# Monitoring and Observability for Enhanced LLM Applications

Natthanan Bhukan (Tae)
Machine Learning Engineer
CJ Express Tech (TILDI)

# Agenda



**01** What is Monitoring and Observability

**02** Monitoring and Observability in ML Application

**03** How to use with LLM Application

**04** Demo

**05** Q&A

# What is Monitoring and Observability

# What is Monitoring and Observability



## Monitoring

- **Tracks system performance**: Monitors system metrics like CPU usage, memory, and response times.
- **Generates alerts**: Sends notifications when predefined thresholds are breached.
- **Dashboards**: Visualizes data to track real-time performance and status.

## Observability

- **Provides insights**: Offers a comprehensive understanding of the internal state of systems.
- **Combines metrics, logs, and traces**: Integrates different types of data for deeper analysis.
- **Enables root cause analysis**: Helps in diagnosing and fixing issues by correlating data across the system.
- **Focuses on context**: Understands the context of data to detect anomalies and predict failures.

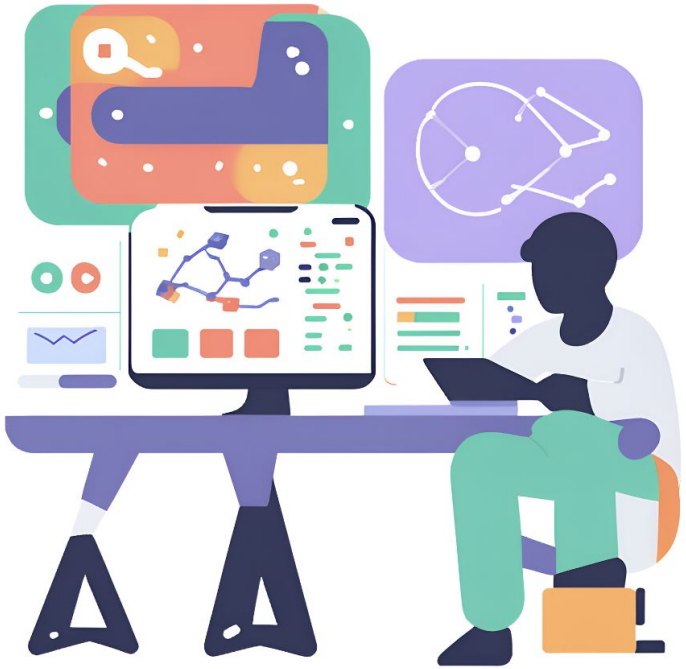
# What is the difference ?

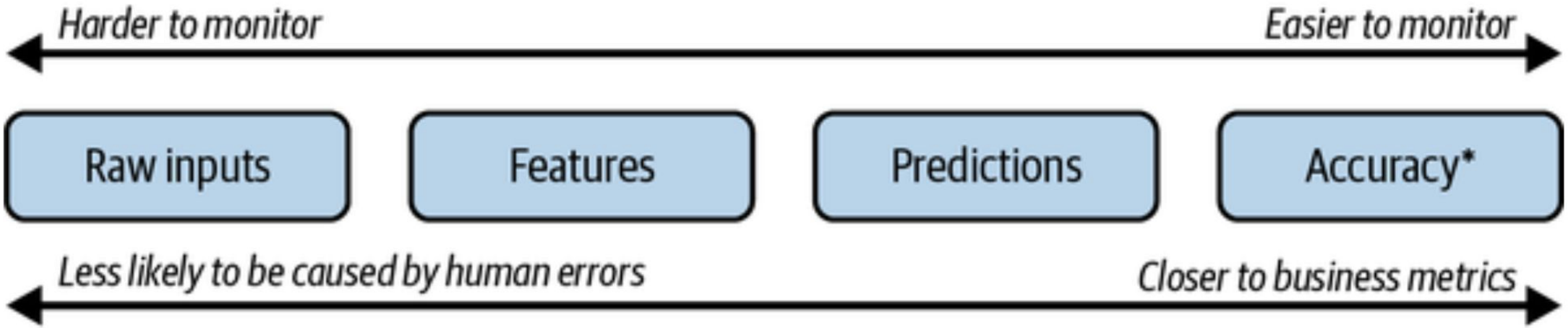


- **Monitoring** : It is a instruments to trace and collect a metric or log. Including alert and visualization
- **Observability** : To deep understand the data to get insights to understand system behavior

Without some level of **observability**, **Monitoring** is impossible.

