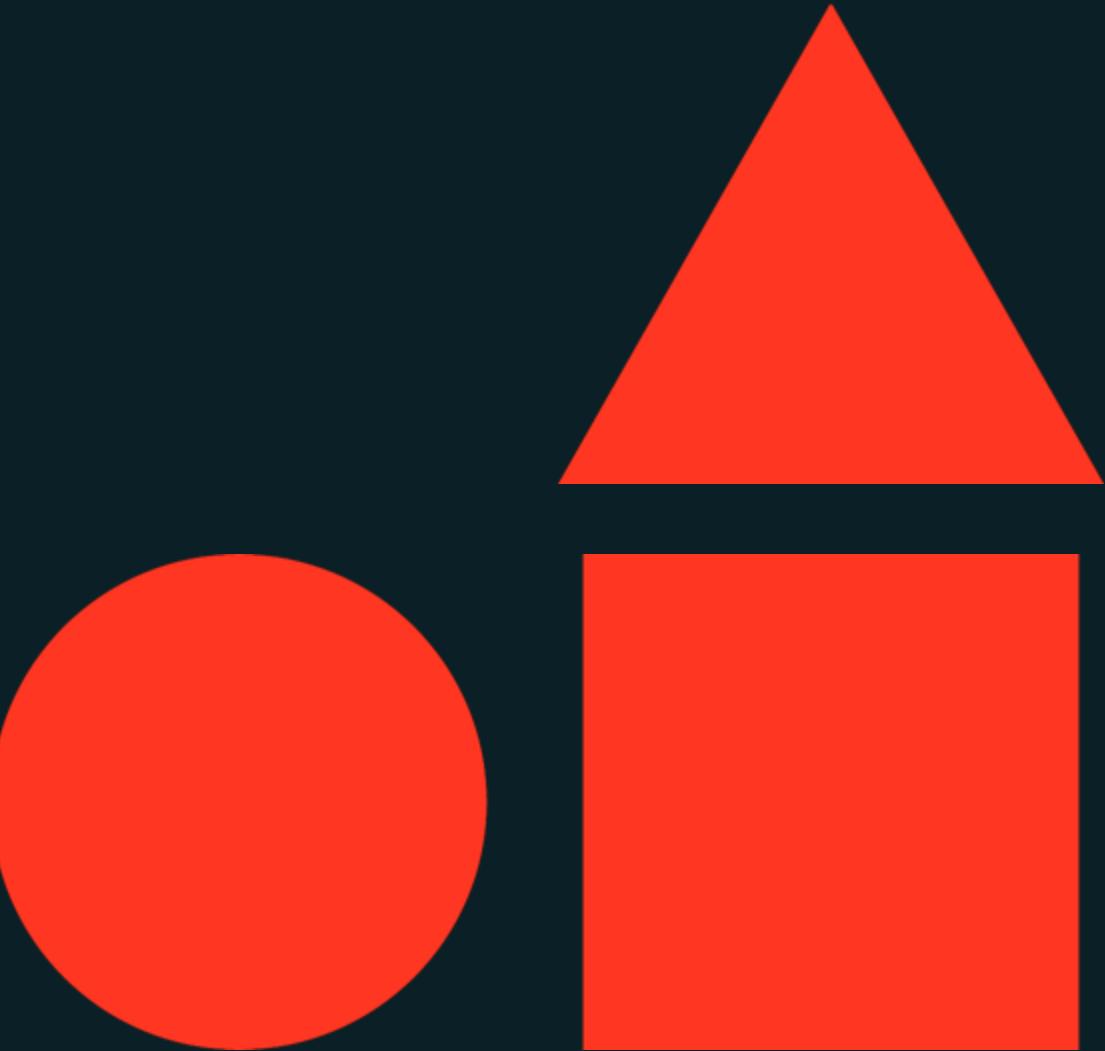




# Get Started with AI Agents on Databricks

---

Databricks Academy



# Agenda

1. Introduction to AI Agents	Lecture	Demo	Lab	Time
<b>What is An AI Agent?</b>	✓			4 mins
<b>AI Agent Patterns</b>	✓			7 mins
<b>AI Agent Components</b>	✓			10 mins
<b>Common Agentic AI Use-Cases</b>	✓			3 mins
2. Building AI Agents on Databricks	Lecture	Demo	Lab	Time
<b>The Challenges with Agents in Production</b>	✓			10 mins
<b>Overview of Mosaic AI for Agent Development</b>	✓			8 mins
<b>Custom Agent Development with Mosaic AI</b>	✓	✓ (2x)		27 mins
3. Agent Evaluation and Deployment with Mosaic AI	Lecture	Demo	Lab	Time
<b>Agent Evaluation and MLflow</b>	✓	✓	✓	33 mins
<b>Deploying Agents &amp; Model Serving</b>	✓			12 mins
4. Production-Ready Agents with Agent Bricks	Lecture	Demo	Lab	Time
<b>Introduction to Agent Bricks</b>	✓		✓	25 mins



# Course Learning Objectives

- What are Generative AI, AI Systems, Agentic AI, and AI Agents.
- Components of an AI Agent such as a LLM, tools, planning, and memory
- Develop tools and data retrievers using Unity Catalog Functions and Agent Bricks Information Extraction Agent.
- Prototype an AI Agent with AI Playgrounds using an LLM with tools use
- Deploy and Evaluate an AI Agent using Mosaic AI Agent Framework
- Productionizing Agentic AI system and AI Agents using Agent Bricks Supervisor Agent.



# Course Prerequisites (REQUIRED)



## Proficient Knowledge of the Databricks Platform

- Databricks Workspaces
- Databricks CLI
- Secrets
- Delta Lake (Nice to have)
- Git Folders (Nice to have)



## Experience of Data Analysis and Machine Learning on Databricks

- Python programming
- Foundational Knowledge of Databricks Platform, including Unity Catalog
- Familiarity with ML/AI concepts
- Familiarity with fundamental agent concepts





# Lab Exercise Environment

## Technical Details

- Your lab environment is provided by Vocareum.
- It will open in a new tab.
- It has been configured with the permissions and resources required to accomplish the tasks outlined in the lab exercise.
- Third party cookies must be enabled in your browser for Vocareum's user experience to work properly.
- Make sure to enable pop ups!



# Before we get started...

## Just a quick note

- Be aware, Databricks is a highly innovative company and through our courses we try to give you the latest and greatest.
- If you encounter inconsistencies in product naming, UI environments, or other areas of this content – let us know!

