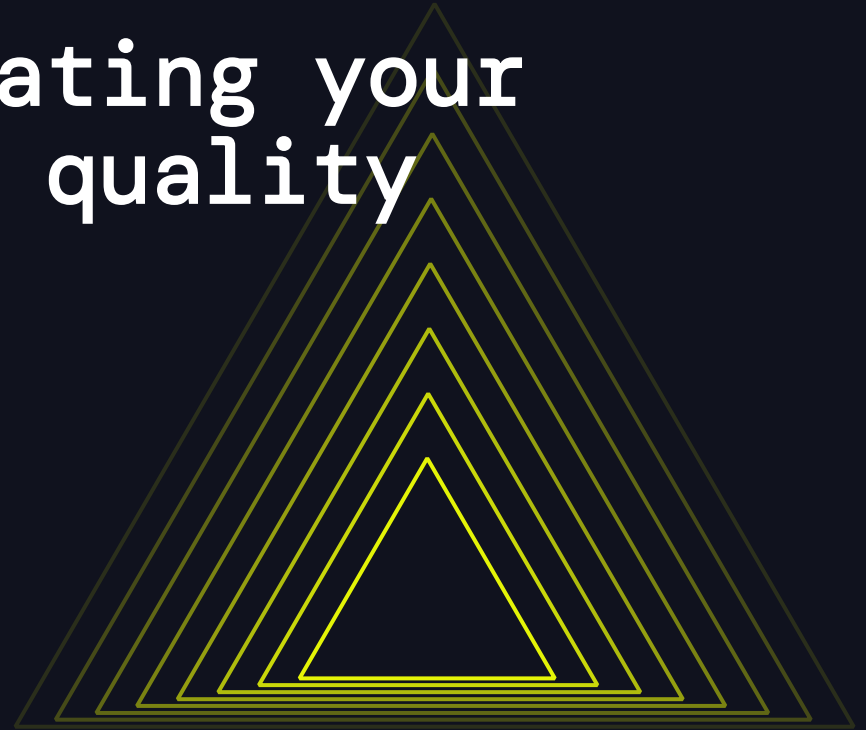


Methods for evaluating your GenAI application quality



Michael Carbin, Alkis Polyzotis
June 2024

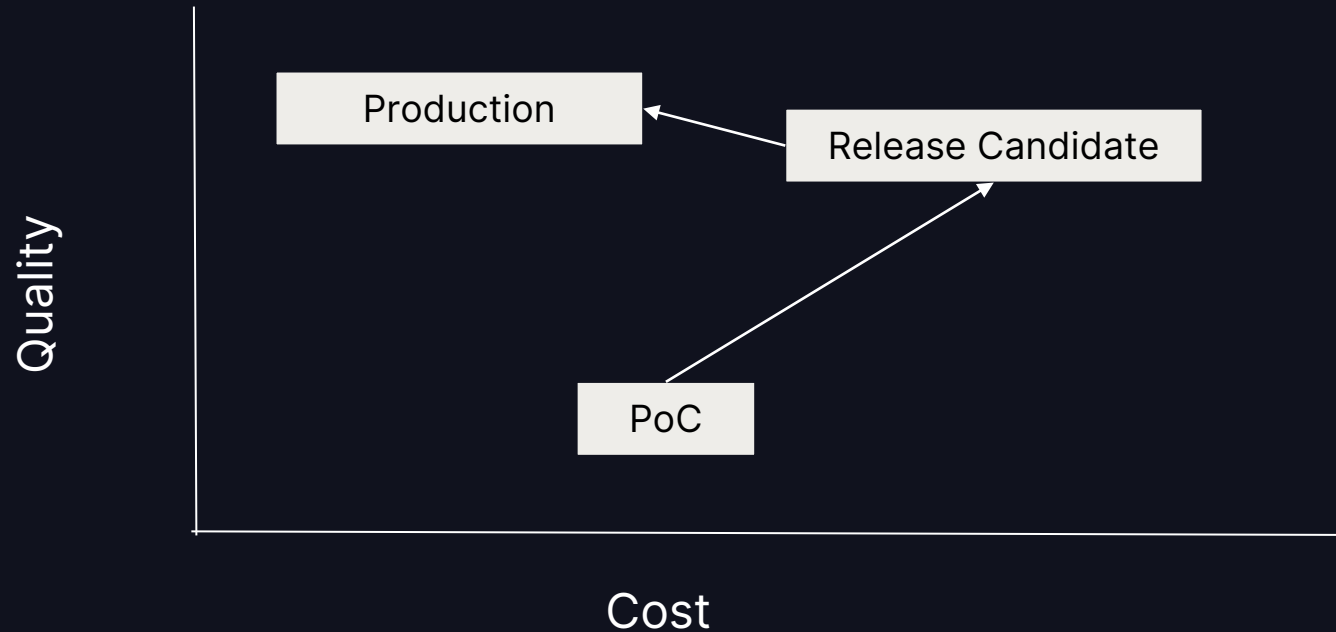