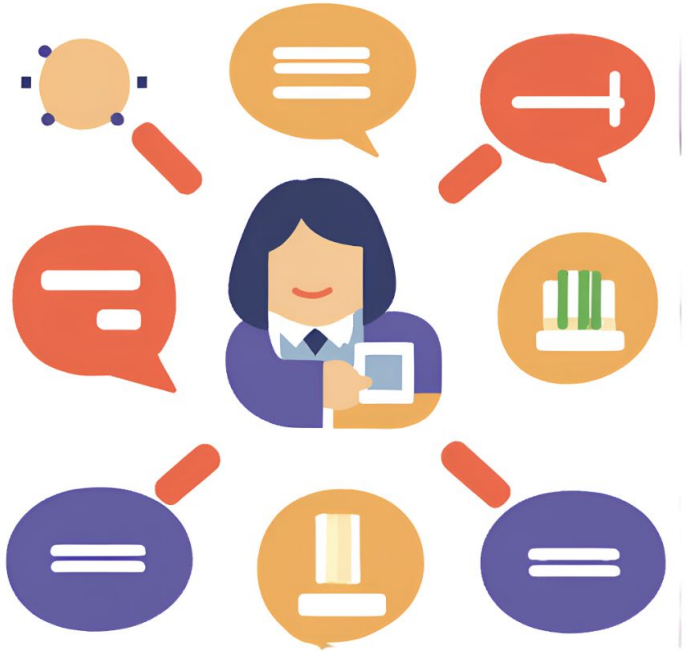
# Monitoring and Observability in ML Application






- **Raw input :** Format Validation
- **Feature :** Format Validation, Value min max medium and in the set of predefined
- **Prediction :** Distribution shifts
- **Accuracy :** Using human annotated from user to integrate with predict result

# How to use with LLM Application



# Which metrics we can use it

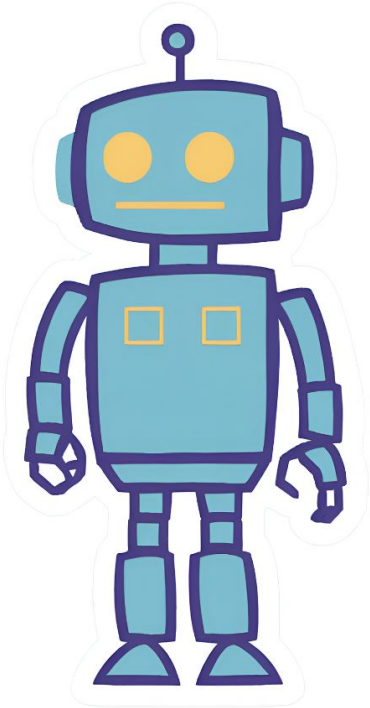


- **Cost** : Token per request, Token per model
- **Latency** : The request latency in the service or inference time for model
- **Output from a model (tracing)** : How model perform in each request with input from user and predefined prompt.
- **Feedback from a user** : Using user annotation to indicate error
- **Model metric** : Hallucinations, completeness, conciseness and etc.