Visit [help.databricks.com](https://help.databricks.com) to submit your feedback.

Enjoy the course!



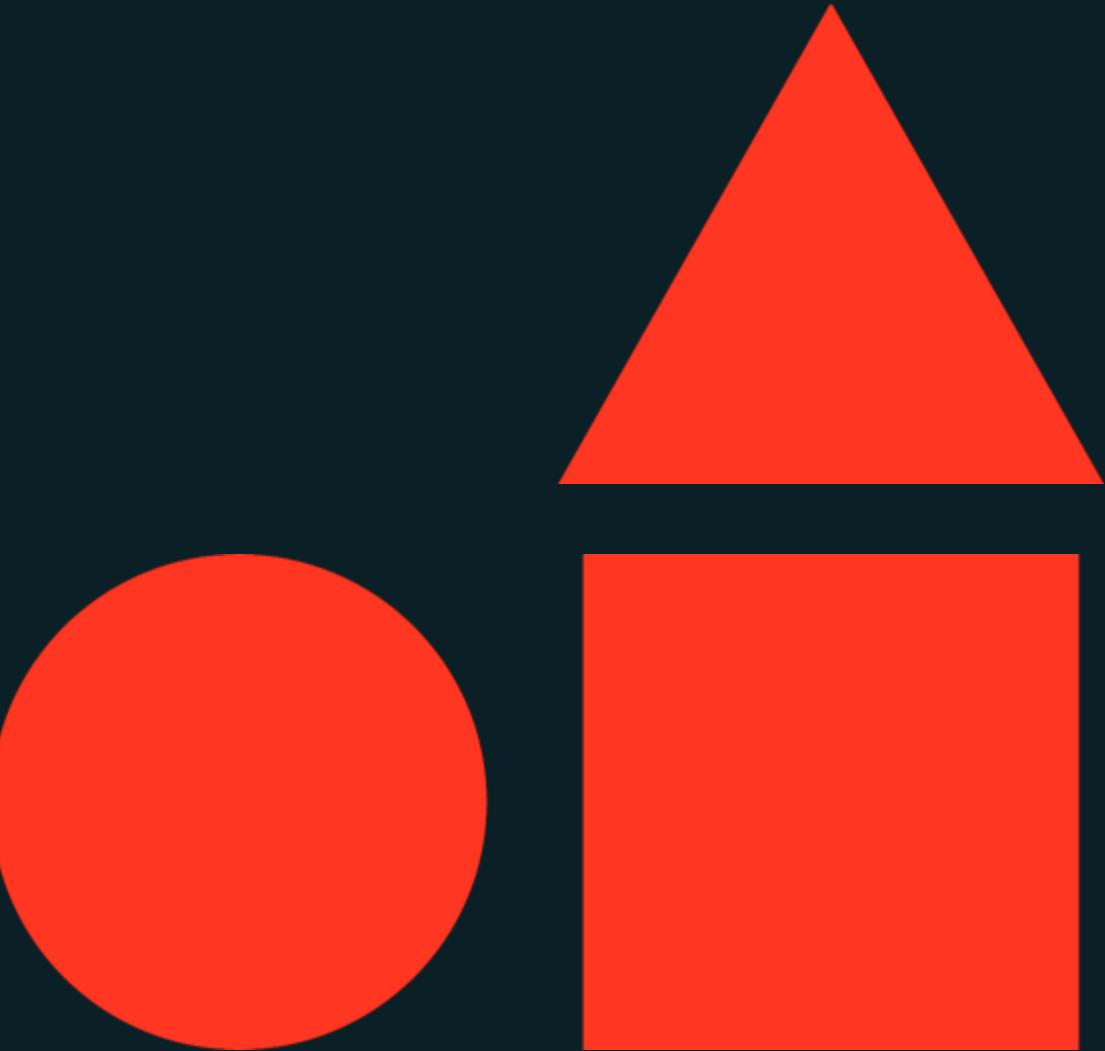


Module 1:

# Introduction to AI Agents & Agentic AI Systems

---

Get Started with AI Agents on Databricks



# Objectives

- Explain what an AI Agent is
- Understand common AI Agent patterns
- Explain various components of an Agent





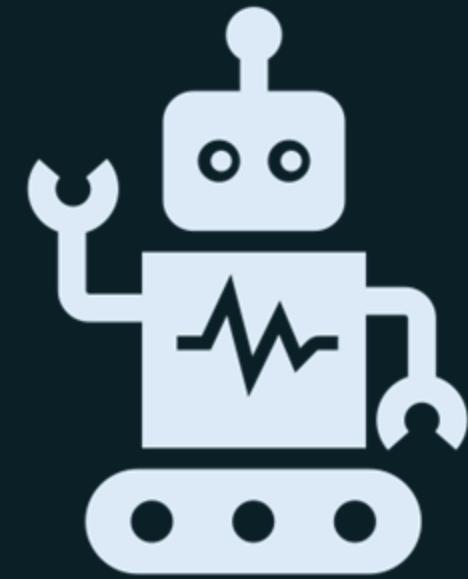
Module 1: Introduction to AI Agents

## LECTURE

# What is an AI agent?



© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).



**An AI agent is an intelligent application  
that uses an AI model and other tools  
to iteratively plan and execute  
sequences of actions to complete a  
complex task**