Agenda

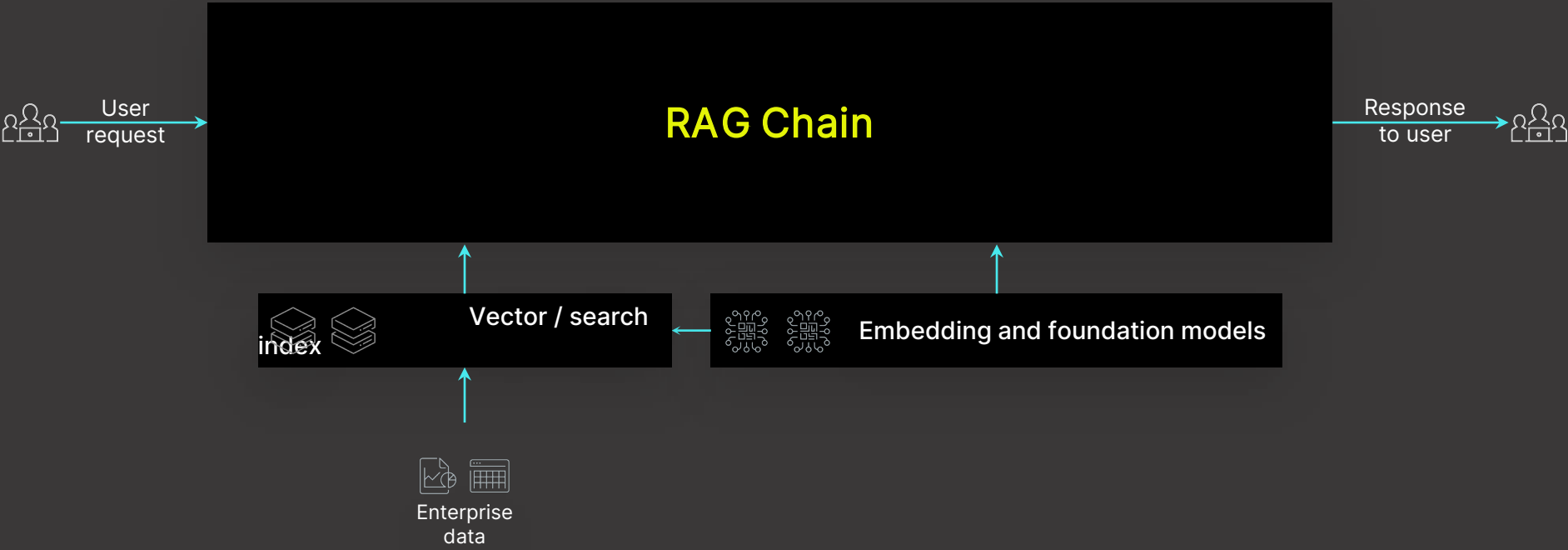
Iterating on quality

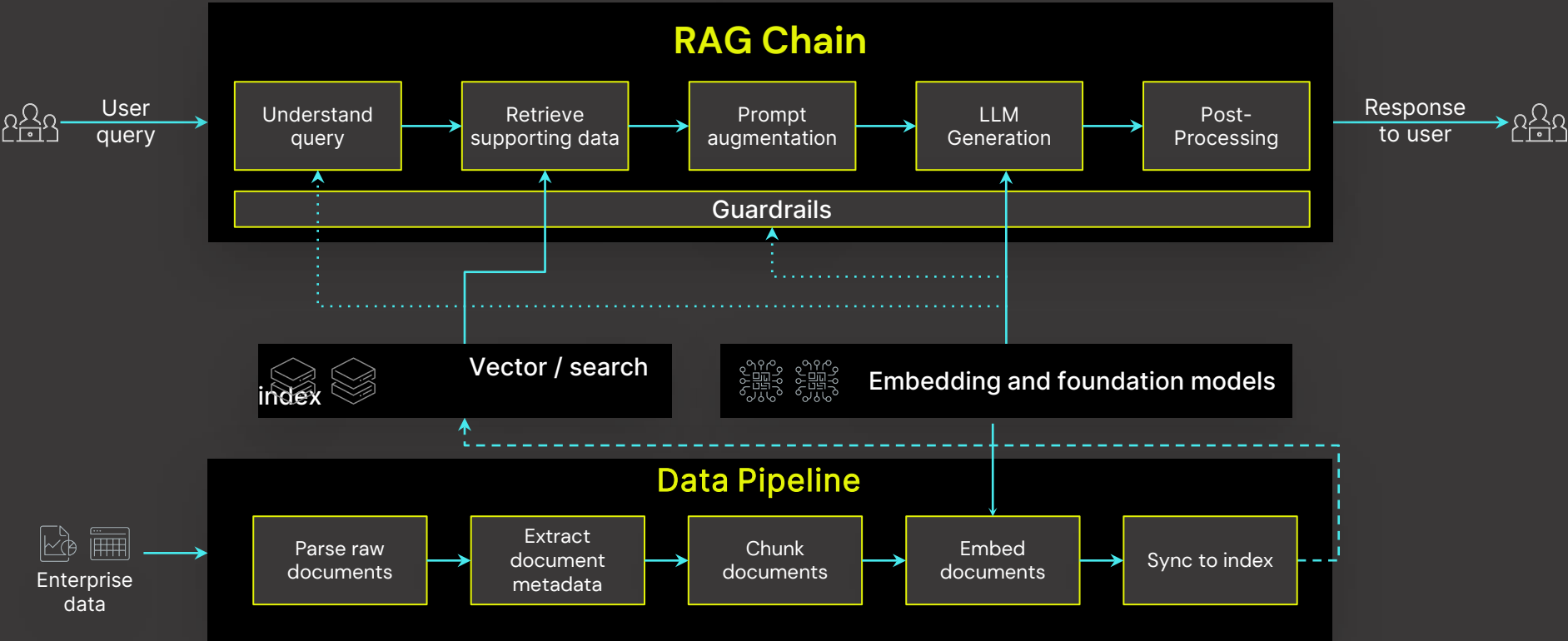
Defining and measuring the quality of GenAI apps