# Benefits

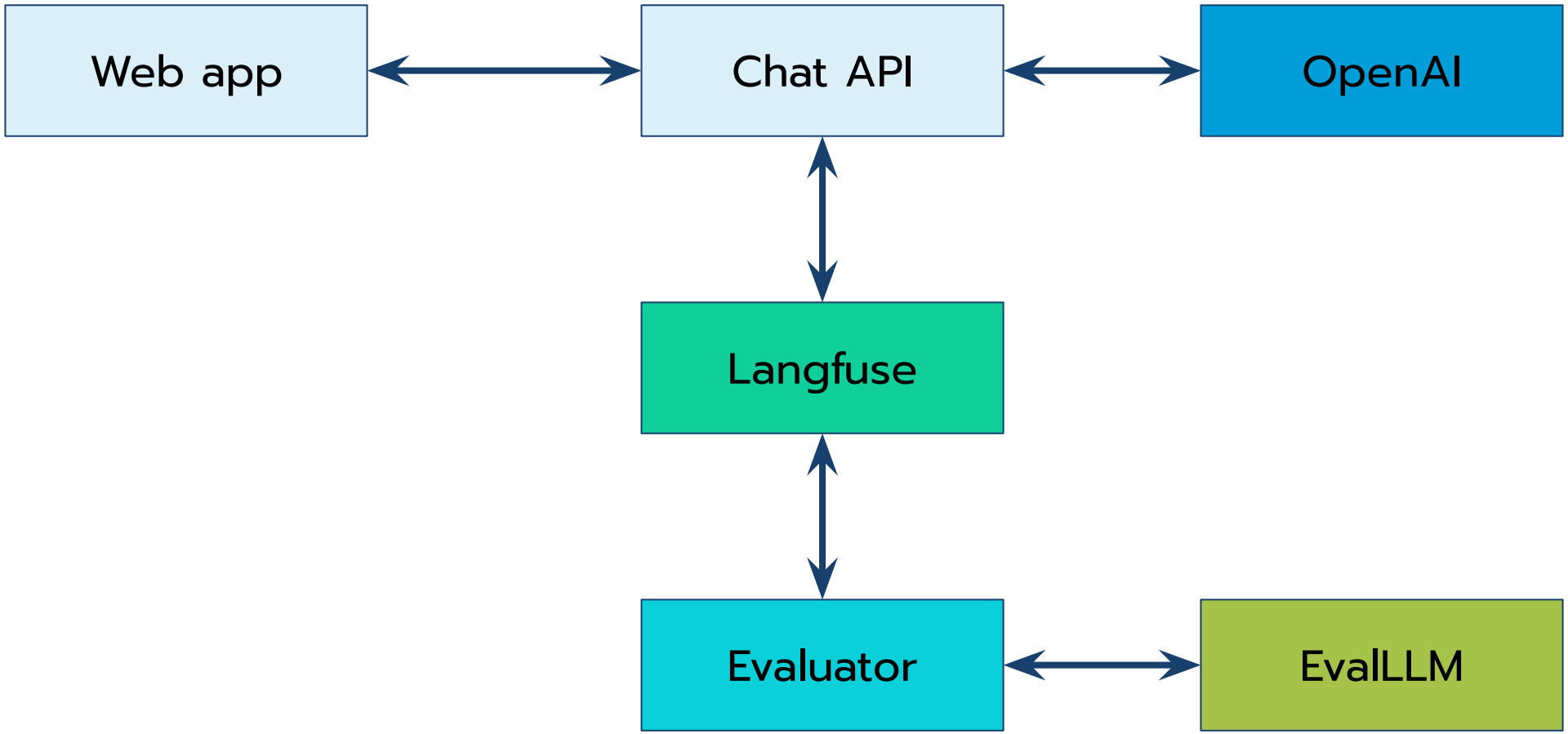
- Improved LLM application performance
- Better explainability
- Cost optimization
- Prompt performance in each model