# What is an AI agent?

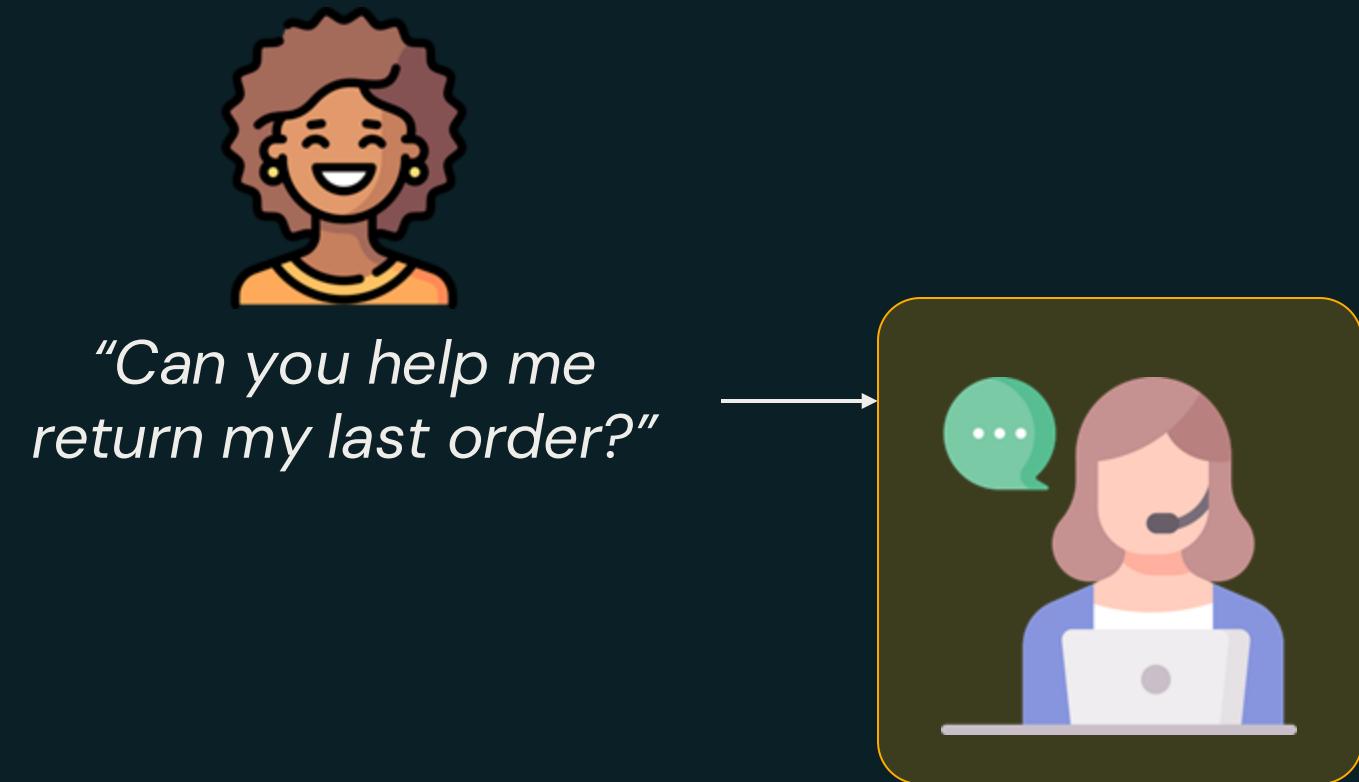


*"Can you help me  
return my last order?"*



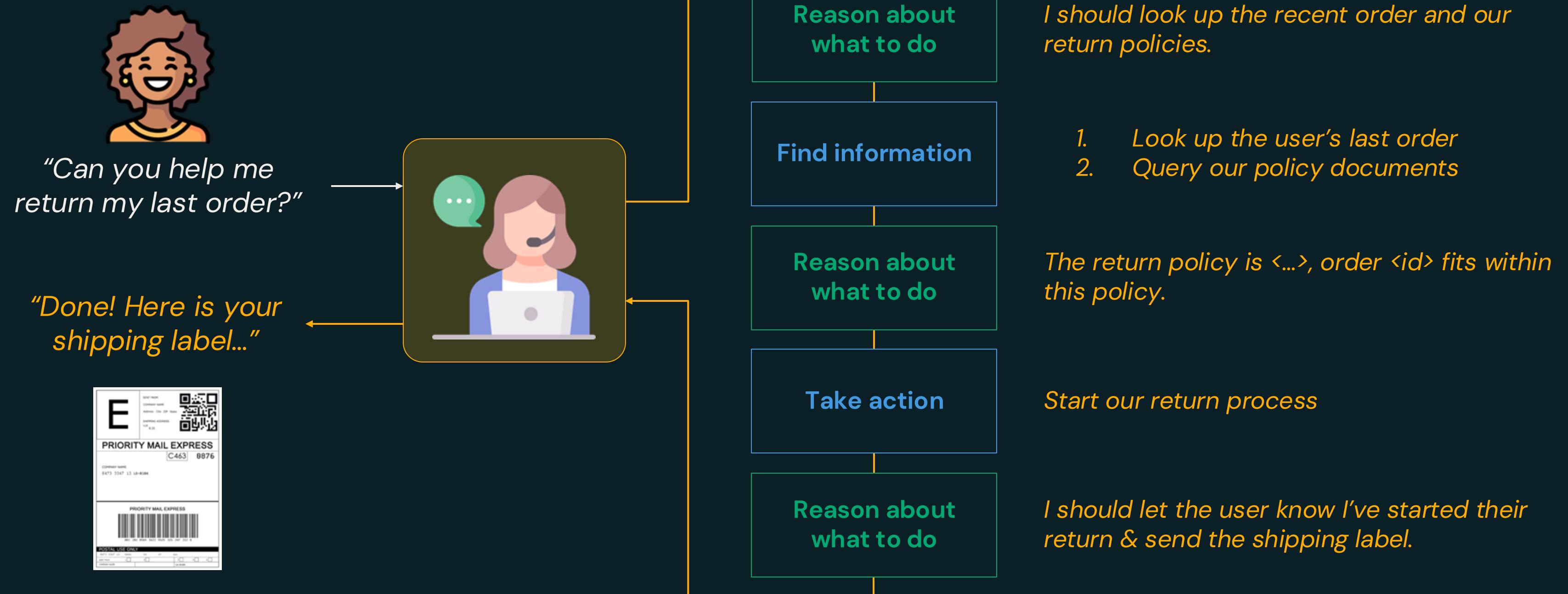
# What is an AI agent?

How would a human do this?