Building high-quality GenAI apps in Databricks : Mosaic AI Agent Evaluation

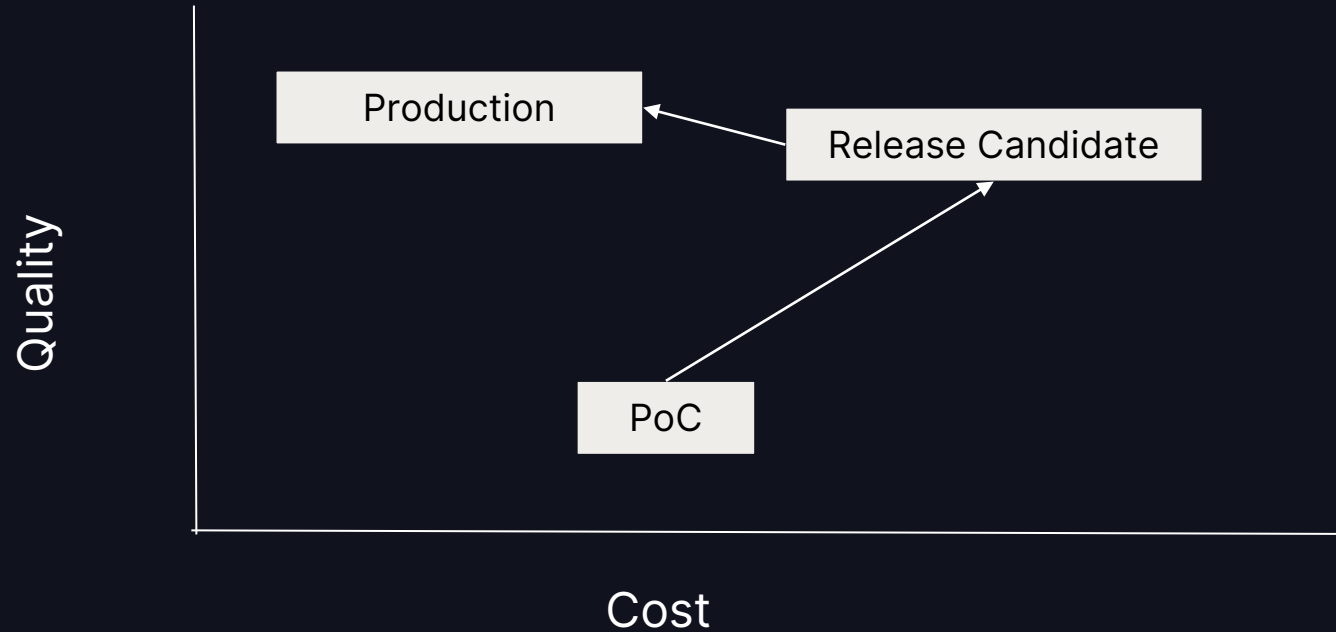
Iterating GenAI App Quality



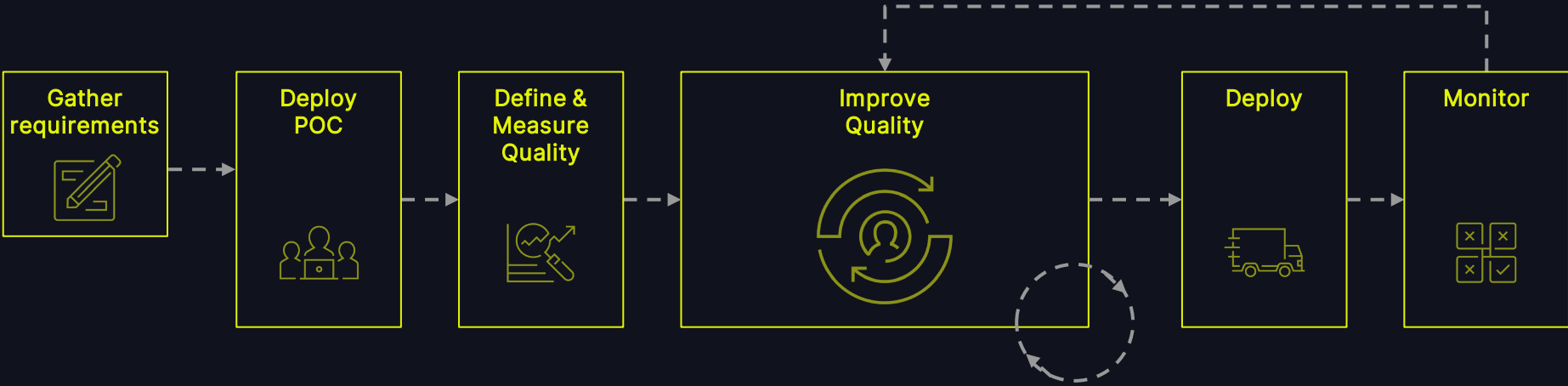




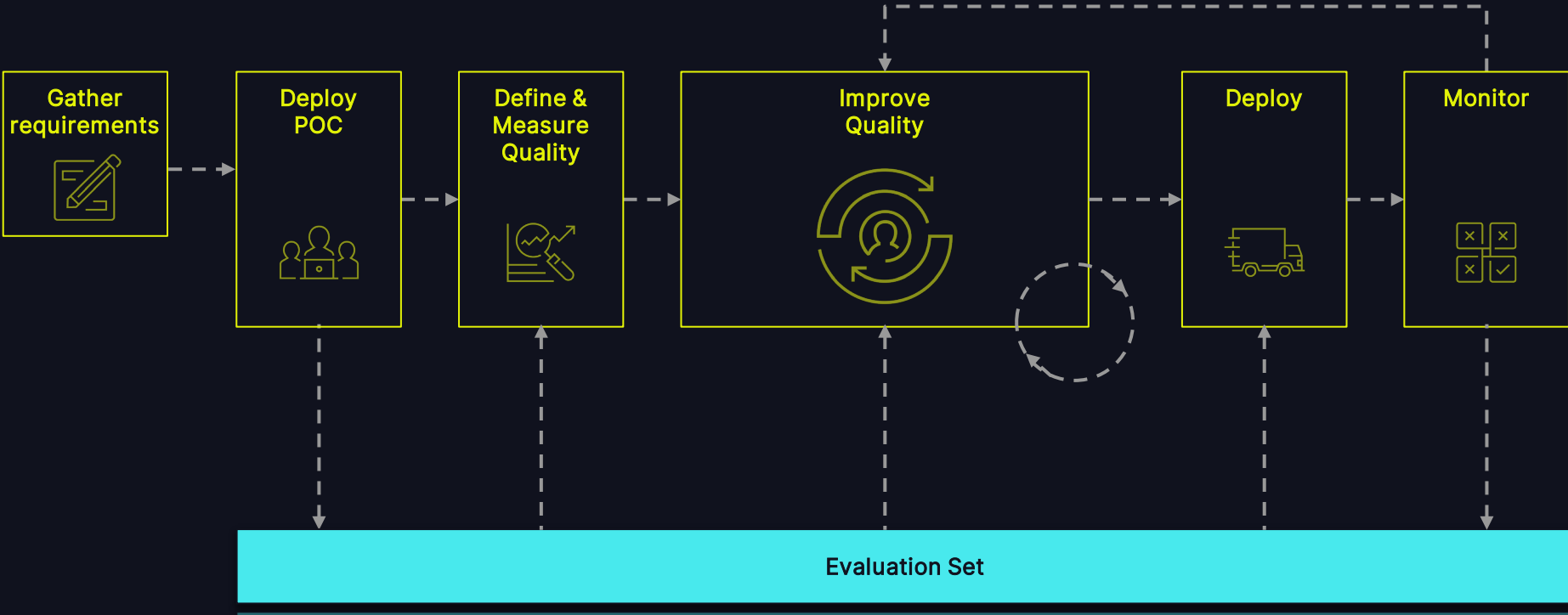
Iterating GenAI App Quality



Evaluation-Driven Development



Evaluation-Driven Development



Evaluation Set

The foundation for measuring quality: expected system input and outputs

Representative: reflect the variety of requests to be seen in production.

