkedIn:

<https://www.linkedin.com/in/derek-mitchell-a-b15491/>

<https://www.linkedin.com/in/seware/>

Thank you

