Agent driven GenAI applications

that understand and react to user intent…

Anindita Mahapatra

Introductions

Lead Solutions Architect – Databricks

“If Data has arrived, it better be served!”

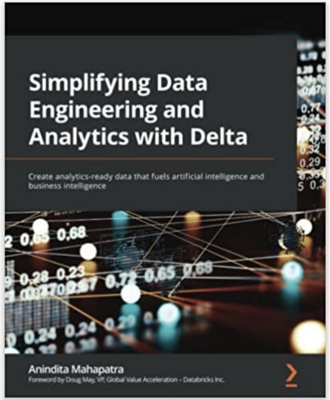
- Past Experience in Big Data
  - Teradata/Think Big Analytics
  - Nokia/Microsoft

- MS in Computer Science - Boston University
- Master of Liberal Arts & Management - Harvard Extension

- I teach a graduate course on Data Engineering (CSCI E-103) at the Harvard Extension School.

- I've authored the book “Simplifying Data Engineering and Analytics with Delta: Create analytics-ready data that fuels artificial intelligence and business intelligence”

Anindita Mahapatra


Simplifying Data Engineering and Analytics with Delta
Create analytics-ready data that fuels artificial intelligence and business intelligence
Anindita Mahapatra
Foreword by Doug May, VP, Global Value Acceleration - Databricks Inc.

ISBN-13: 978-1801814867
ISBN-10: 1801814864

Agenda

- Compound Systems
  - Types
  - Characteristics
  - Components: chains, routers, agents
  - Examples
- Agentic Architecture
- Using Databricks as a platform to understand agents in DE & ML Pipelines
  - What are the challenges
  - RAG is a compound system

Compound System

Compound systems in the GenAI era represent a leap forward in how AI is used to solve complex, multi-domain problems.

By integrating multiple AI models, external tools, and dynamic orchestration mechanisms, these systems can autonomously adapt, learn, and scale across diverse use cases. Their versatility makes them especially valuable in fields requiring complex data processing, automation, and decision-making at scale.

The shift from models to compound systems Link

Types of Compound Systems

- Hybrid AI systems (integrating multiple AI architectures or models into one cohesive system)
  - Multi-modal AI
    - Eg. GPT-4 and CLIP can work together to understand and generate text based on images.
  - Hierarchical AI
    - high-level models handle overarching tasks (e.g., decision-making or planning) and
    - lower-level models perform specific functions (e.g., solving specific subproblems)
- AI Agents and LLM Orchestration (intelligent routing of tasks to specialized models or tools)
  - Chained Agents (Eg. Using **chain-of-thought reasoning)**
    - For example, an LLM may generate a SQL query, call an external database API to retrieve data, and then use a machine learning model to analyze that data.
  - Dynamic Pipelines
    - automatically create and adjust data or task pipelines by chaining together models or APIs.