Challenging: reflect difficult and diverse cases to effectively test the model's capabilities, including malformed or adversarial inputs (truncation, unsafe).

Continually updated: reflect changing nature production data, and any changes to the application requirements.

Evaluation Set

Encodes requirements and behaviors

“Write pyspark code which joins two tables”

To join two tables in PySpark, you can use the following example code:

```
employees = [
    (1, "John", 1),
    (2, "Emma", 2),
    (3, "Raj", None),
    (4, "Nina", 4),
]

departments = [
    (1, "HR"),
    (2, "Tech"),
    (3, "Marketing"),
    (None, "Temp"),
]

df_employees = spark.createDataFrame(employees, ["emp_id", "emp_name", "dept_id"])
df_departments = spark.createDataFrame(departments, ["dept_id", "dept_name"])
# Perform inner join, which is the default
df_joined = df_employees.join(df_departments, df_employees.dept_id == df_departments.dept_id)
```

Curate with Mosaic AI Agent Evaluation

To join two tables in PySpark along with dummy data, you can follow the steps below. This example demonstrates how to create two DataFrames, perform an inner join, and display the results.

```
from pyspark.sql import SparkSession
from pyspark.sql import Row

# Create dummy data for the first DataFrame
data1 = [
    Row(id=1, name="Alice", age=25),
    Row(id=2, name="Bob", age=30),
    Row(id=3, name="Charlie", age=35)
]

# Create dummy data for the second DataFrame
data2 = [
    Row(id=1, city="New York"),
    Row(id=2, city="Los Angeles"),
    Row(id=4, city="Chicago")
]

# Create DataFrames from the dummy data
df1 = spark.createDataFrame(data1)
df2 = spark.createDataFrame(data2)

# Perform an inner join on the 'id' column
joined_df = df1.join(df2, df1.id == df2.id, "inner")

# Select relevant columns to avoid duplicate 'id' columns
result_df = joined_df.select(df1.id, df1.name, df1.age, df2.city)

# Show the result
result_df.show()

# Stop the Spark session
```

Evaluation Set

Encodes requirements and behaviors

“Write pyspark code which joins two tables”

To join two tables in PySpark, you can use the following example code:

```
employees = [
    (1, "John", 1),
    (2, "Emma", 2),
    (3, "Raj", None),
    (4, "Nina", 4),
]

departments = [
    (1, "HR"),
    (2, "Tech"),
    (3, "Marketing"),
    (None, "Temp"),
]

df_employees = spark.createDataFrame(employees, ["emp_id", "emp_name", "dept_id"])
df_departments = spark.createDataFrame(departments, ["dept_id", "dept_name"])
# Perform inner join, which is the default
df_joined = df_employees.join(df_departments, df_employees.dept_id == df_departments.dept_id)
```

Curate with Mosaic AI Agent Evaluation

To join two tables in PySpark along with dummy data, you can follow the steps below. This example demonstrates how to create two DataFrames, perform an inner join, and display the results.

```
from pyspark.sql import SparkSession
from pyspark.sql import Row

# Create dummy data for the first DataFrame
data1 = [
    Row(id=1, name="Alice", age=25),
    Row(id=2, name="Bob", age=30),
    Row(id=3, name="Charlie", age=35)
]

# Create dummy data for the second DataFrame
data2 = [
    Row(id=1, city="New York"),
    Row(id=2, city="Los Angeles"),
    Row(id=4, city="Chicago")
]

# Create DataFrames from the dummy data
df1 = spark.createDataFrame(data1, ["id", "name", "age"])
df2 = spark.createDataFrame(data2, ["id", "city"])
```

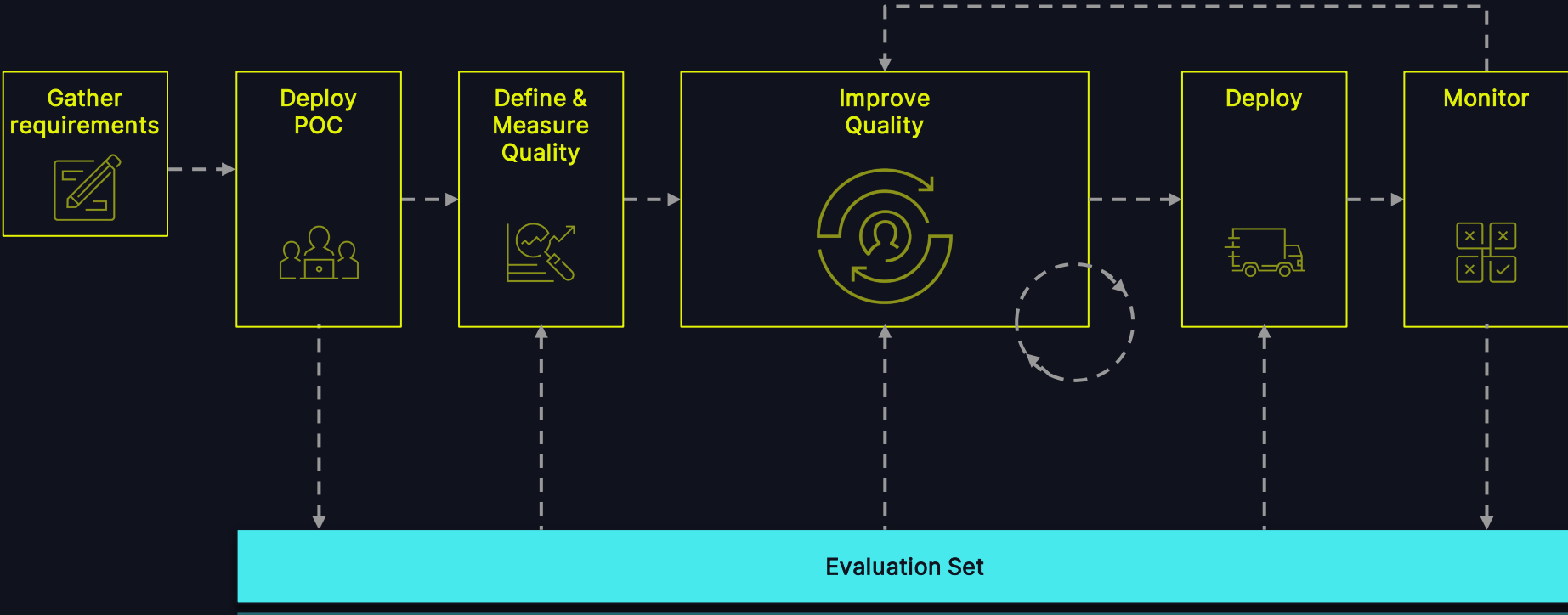
Explanation:

```
df1 = spark.createDataFrame(data1, ["id", "name", "age"])
df2 = spark.createDataFrame(data2, ["id", "city"])
# Perform inner join
joined_df = df1.join(df2, df1.id == df2.id, how="inner")
# Select relevant columns to avoid duplicate 'id' columns
result_df = joined_df.select("name", "age", "city")
# Show the result
result_df.show()
# Stop the Spark session
spark.stop()
```