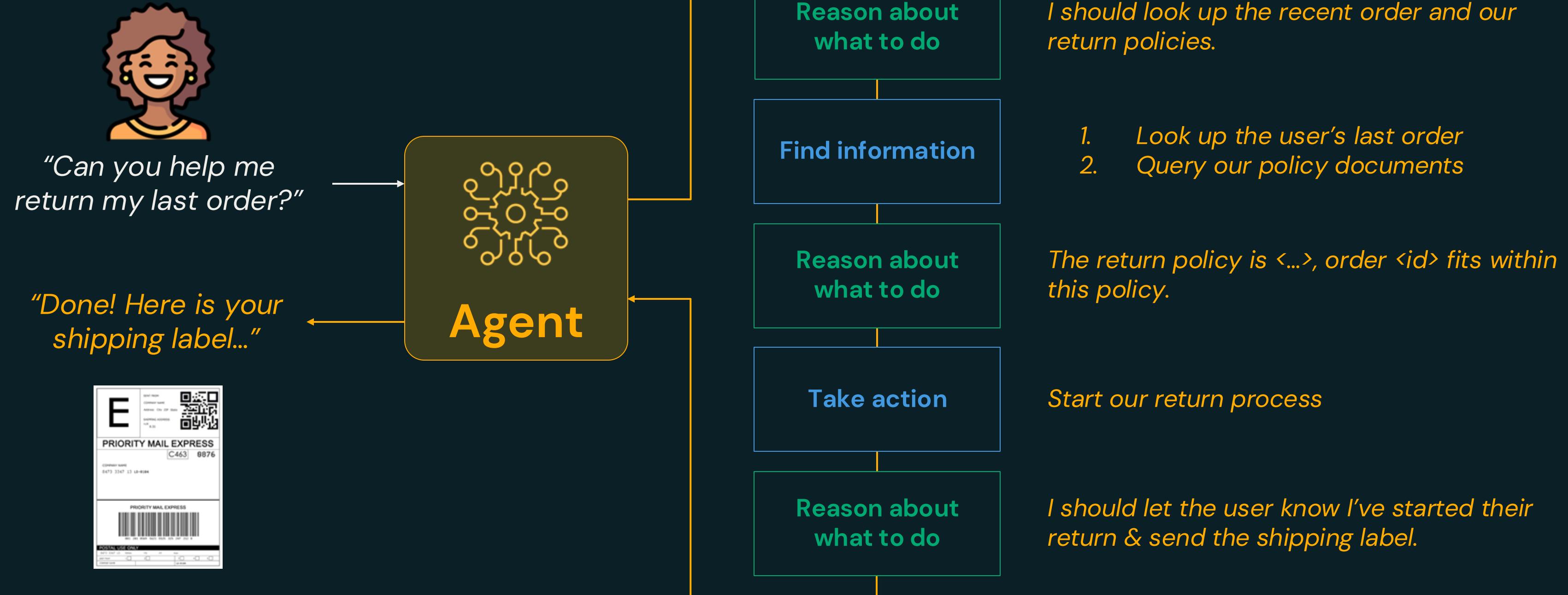
# What is an AI agent?

A human would reason about what to do, find relevant information, and take an action



# What is an AI agent?

An agent does the same thing except it uses an LLM to reason, find information, and take actions





Module 1: Introduction to AI Agents

LECTURE

# AI Agent Patterns



# AI Systems, Agentic AI, and AI Agents

Agents, bots, AI?

**AI systems** are any software system that uses AI, ML models

While an **AI Agent** is a system composed of an AI model, planning and memory, with tool use.