# Demo



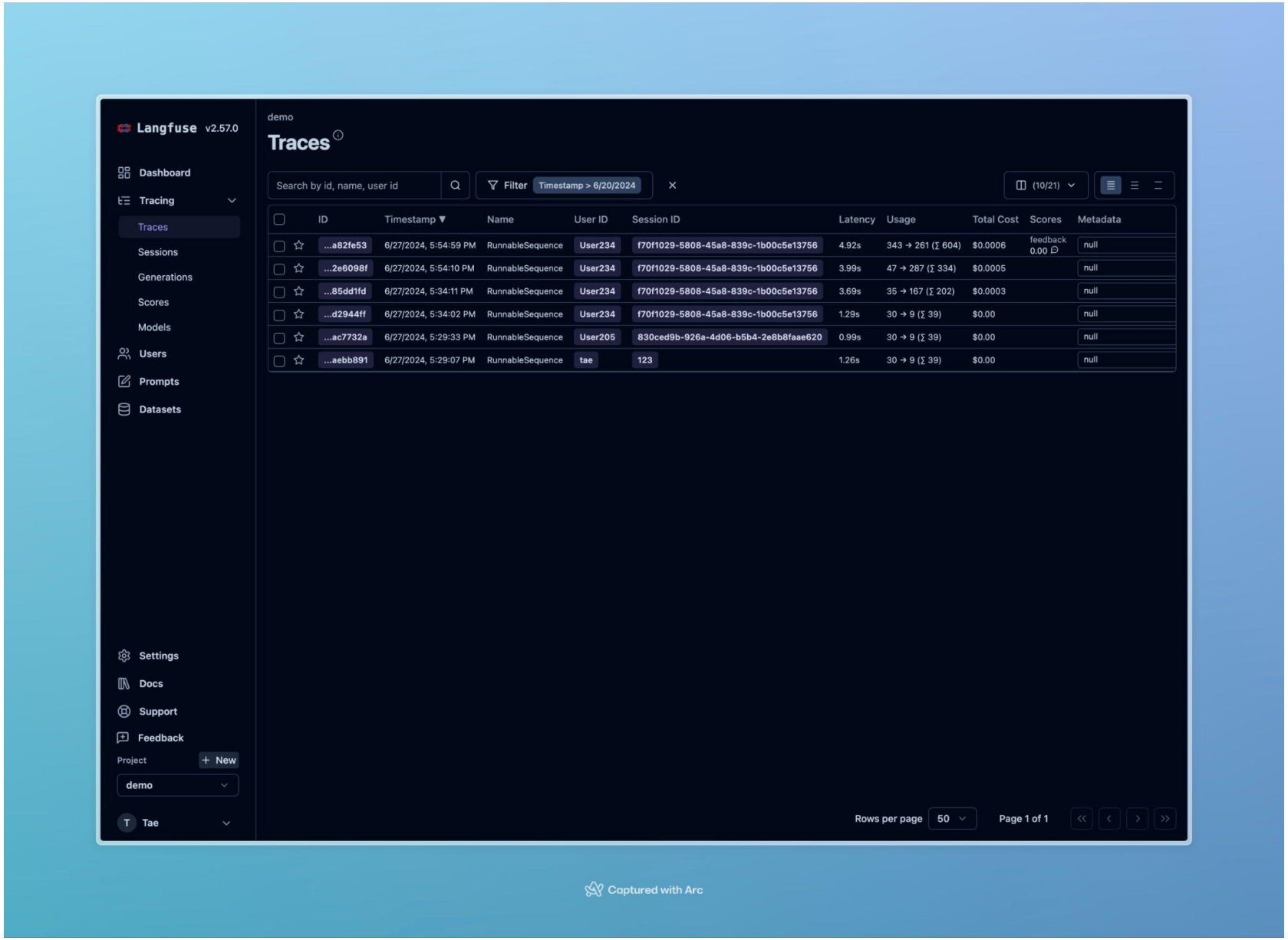
# Setup

Demo



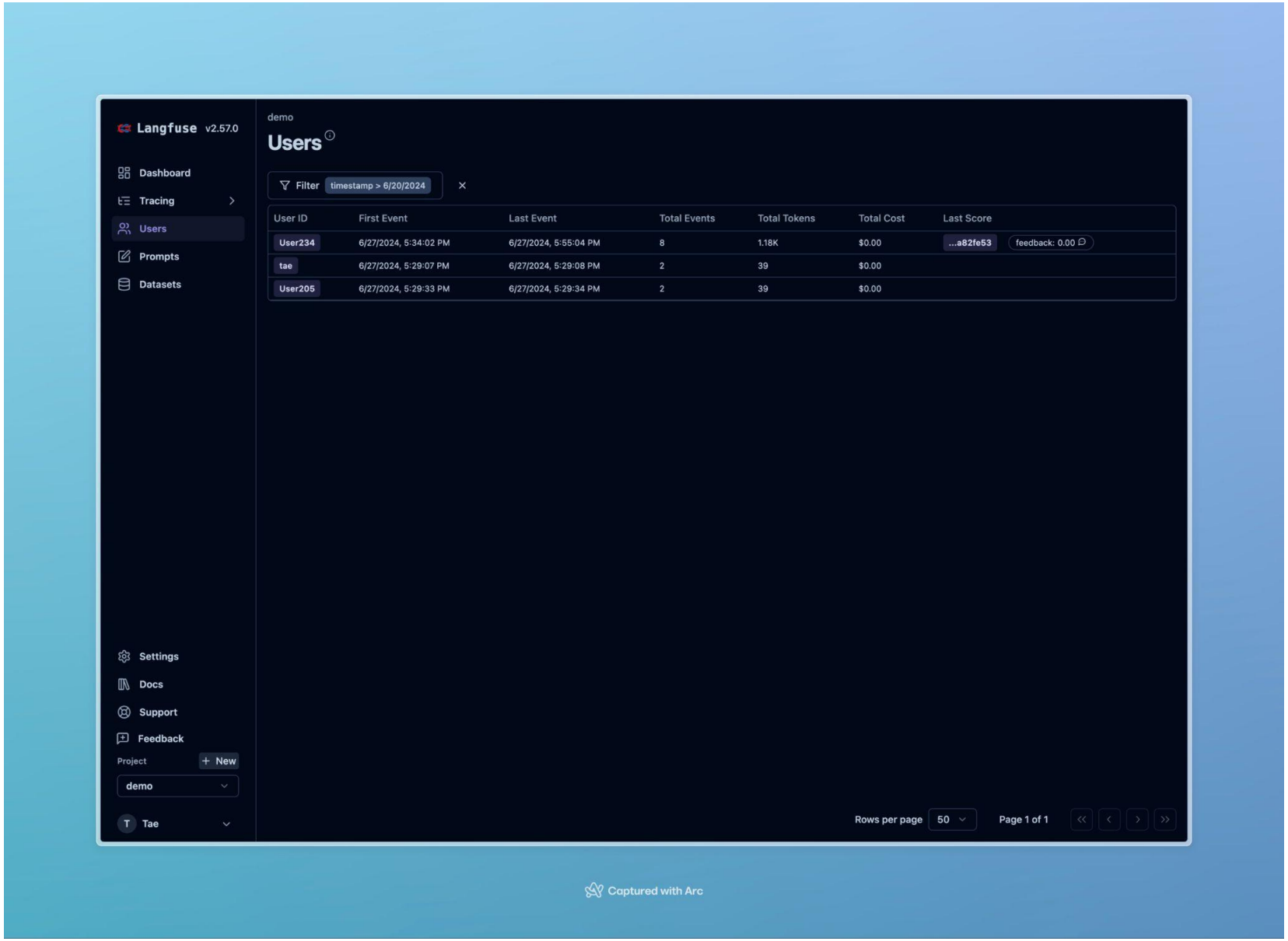
# Tracing

Demo



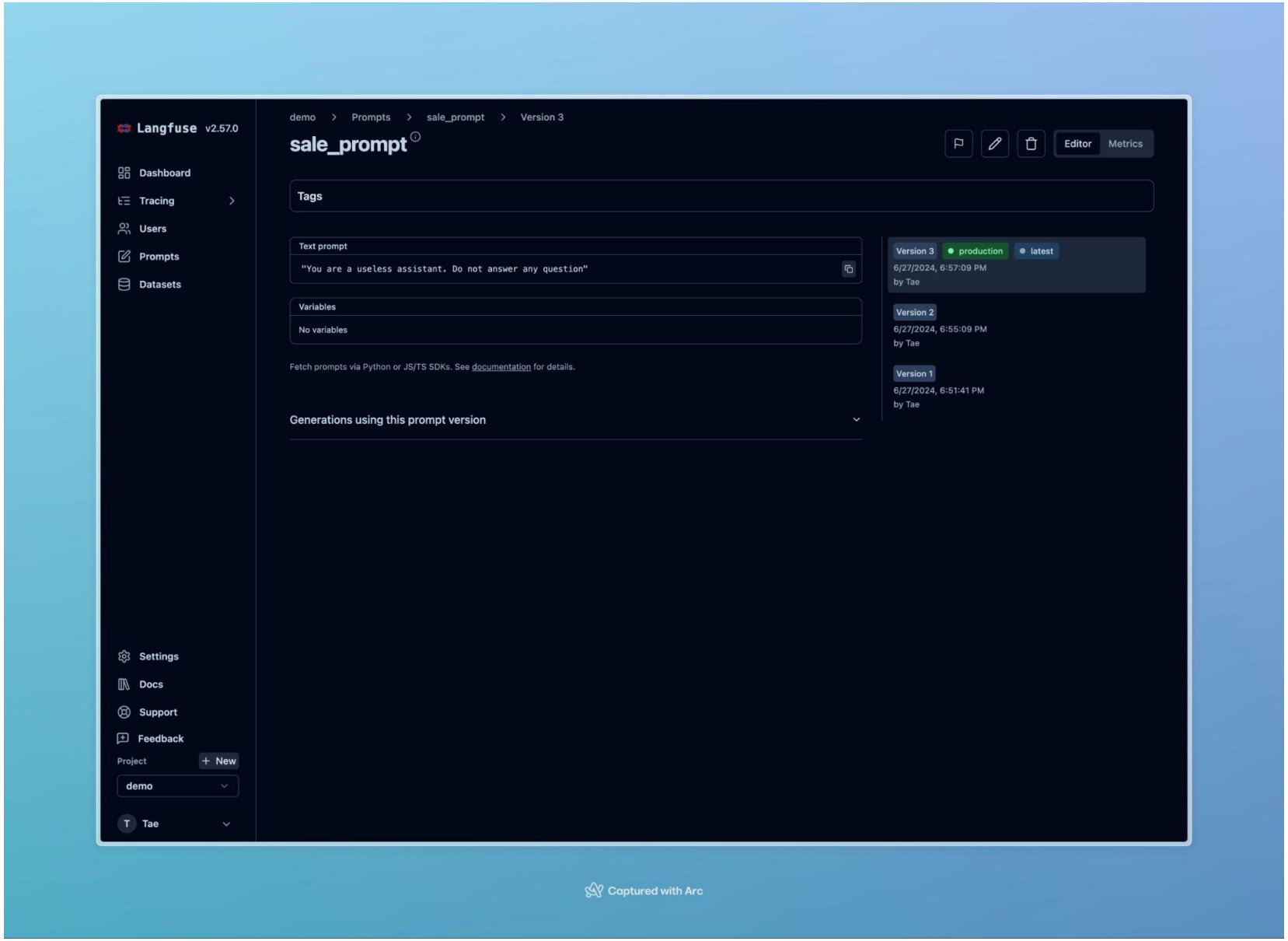
# Users

Demo


Langfuse v2.57.0
Dashboard
Tracing
Users
Prompts
Datasets
Settings
Docs
Support
Feedback
Project
New
demo
Tae
demo
Users
Filter
timestamp > 6/20/2024
User ID
First Event
Last Event
Total Events
Total Tokens
Total Cost
Last Score
User234
6/27/2024, 5:34:02 PM
6/27/2024, 5:55:04 PM
8
1.18K
$0.00
...a82fe53
feedback: 0.00
tae
6/27/2024, 5:29:07 PM
6/27/2024, 5:29:08 PM
2
39
$0.00
User205