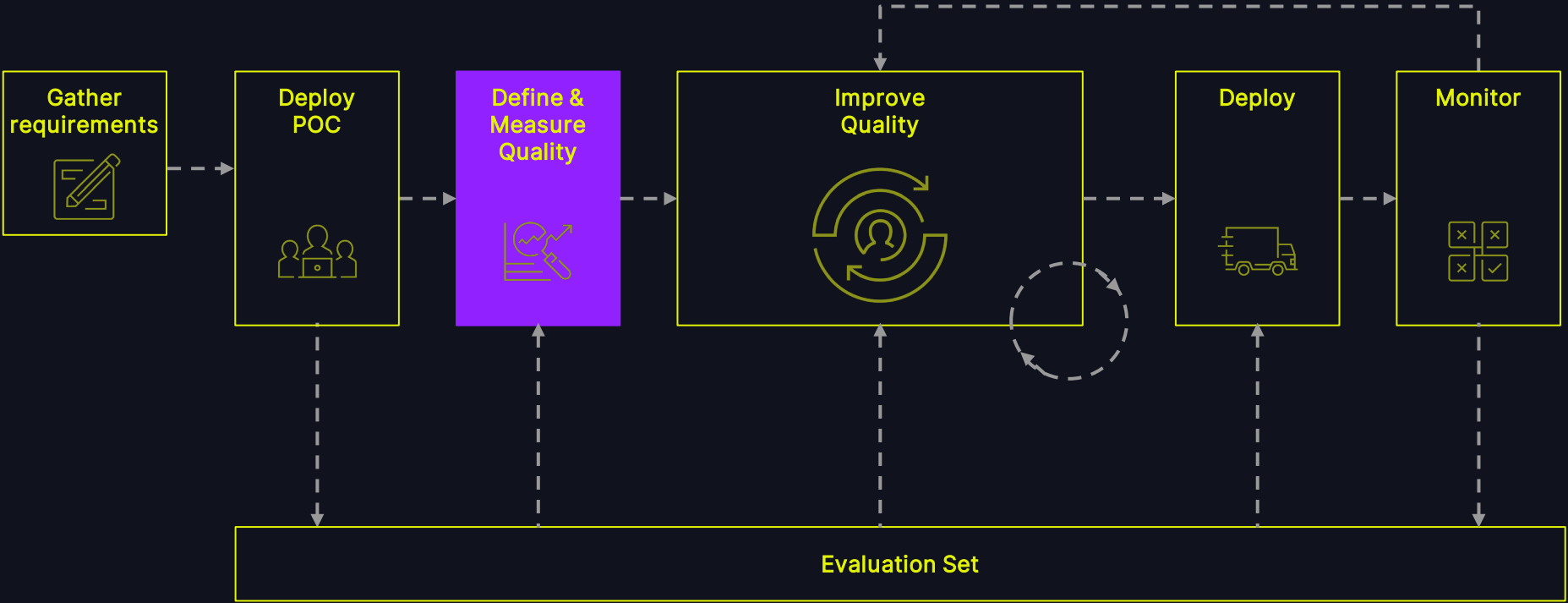
- 1. Initialize Spark Session:** Start by creating a Spark session.
- 2. Create Dummy Data:** Define two sets of dummy data using the `Row` class.
- 3. Create DataFrames:** Convert the dummy data into DataFrames.
- 4. Perform Join:** Use the `join` method to perform an inner join on the 'id' column.
- 5. Select Columns:** Select the relevant columns to avoid duplicate 'id' columns in the result.
- 6. Show Result:** Display the joined DataFrame.
- 7. Stop Spark Session:** Stop the Spark session to free up resources.

This code will output the joined DataFrame, showing only the rows with matching 'id' values from both DataFrames.

Evaluation-Driven Development



Evaluation-Driven Development



Evaluation-Driven Development

Measuring Quality

Data: Completeness, Consistency

Retrieval: Relevance, Precision, Recall

Response: Correctness, Groundedness,
Safety, Relevance

System Performance: Latency, Cost



Databricks
Generative AI Cookbook

Retrieval Quality

Measuring Quality

Relevance, Precision, Recall

While RAG is new, retrieval is an old problem with these measurement concepts dating to the 1950s. [1]

[1] Perry, et al. Machine literature searching X. Machine language; factors underlying its design and development, 1955

Relevance

Is a document relevant to the question? (by example [1])

Which board game was published most recently, Pirate's Cove or Catan?

'The Settlers of Catan, sometimes shortened to Catan or Settlers, is a multiplayer board game designed by Klaus Teuber and first published in 1995 in Germany by Franckh-Kosmos Verlag (Kosmos) as Die Siedler von Catan.

American architect Alfred Mosher Butts invented the game in 1938. Scrabble is produced in the United States and Canada by Hasbro, under the brands of both of its subsidiaries, Milton Bradley and Parker Brothers.