**Agentic AI** is any AI system that may have AI agents in it. Often this refers to multi agent strategies.



# Common Agentic AI patterns

Mosaic AI supports these and more

AI System  
with documents

*aka unstructured retrieval  
or "RAG"*

AI System  
with tables

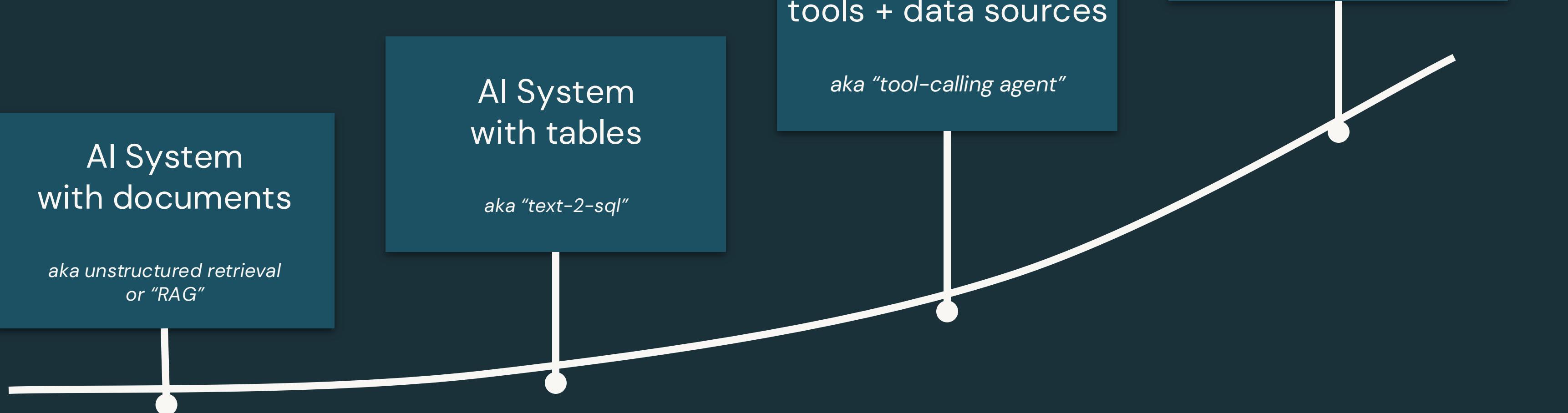
*aka "text-2-sql"*

Agent with multiple  
tools + data sources

*aka "tool-calling agent"*

Agent that  
orchestrates  
other agents

*aka "multi-agent"*



Interactivity, Capability and Scope Increases

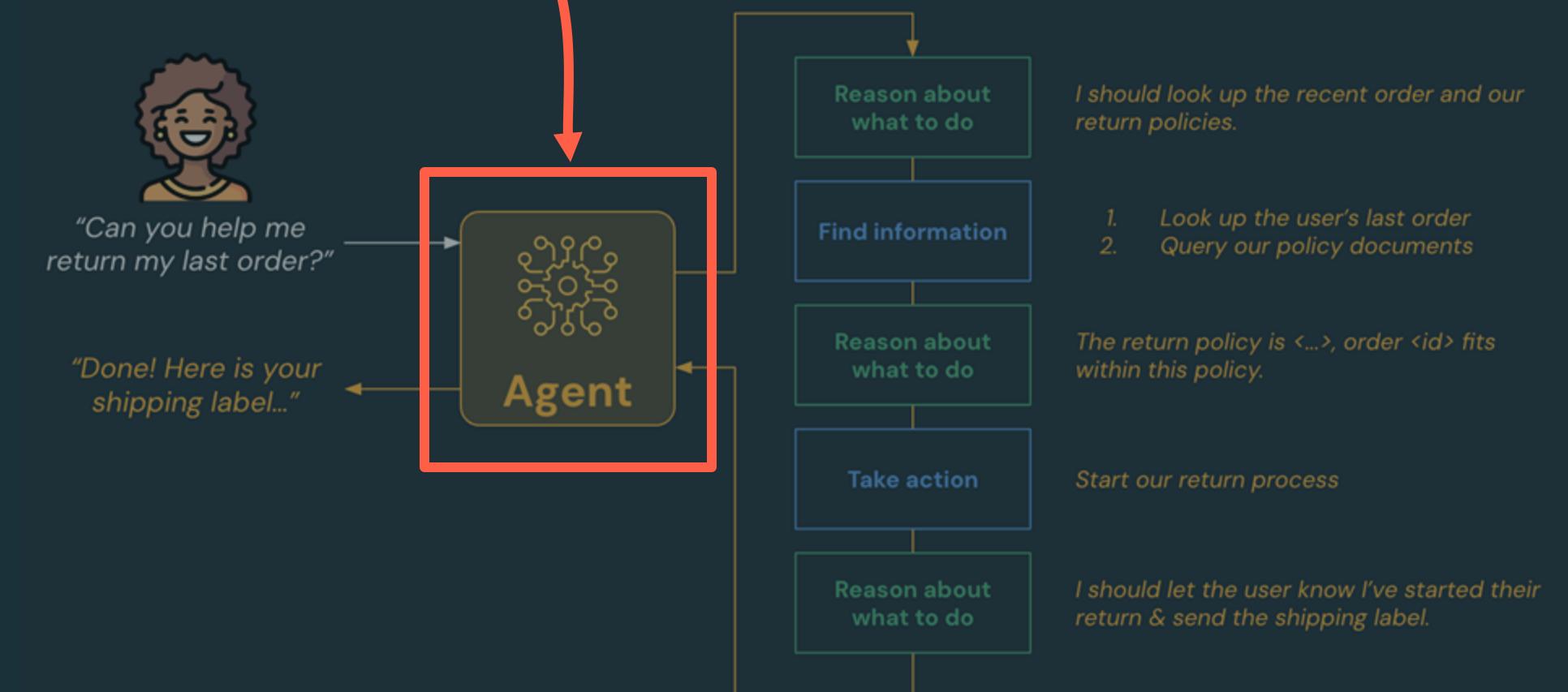
Security, Reliability and Robustness Decreases



# Definition: Conversational Agent

TLDR: Any multi-turn chatbot, no matter how it is implemented

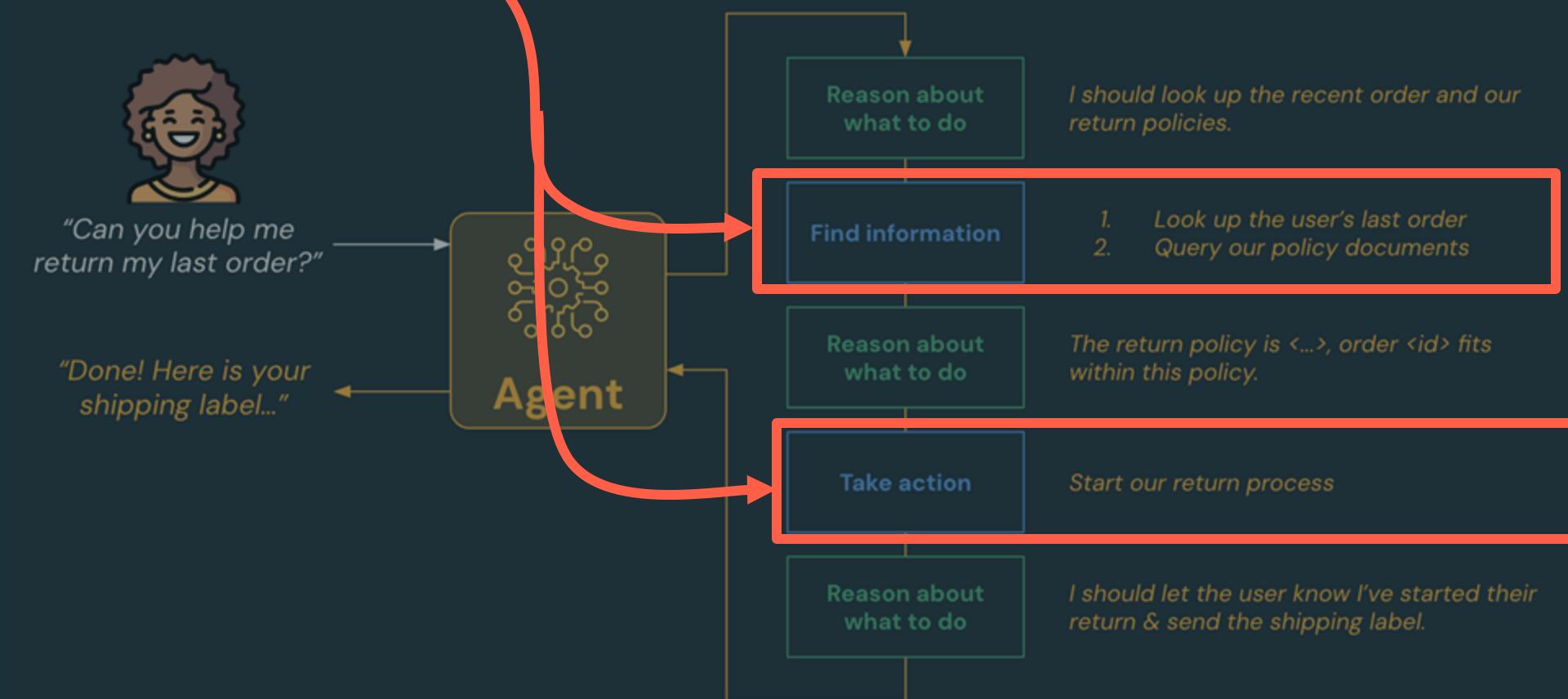
**Technically:** A conversational agent (chat agent, chatbot) designed for iterative collaboration with humans and other agents through multi-turn conversations, where the conversation history is maintained somewhere.



# Definition: Tool

TLDR: A tool is a function designed to complete a specific task in one step.

**Technically:** A tool is a specific function that an AI model uses to perform a task. The details needed for the function (called parameters) are selected by the AI itself. A tool is designed to complete a specific task in a **single step**.