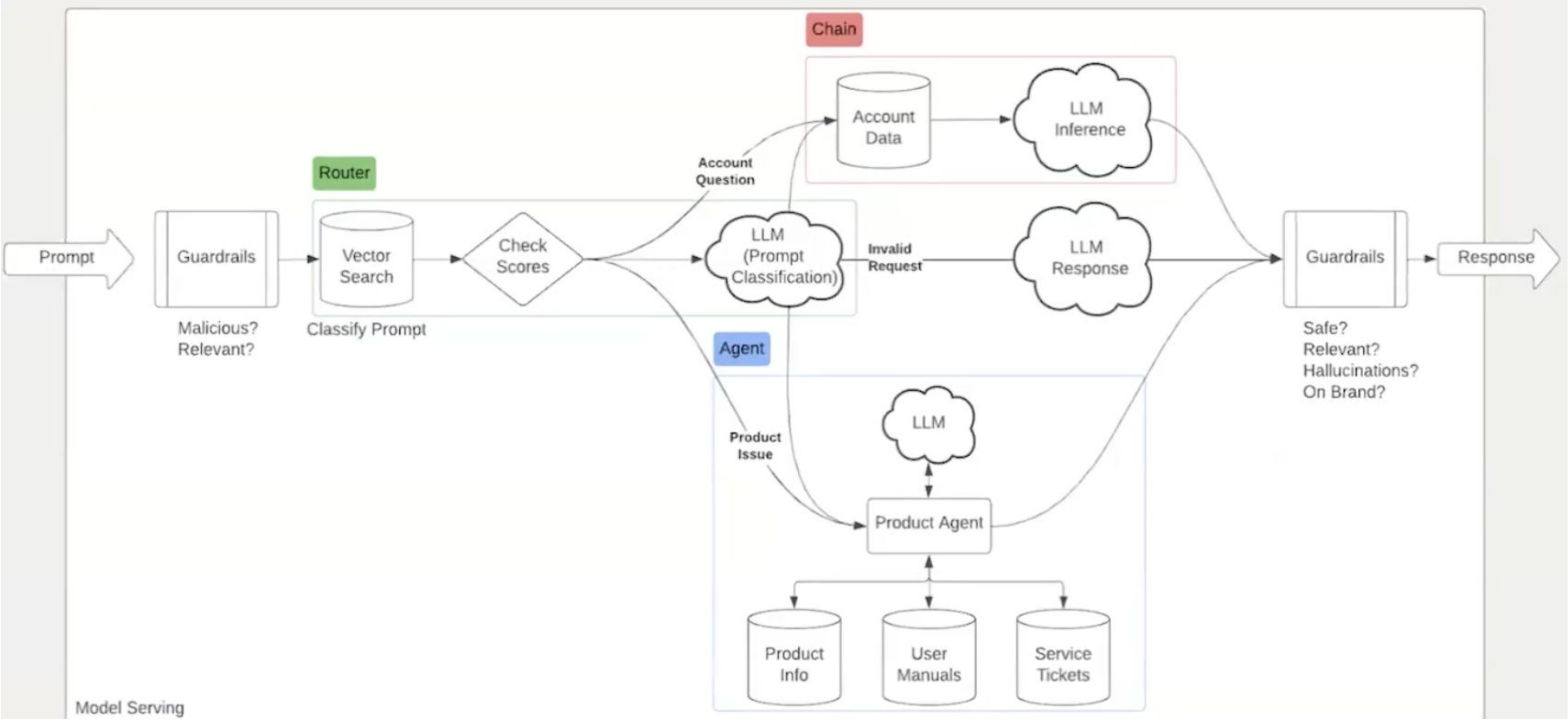
- Ensemble Learning and Model Fusion
  - Ensemble of Models:
    - for different tasks
  - Model Fusion:
    - Combining the outputs of different models to generate a single output or decision. For instance, a system might fuse predictions from a natural language model, a recommendation system, and a sentiment analysis model to make more nuanced suggestions or decisions.
- AI Agents with Integrated Tools
  - LLM agents like **AutoGPT** and **LangChain** can dynamically call external tools such as web search engines, calculators, code execution environments, or APIs to retrieve information, perform calculations, or manipulate data.
- Domain-Specific GenAI Platforms
  - **Healthcare**: Combine LLMs for natural language understanding (e.g., parsing patient reports), computer vision for analyzing medical images, and specialized diagnostic models to recommend treatment plans.
  - **Financial Services**: Combine natural language processing models for sentiment analysis, time-series models for forecasting, and algorithmic trading bots for executing strategies.
- Adaptive and Autonomous Systems
  - By combining models that handle different types of contextual understanding (e.g., location, user behavior, time), compound systems can produce more personalized or relevant outcomes.

- ## Generative & Discriminative model Integration
  - **Dual-Model Systems**: In one example, a generative model might generate new product descriptions for an e-commerce site, while a discriminative model reviews the content for accuracy, compliance, and tone. The system iterates, refining the output before final publication.
  - **AI Feedback Loops**: A compound system might use generative models to create data (e.g., synthetic data for training), which is then fed into a discriminative model for improving its predictions or classifications.
- ## Human AI Collaborative System
  - **AI-Assisted Creative Tools**: A compound system might combine an LLM for text generation, a generative model for image creation, and a recommendation model for suggesting improvements. This enables content creators to collaborate with AI in producing marketing content, blog posts, or design assets.
  - **Decision Support Systems**: In domains like healthcare, finance, or manufacturing, AI models can support human decision-making by analyzing large datasets, generating insights, and proposing actions, with human experts providing the final judgment.
- ## Composable AI Platforms
  - **Plug-and-Play AI Components**: Platforms like **Hugging Face**, **LangChain**, or **Haystack** allow users to mix and match AI components to build customized solutions. A user can combine a language model with a search engine, then integrate a summarization model or a document retriever for a complete pipeline.