Pirate's Cove (in German, Piratenbucht) is a board game designed by Paul Randles and Daniel Stahl, originally published in Germany in 2002 by Amigo Spiele, illustrated by Markus Wagner and Swen Papenbrock.

The Settlers of Catan became one of the first German-style board games to become popular outside Europe.[1] As of 2015, more than 22 million copies in 30 languages had been sold.

[1] HOTPOTQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. Yang, Qi, Zhang, et al., Empirical Methods in Natural Language Processing, 2018

Precision and Recall

What is the quality of my retrieval?

Precision - what % of the retrieved documents are relevant to the request?

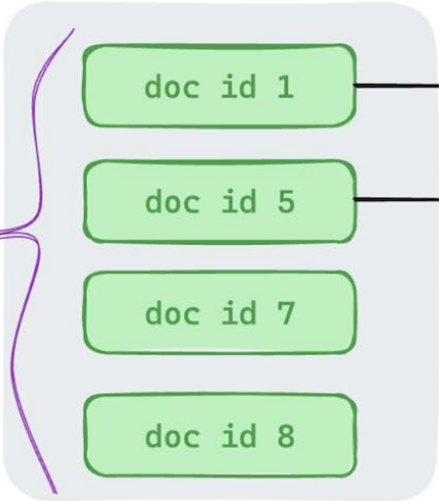
$$\text{Precision} = \frac{\text{\# of relevant retrieved items}}{\text{\# of total retrieved items}}$$

Recall - what % of the required documents are retrieved?

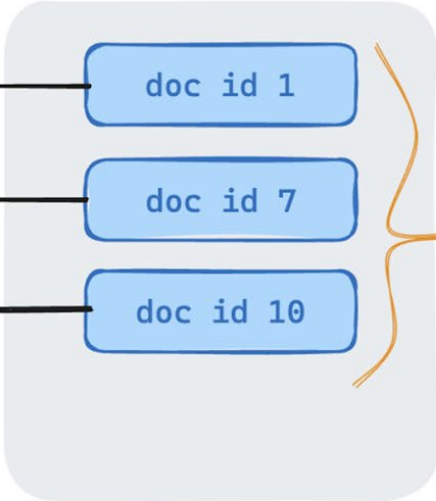
$$\text{Recall} = \frac{\text{\# of relevant retrieved items}}{\text{\# of total relevant items}}$$

Precision and Recall

Relevant documents
"Documents that are known to be needed to answer this query"



Retrieved documents
"Documents that were retrieved by the Vector DB for this query"



$$\text{Precision} = \frac{2}{3}$$

$$\text{Recall} = \frac{2}{4}$$

Response Quality

What is the quality of the system's generation?

Groundedness – is the answer supported by evidence in the retrieved context?

Correctness – is the answer correct relative to a ground truth answer?

Safety – does the answer contain any toxic/harmful content?

Relevance – is the answer relevant to the question?

Groundedness and Correctness

Example

Which board game was published most recently, Pirate's Cove or Catan?

Retrieved Documents

'The Settlers of Catan, sometimes shortened to Catan or Settlers, is a multiplayer board game designed by Klaus Teuber and first published in 1995 in Germany by Franckh-Kosmos Verlag (Kosmos) as Die Siedler von Catan.

Pirate's Cove (in German, Piratenbucht) is a board game designed by Paul Randles and Daniel Stahl, originally published in Germany in 2002 by Amigo Spiele, illustrated by Markus Wagner and Swen Papenbrock.

Answer	Grounded	Correct
Pirate's Cove	+	+
Catan	+	-
Scrabble	-	-



Groundedness and Correctness

Example

Which board game was published most recently, Pirate's Cove or Catan?

Retrieved Documents

'The Settlers of Catan, sometimes shortened to Catan or Settlers, is a multiplayer board game designed by Klaus Teuber and first published in 1995 in Germany by Franckh-Kosmos Verlag (Kosmos) as Die Siedler von Catan

Answer	Grounded	Correct
Pirate's Cove	-	+
Catan	+	-
Scrabble	-	-



Correctness

Encodes requirements and behaviors

“Write pyspark code which joins two tables”

To join two tables in PySpark, you can use the following example code:

```
employees = [
    (1, "John", 1),
    (2, "Emma", 2),
    (3, "Raj", None),
    (4, "Nina", 4),
]

departments = [
    (1, "HR"),
    (2, "Tech"),
    (3, "Marketing"),
    (None, "Temp"),
]

df_employees = spark.createDataFrame(employees, ["emp_id", "emp_name", "dept_id"])
df_departments = spark.createDataFrame(departments, ["dept_id", "dept_name"])
# Perform inner join, which is the default
df_joined = df_employees.join(df_departments, df_employees.dept_id == df_departments.dept_id)
```

To join two tables in PySpark along with dummy data, you can follow the steps below. This example demonstrates how to create two DataFrames, perform an inner join, and display the results.

```
from pyspark.sql import SparkSession
from pyspark.sql import Row

# Create dummy data for the first DataFrame
data1 = [
    Row(id=1, name="Alice", age=25),
    Row(id=2, name="Bob", age=30),
    Row(id=3, name="Charlie", age=35)
]

# Create dummy data for the second DataFrame
data2 = [
    Row(id=1, city="New York"),
    Row(id=2, city="Los Angeles"),
    Row(id=4, city="Chicago")
]

# Create DataFrames from the dummy data
df1 = spark.createDataFrame(data1, ["id", "name", "age"])
df2 = spark.createDataFrame(data2, ["id", "city"])

# Perform inner join
joined_df = df1.join(df2, df1.id == df2.id, how="inner")

# Select relevant columns to avoid duplicate 'id' columns in the result
result_df = joined_df.select("name", "age", "city")

# Show the result
result_df.show()

# Stop the Spark session
spark.stop()
```

Explanation:

1. **Initialize Spark Session:** Start by creating a Spark session.
2. **Create Dummy Data:** Define two sets of dummy data using the `Row` class.
3. **Create DataFrames:** Convert the dummy data into DataFrames.
4. **Perform Join:** Use the `join` method to perform an inner join on the 'id' column.
5. **Select Columns:** Select the relevant columns to avoid duplicate 'id' columns in the result.
6. **Show Result:** Display the joined DataFrame.
7. **Stop Spark Session:** Stop the Spark session to free up resources.