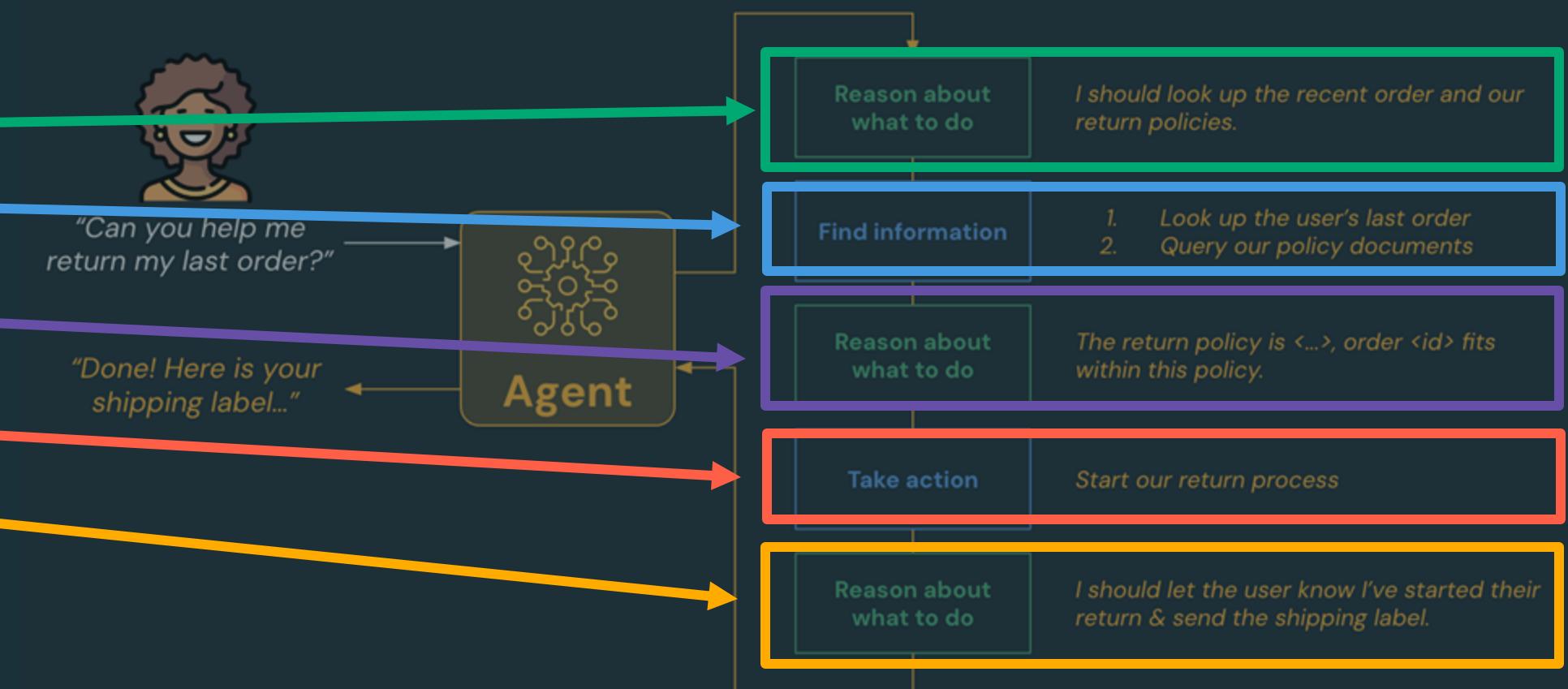
# Definition: Tool-Calling Agent

**TLDR:** An agent that uses an LLM to reason about & respond to user queries iteratively using tools.

**Technically:** an Agent that has a set of Tools. The Agent uses an LLM to:

1. Reasons about which Tool(s) to call
2. Calls those tools
3. Reasons about tool output and either:
  - a. Call additional Tool(s)
  - b. Respond to the user

This loop repeats until the Agent decides to respond back to the user.

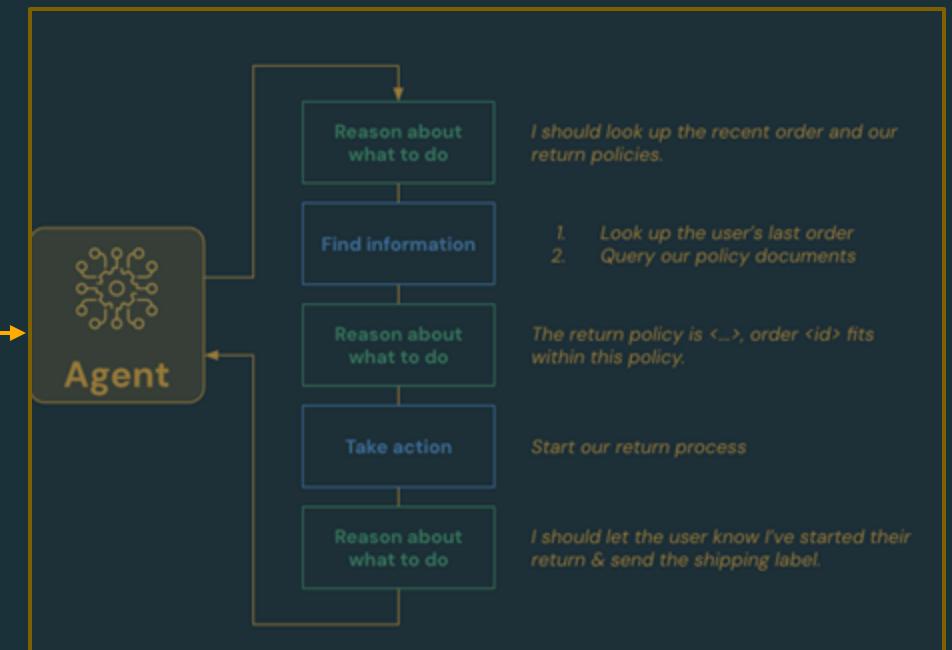
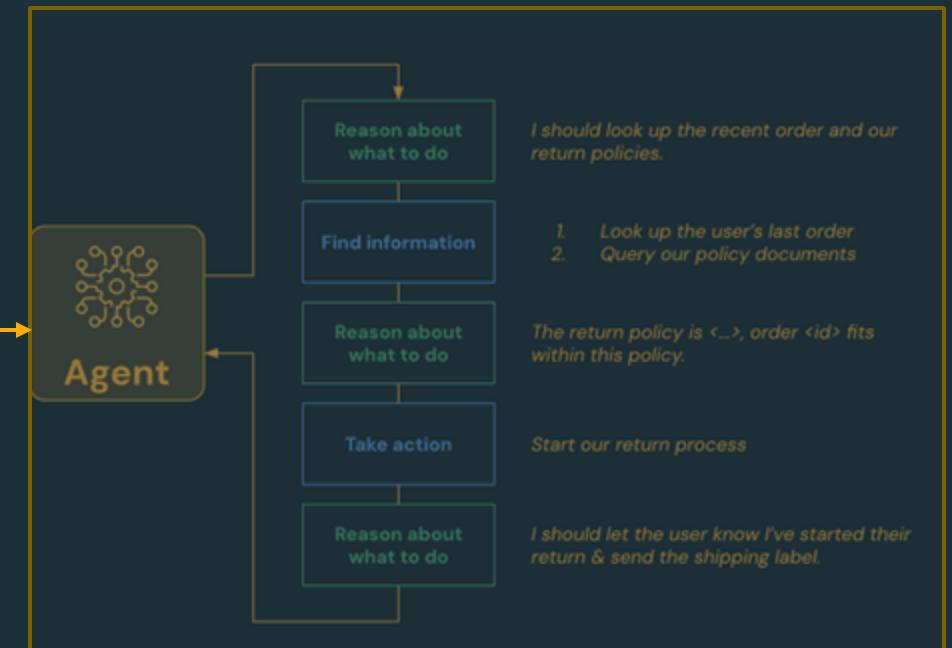
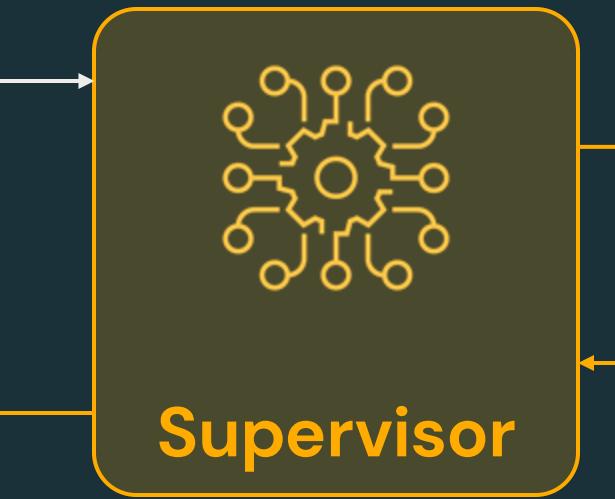