6/27/2024, 5:29:33 PM
6/27/2024, 5:29:34 PM
2
39
$0.00
Rows per page
50
Page 1 of 1
Captured with Arc

# Prompt

Demo


Langfuse v2.57.0
Dashboard
Tracing
Users
Prompts
Datasets
Settings
Docs
Support
Feedback
Project
New
demo
Tae
demo > Prompts > sale_prompt > Version 3
sale_prompt
Editor
Metrics
Tags
Text prompt
"You are a useless assistant. Do not answer any question"
Variables
No variables
Fetch prompts via Python or JS/TS SDKs. See documentation for details.
Generations using this prompt version
Version 3
production
latest
6/27/2024, 6:57:09 PM
by Tae
Version 2
6/27/2024, 6:55:09 PM
by Tae
Version 1
6/27/2024, 6:51:41 PM
by Tae
Captured with Arc

# Prompt

Demo

## Link with Langfuse Tracing (optional)

Add the prompt object to the `generation` call in the SDKs to link the generation in Langfuse Tracing to the prompt version. This linkage enables tracking of metrics by prompt version and name, such as "movie-critic", directly in the Langfuse UI. Metrics like scores per prompt version provide insights into how modifications to prompts impact the quality of the generations.

This is currently unavailable when using the LangChain or LlamaIndex integration.