Key Characteristics of Compound System

- **Modularity**: Different AI components are combined to perform distinct tasks (e.g., data ingestion, preprocessing, inference, etc.), making the system more flexible and customizable.
- **Interoperability**: AI models and external tools communicate seamlessly, allowing the system to perform complex workflows involving both AI-generated content and traditional data processing.
- **Autonomy**: Compound systems are often designed to operate autonomously, dynamically adjusting workflows based on real-time inputs, performance, and contextual data.
- **Scalability**: These systems can scale across various industries and use cases, thanks to their ability to integrate different models and technologies.

Example: Customer Support

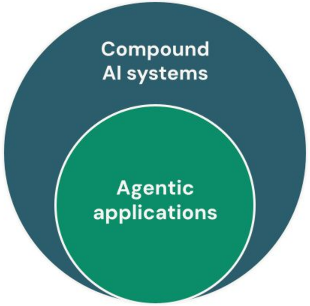


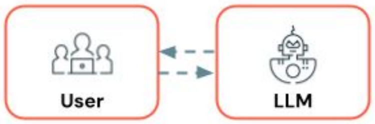
Example of a Compound System in the GenAI era

Imagine a **business analytics platform** using a compound system:

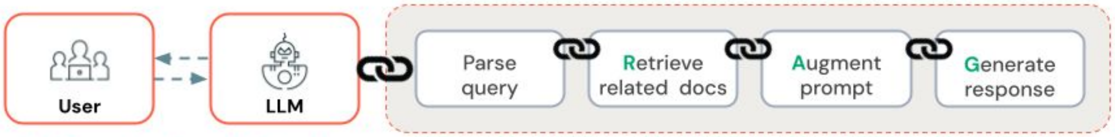
- **Data Ingestion**: LLM agents automatically connect to different data sources (e.g., CRM, social media, financial reports) and extract relevant data.
- **Analysis**: A combination of time-series forecasting models, sentiment analysis models, and recommendation engines works together to generate insights about customer behavior and future sales trends.
- **Decision-Making**: A generative model produces reports summarizing key insights in natural language, while a reinforcement learning agent suggests strategies based on historical data and simulations.
- **Execution**: Finally, the system interfaces with an automation platform (e.g., marketing automation or sales platforms) to execute the strategies.

From LLMs to Agents

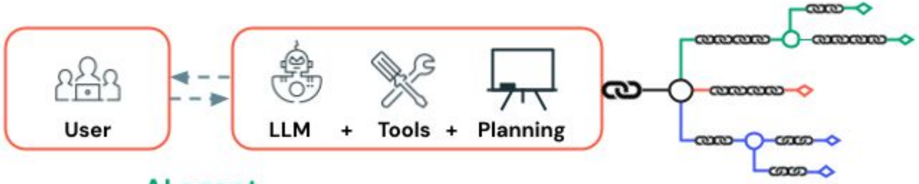




Large Language Models (LLM)