This code will output the joined DataFrame, showing only the rows with matching 'id' values from both DataFrames.

How to implement Metrics?

LLMs can serve as useful, scalable implementations of metrics [1,2]

Prompt:

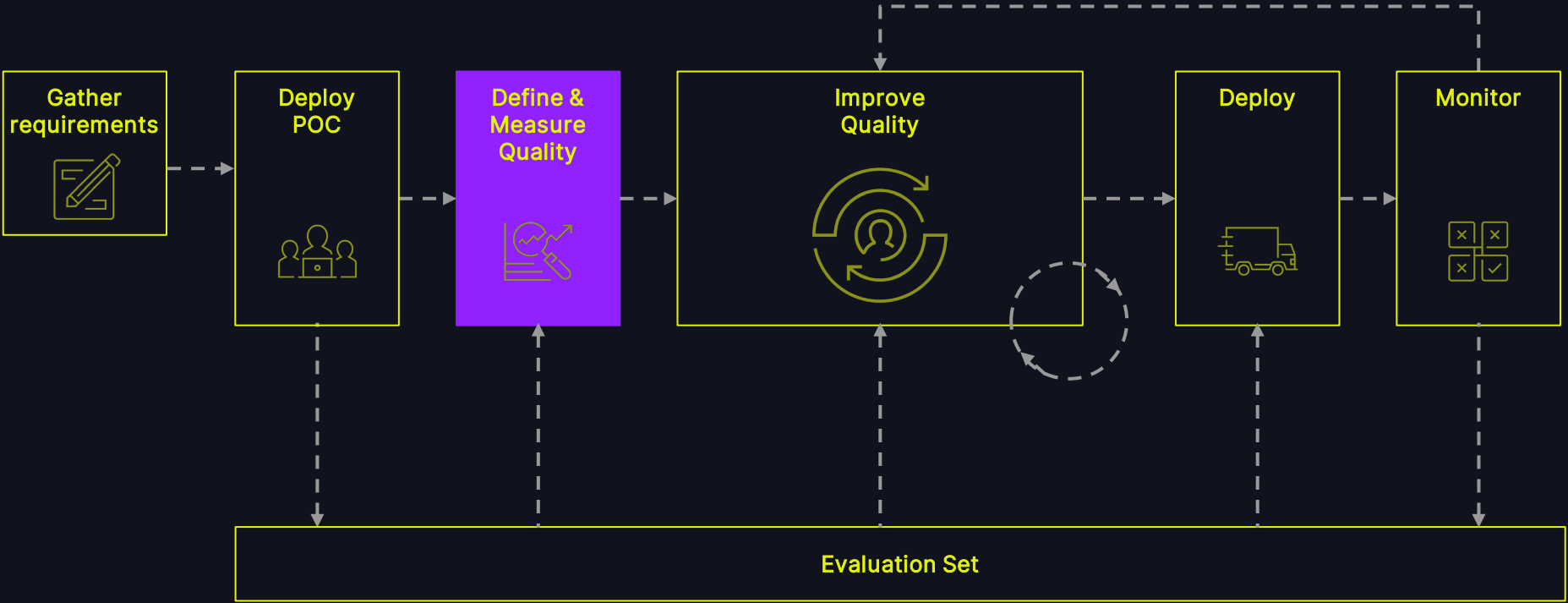
Given the following question and document, analyze the provided document and determine whether it is relevant for answering the question.
Question: {question}
Document: {document}

Agent Evaluation leverages LLMs-as-Judge to offer built-in metrics for Precision, Groundedness, Answer Correctness, and more.

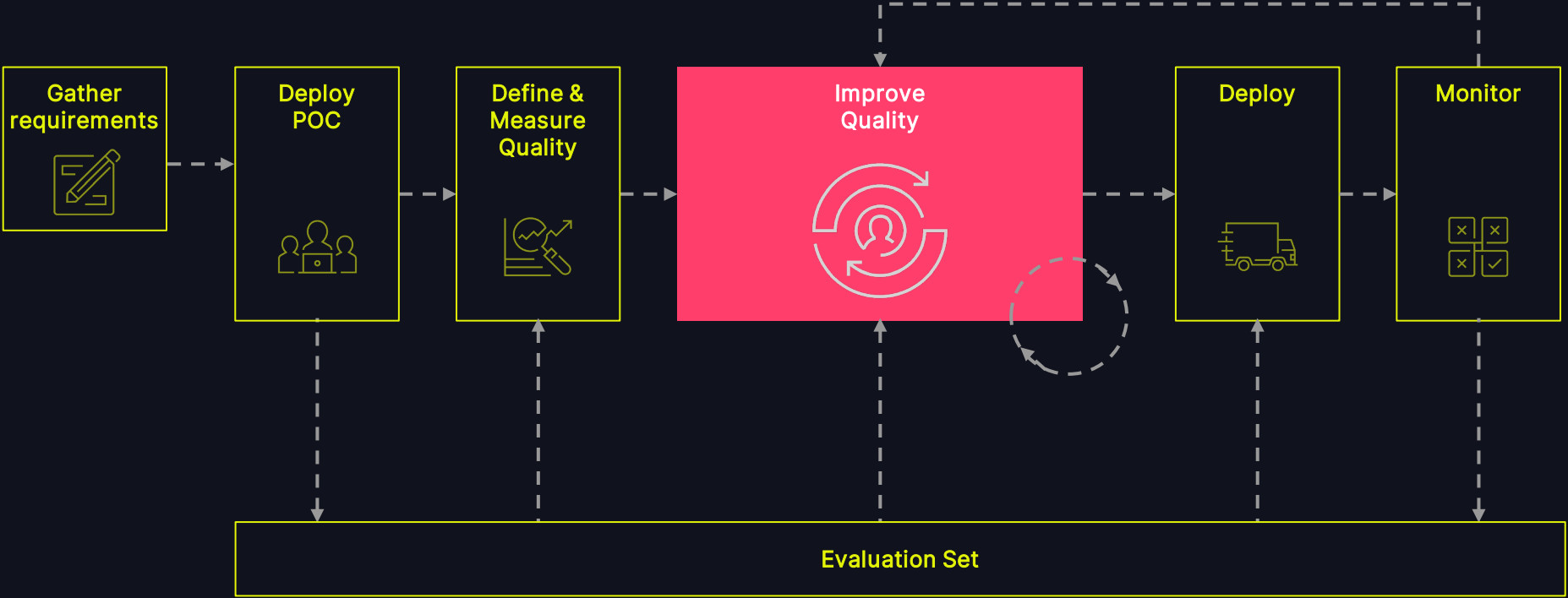
[1] Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena.
Zheng, et al., NeurIPS Datasets and Benchmarks Track, 2023