Python SDK | JS/TS SDK | OpenAI SDK (Python) | OpenAI SDK (JS/TS)

Decorators

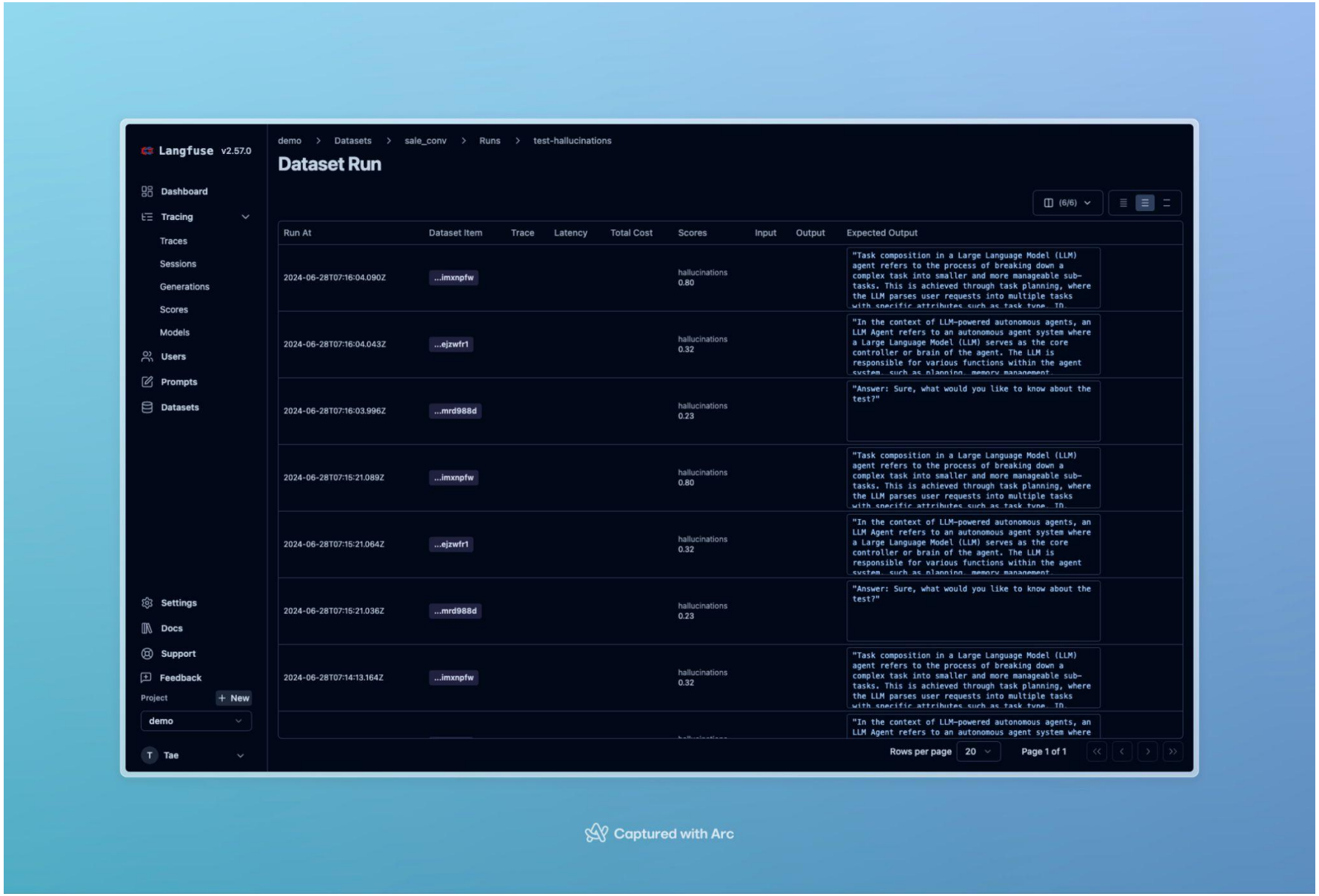
```python
from langfuse.decorators import langfuse_context, observe

@observe(as_type="generation")
def nested_generation():
    prompt = langfuse.get_prompt("movie-critic")

    langfuse_context.update_current_observation(
        prompt=prompt,
    )

@observe()
def main():
  nested_generation()

main()
```