LLMs + hardcoded tools



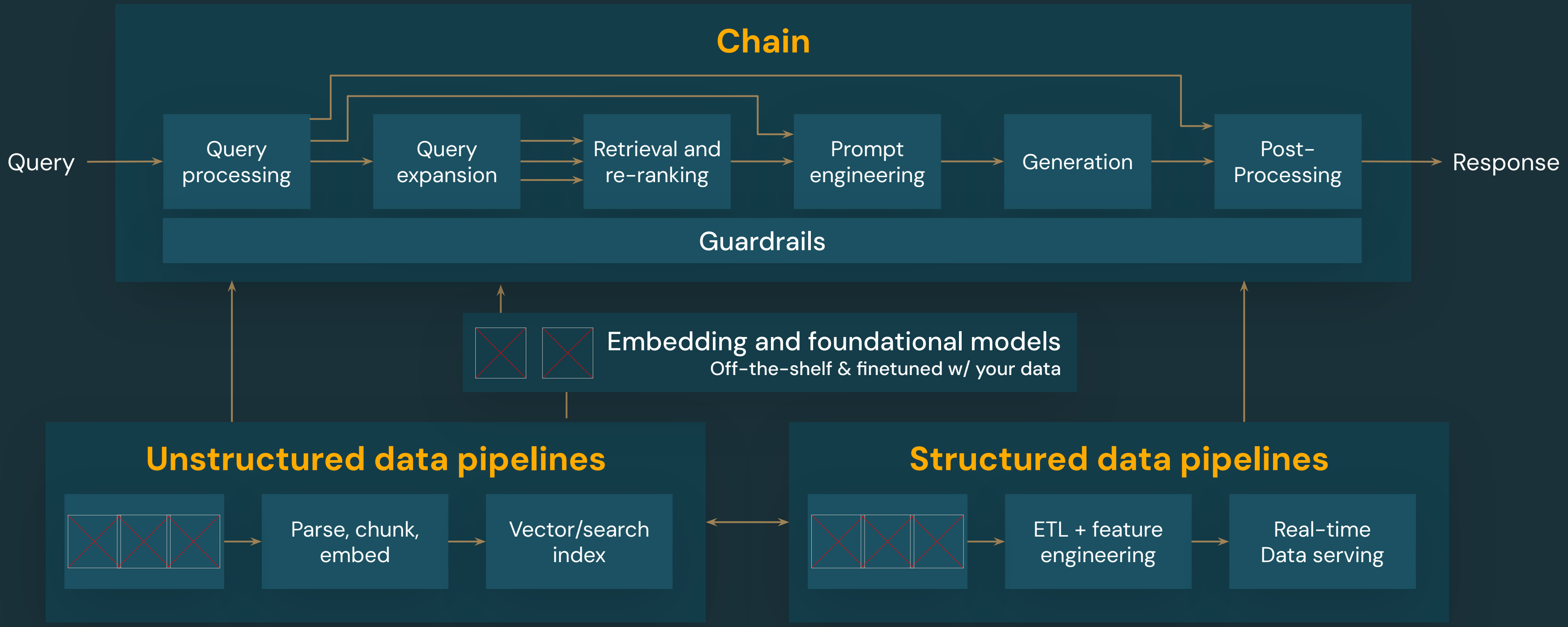
AI agent

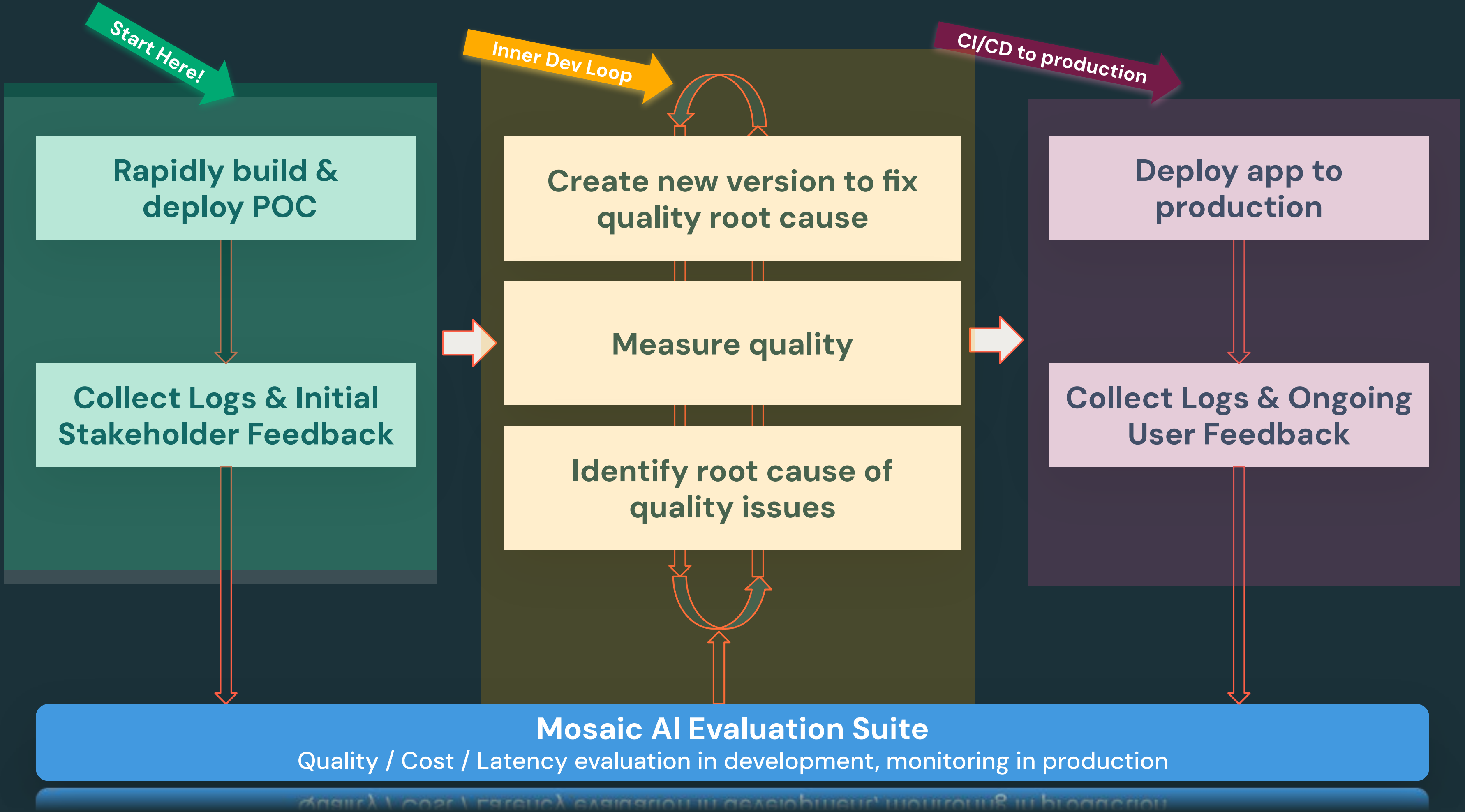
Now, AI agents autonomously create plans and execute tasks based on their understanding of the problem. AI agents still use tools but it's up to them to decide which tool to use and when. The key distinction is in the level of autonomy and decision-making capabilities compared to compound AI systems.