*-- Function-calling Agent*



*"Can you help me return my last order?"*

*"Done! Here is your shipping label..."*



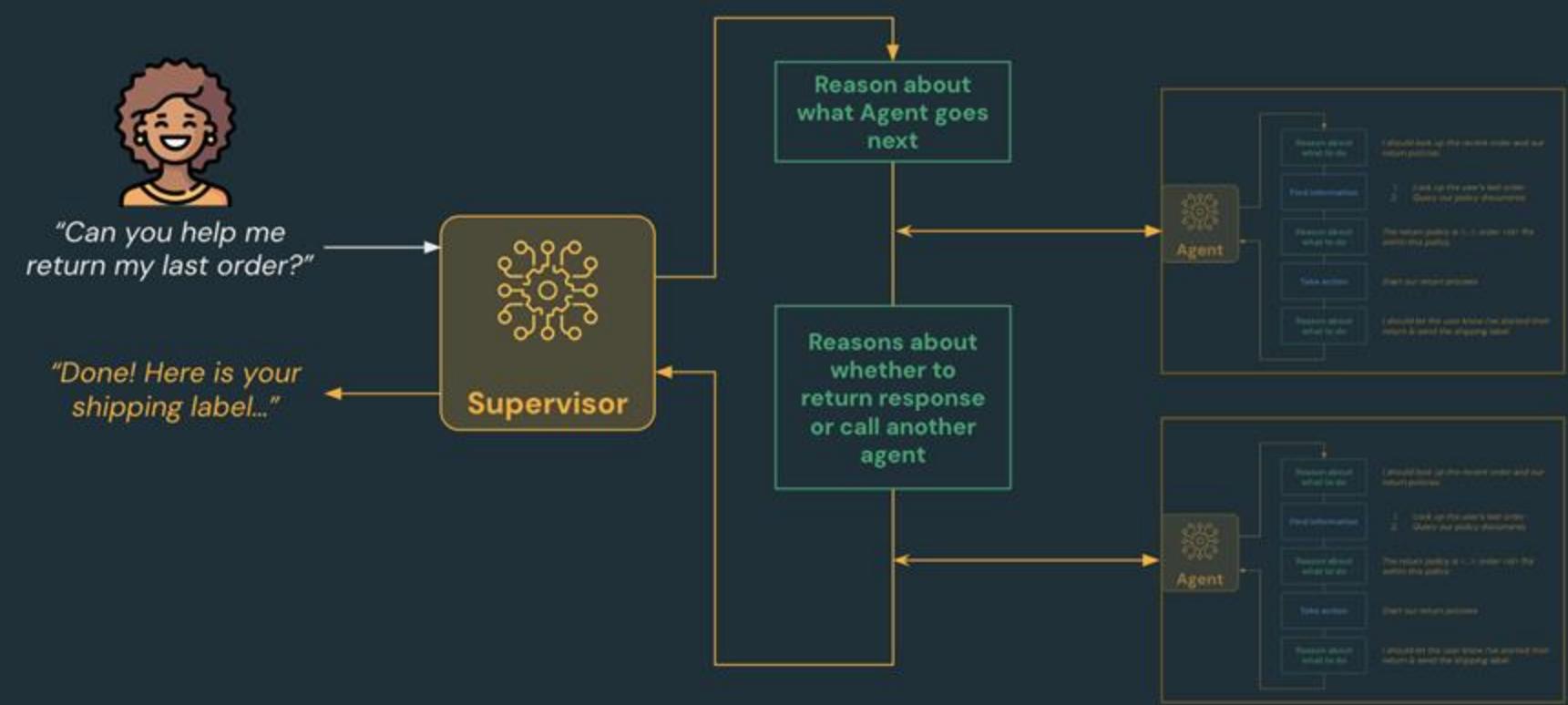
# Definition: Multi-Agent

**TLDR:** A supervisor agent coordinates multiple agents, directing conversation flow to the each agent..

**Technically:** A type of Agent, called a supervisor (moderator/router) that combines multiple Agents together.

The supervisor:

1. Uses an LLM to decide which agent should speak next
2. Deterministically passes the conversation history to that agent
  - a. e.g., via code, no LLM regurgitating the tokens
3. Lets that agent respond
4. Based on that response + conversation history, decides to either
  - a. repeat (1)
  - b. return the response from (3) to the user