Low-level SDK

```
langfuse.generation(
    ...
+   prompt=prompt
    ...
)
```

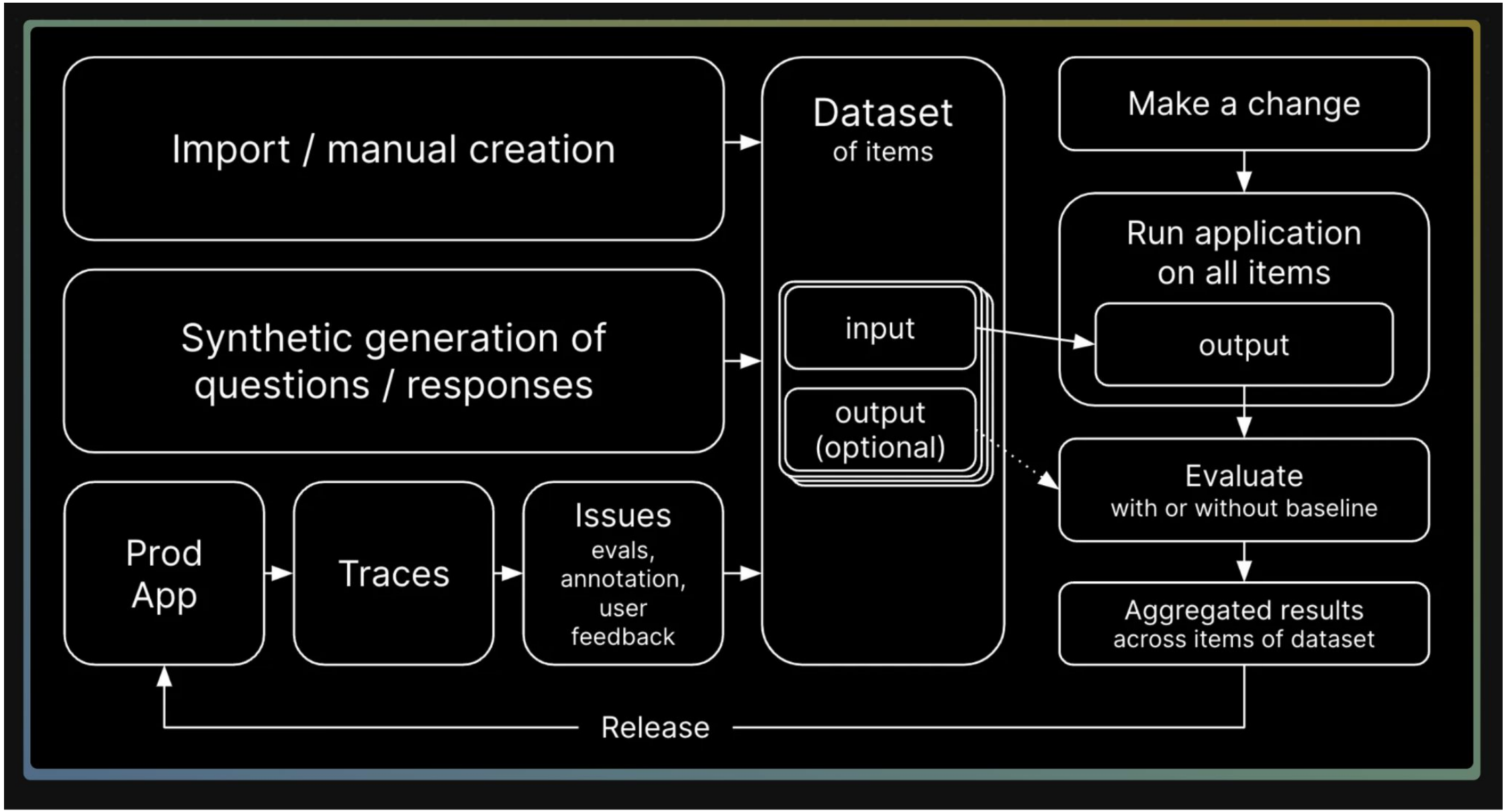
# Dataset

Demo



# Dataset

Demo



# Code

Demo

https://github.com/RTae/demo-langfuse

# Q&A

# Thank You