Agentic Architecture

- System design paradigm in which multiple autonomous agents operate and collaborate within a larger system to achieve specific goals.
- In this architecture, agents are software entities with specialized tasks and capabilities, capable of making independent decisions, interacting with other agents, and adapting their behavior based on environmental conditions or system needs.
- In the context of AI, and particularly the **Generative AI (GenAI)** era, agentic architecture is a natural evolution where AI agents (powered by LLMs and other models) are integrated into complex workflows, enabling systems to become more adaptive, interactive, and autonomous.
- Core Principles
    - Autonomy
    - Specialization
    - Communication and Collaboration
    - Goal-Oriented Behavior
    - Modularity and Scalability
    - Adaptability

Quality is a systems problem





Start Here!

Rapidly build & deploy POC

Collect Logs & Initial Stakeholder Feedback

Inner Dev Loop

Create new version to fix quality root cause

Measure quality

Identify root cause of quality issues

CI/CD to production

Deploy app to production

Collect Logs & Ongoing User Feedback

Mosaic AI Evaluation Suite

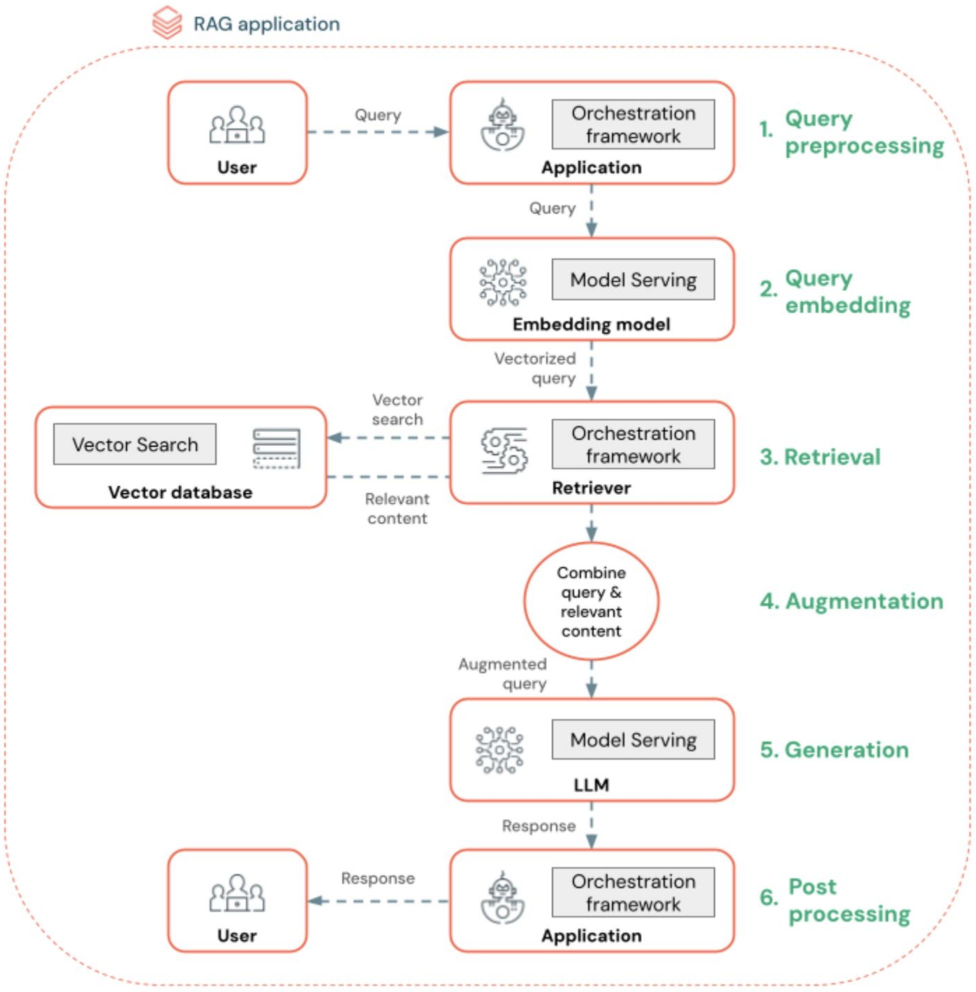
Quality / Cost / Latency evaluation in development, monitoring in production