[2] ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems
Saad-Falcon, et. al, North American Chapter of the Association for Computational Linguistics, 2024

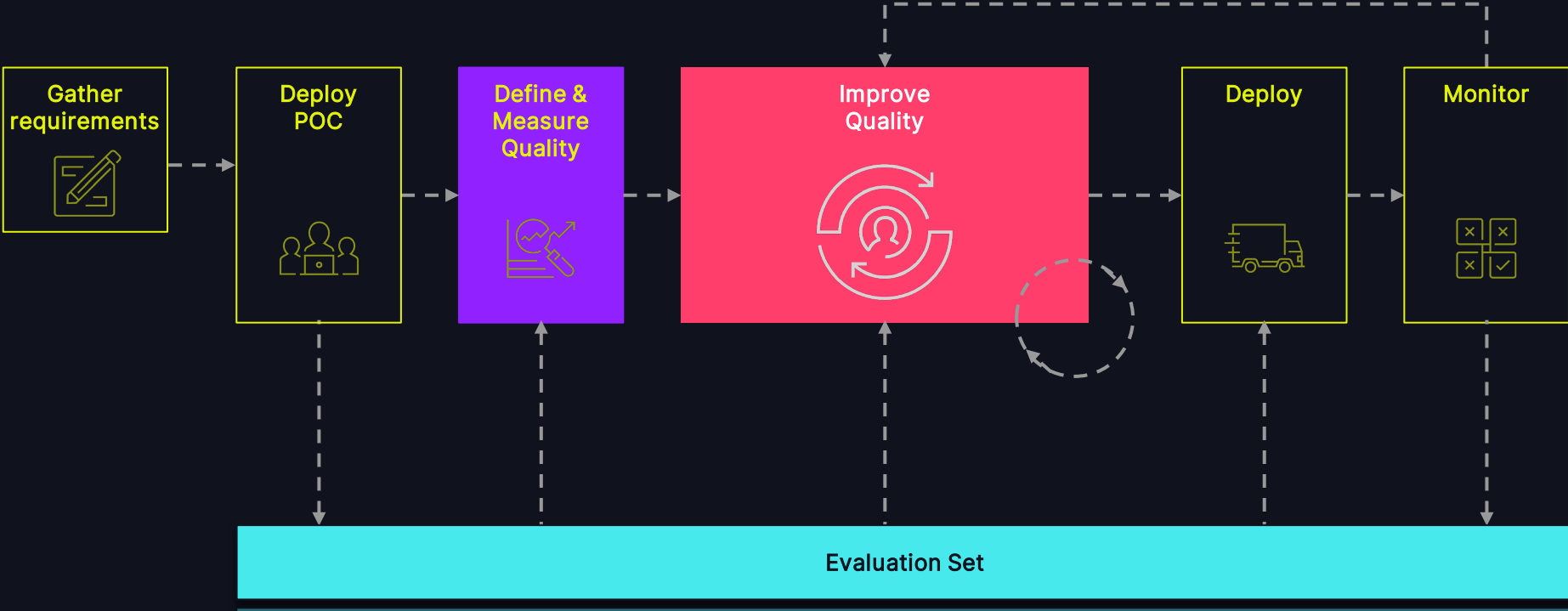
Evaluation-Driven Development



Evaluation-Driven Development



Evaluation-Driven Development

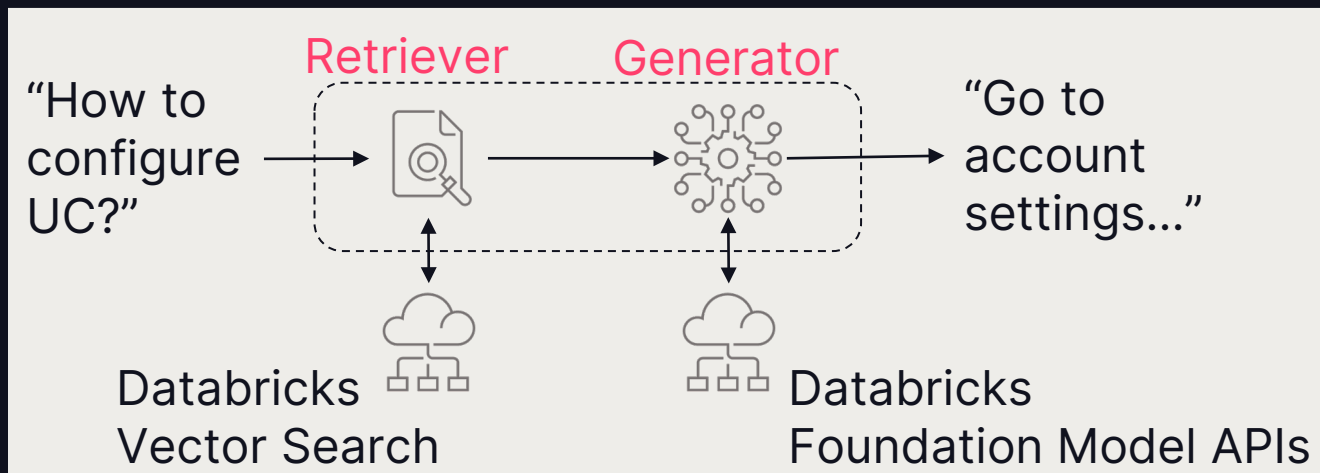


Mosaic AI

Agent Evaluation
Agent Framework

Example app: Databricks Docbot

A RAGbot to help our field engineers



Build this yourself!

Recap

Review App: Gather feedback from stakeholders, bootstrap evaluation set

Built-in LLM judges: Measure quality, guide iterations for improvement

Agent Serving & Lakehouse Monitoring: Deploy and monitor in production

Mosaic AI Agent Evaluation + Framework

Build and deploy high-quality GenAI apps in Databricks

Now in Public Preview!