RAG is an example of Compound System

RAG

Type of system	Components
Prompt engineering	● Prompt generation logic ● LLM
Unstructured docs RAG	● LLM ● Retrieval system
Structured data RAG	● LLM ● Data API e.g., Databricks Online Table ● Text-to-SQL engine e.g., Genie ← *more complicated in reality*
Agent-based Chain	● Function-calling capable LLM ● RAG chain ● Orchestration chain
Orchestration Chain	● Function-calling capable LLM ● API services

Challenges Difficult to …
measure and evaluate bot accuracy
collect adequate feedback
Improve accuracy & reduce hallucination
Enforce guardrails



RAG application

User

Query

Orchestration framework

Application

1. Query preprocessing

Query

Model Serving

Embedding model

2. Query embedding

Vectorized query

Vector Search

Vector database

Vector search

Relevant content

Orchestration framework

Retriever

3. Retrieval

Combine query & relevant content

4. Augmentation

Augmented query

Model Serving

LLM

5. Generation

Response

User

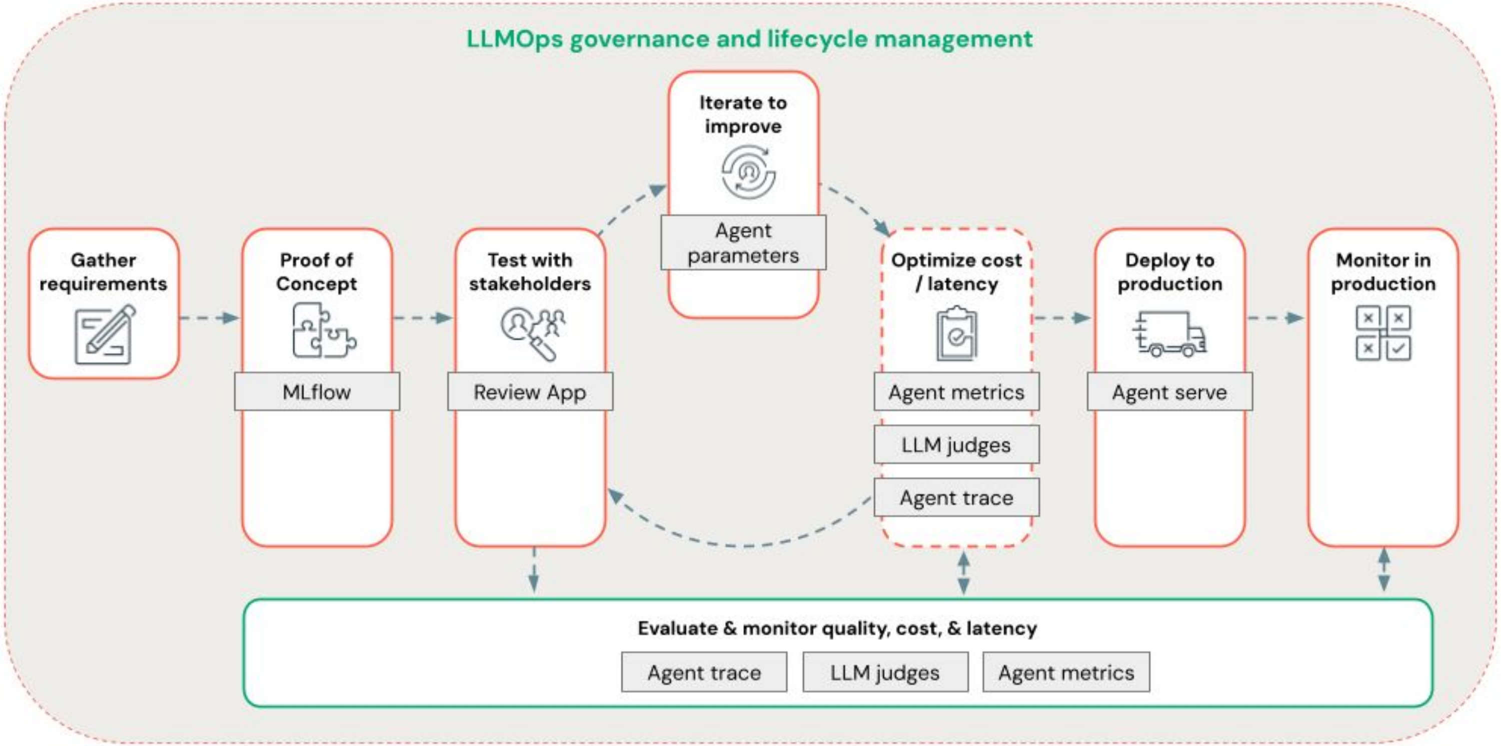
Response

Orchestration framework

Application

6. Post processing

Mosaic AI Agent Framework



Mosaic AI Agent Evaluation

evaluate the quality, cost, and latency of agentic AI applications

Agent Evaluation includes proprietary LLM judges and agent metrics to evaluate retrieval and request quality as well as overall performance metrics like latency and token cost.

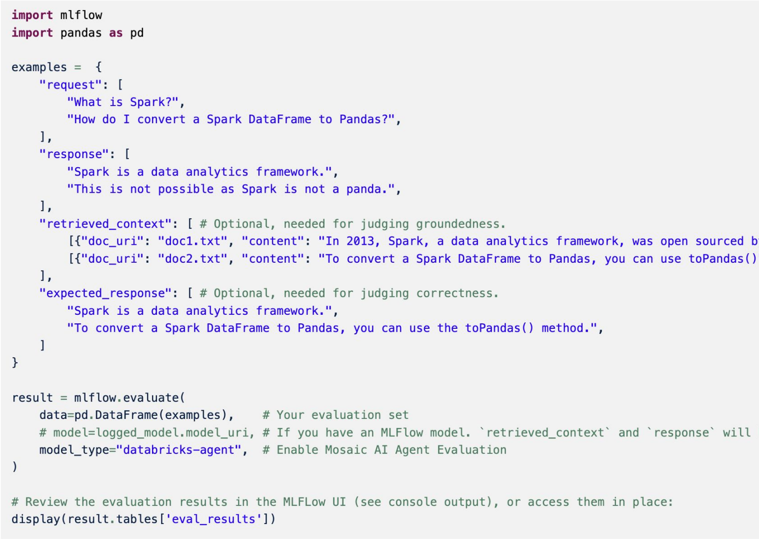
Establish Ground truth with an evaluation set

Online Vs Offline Evaluations

Assess performance with the right metrics

Get human feedback about the quality of a GenAI application

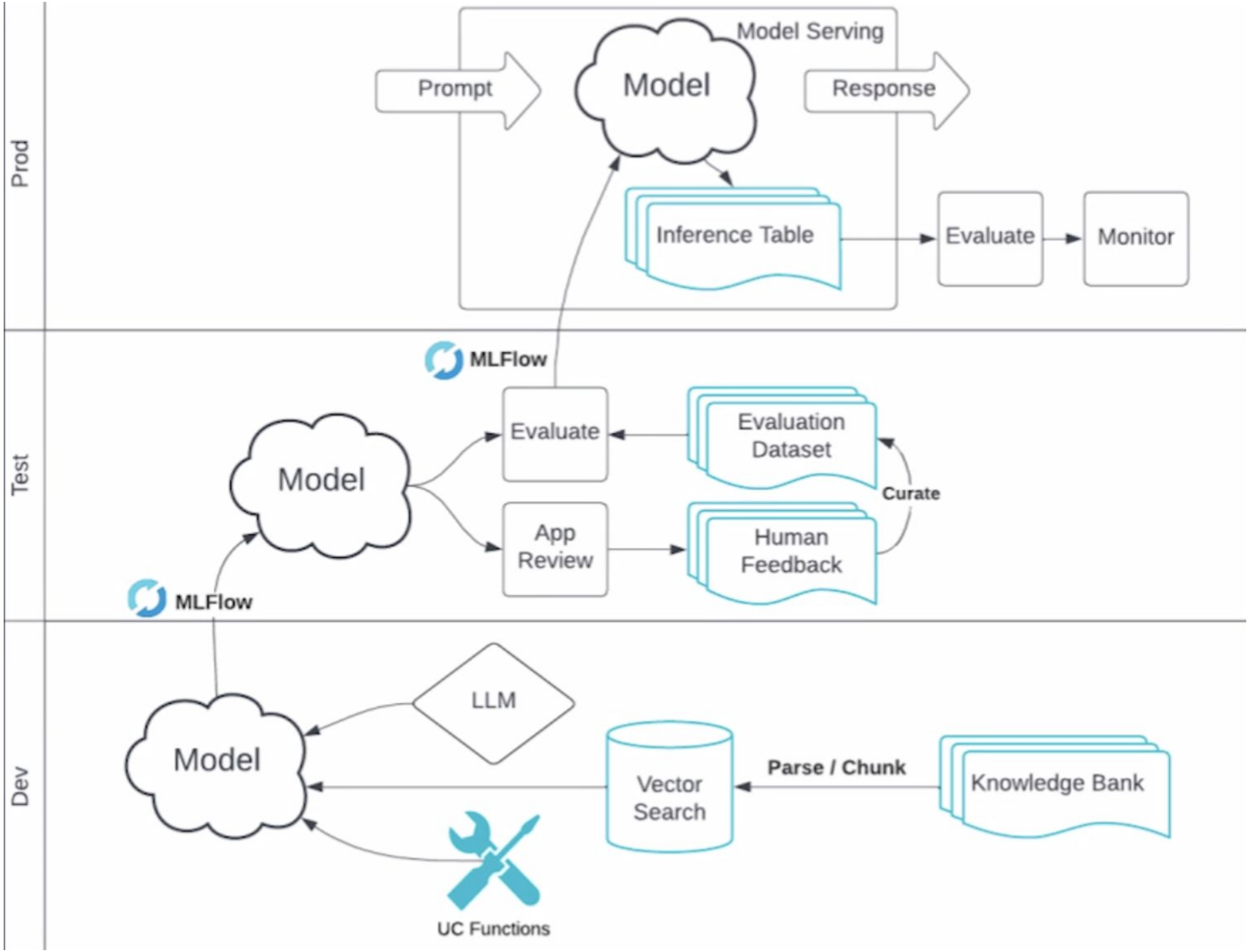
LLM judges are intended to help customers evaluate their RAG applications, and LLM judge outputs should not be used to train, improve, or fine-tune an LLM.



```
import mlflow
import pandas as pd

examples =  {
    "request": [
        "What is Spark?",
        "How do I convert a Spark DataFrame to Pandas?",
    ],
    "response": [
        "Spark is a data analytics framework.",
        "This is not possible as Spark is not a panda.",
    ],
    "retrieved_context": [ # Optional, needed for judging groundedness.
        [{"doc_uri": "doc1.txt", "content": "In 2013, Spark, a data analytics framework, was open sourced b
        [{"doc_uri": "doc2.txt", "content": "To convert a Spark DataFrame to Pandas, you can use toPandas()
    ],
    "expected_response": [ # Optional, needed for judging correctness.
        "Spark is a data analytics framework.",
        "To convert a Spark DataFrame to Pandas, you can use the toPandas() method.",
    ]
}

result = mlflow.evaluate(
    data=pd.DataFrame(examples),    # Your evaluation set
    # model=logged_model.model_uri, # If you have an MLFlow model. `retrieved_context` and `response` will
    model_type="databricks-agent",  # Enable Mosaic AI Agent Evaluation
)

# Review the evaluation results in the MLFLow UI (see console output), or access them in place:
display(result.tables['eval_results'])
```

LLM Ops

- Rapid Development
- Human Feedback
- Automated Evaluation
- A/B Testing
- Auditing/Tracing
- Monitoring



Demo

AI/BI

Databricks AI Cookbook

Demo Link

- Mosaic AI Agent Framework that provides a fast developer workflow with enterprise-ready LLMops & governance.
- Mosaic AI Agent Evaluation that provides reliable, quality measurement using proprietary AI-assisted LLM judges to measure quality metrics that are powered by human feedback collected through an intuitive web-based chat UI.