Module 1: Introduction to AI Agents

LECTURE

# AI Agent Components



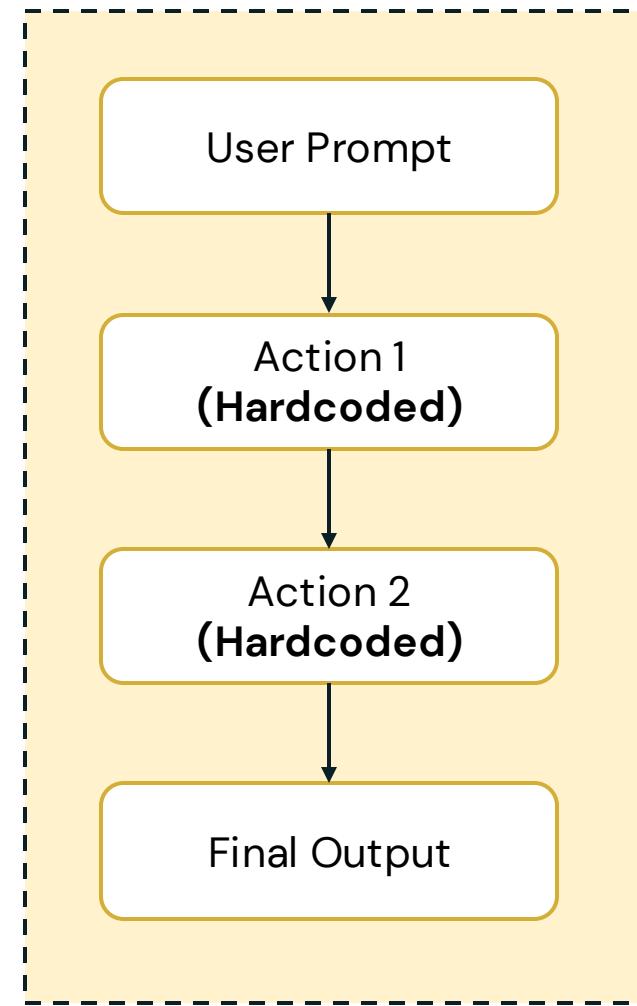
© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).

# AI Agent Systems

## Non-agentic vs. agentic workflows

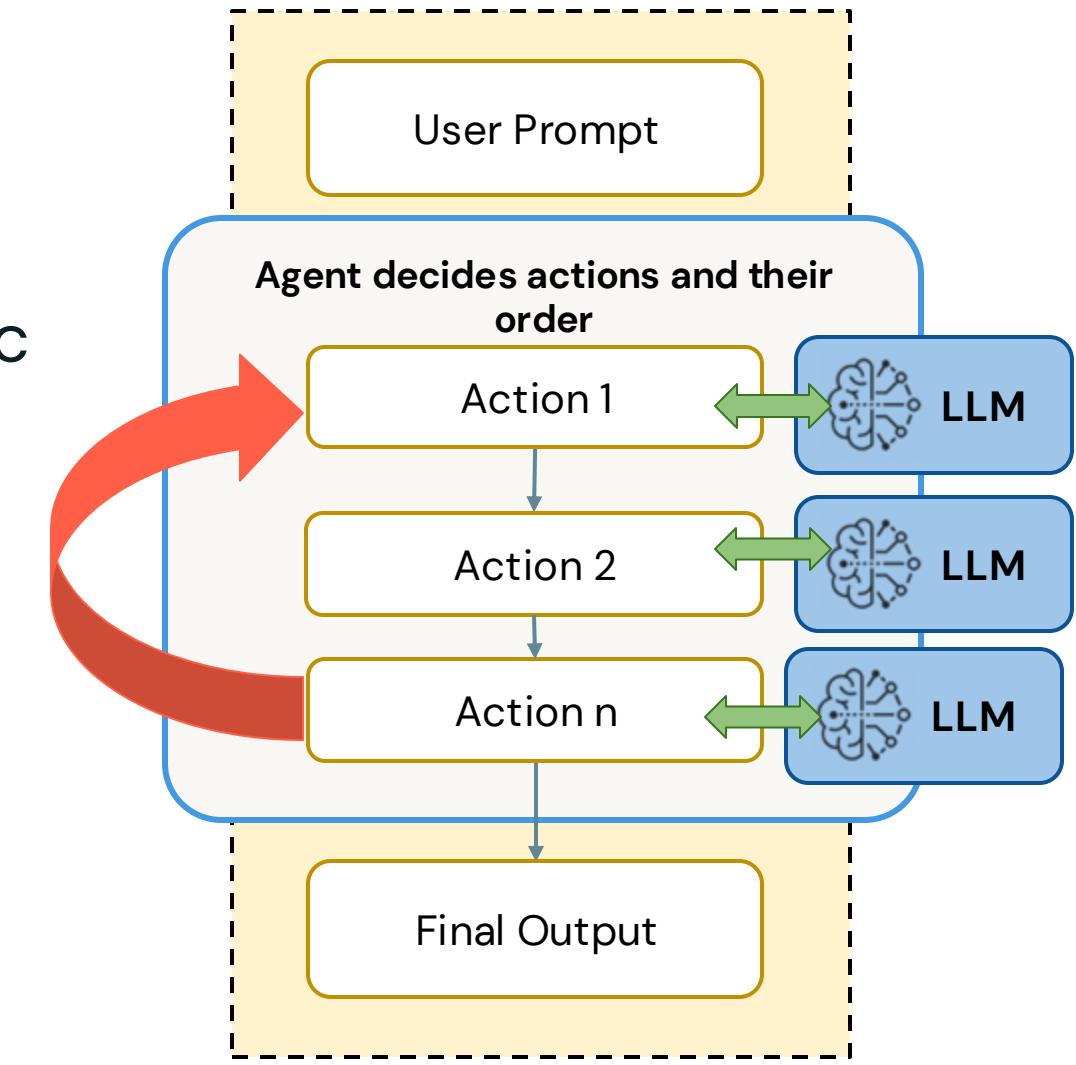
### Non-agentic (static) workflow

- Static prompt response or fixed pipelines of hardcoded prompt response systems
- Could also be an API call to another service or ML model.
- Deterministic actions



### Agentic (dynamic, iterative) workflow

- Planning and execution by AI
- Tool calling by AI
- Non-deterministic actions
- Iterative workflows



# AI Agent Systems

## Core components

### LLM

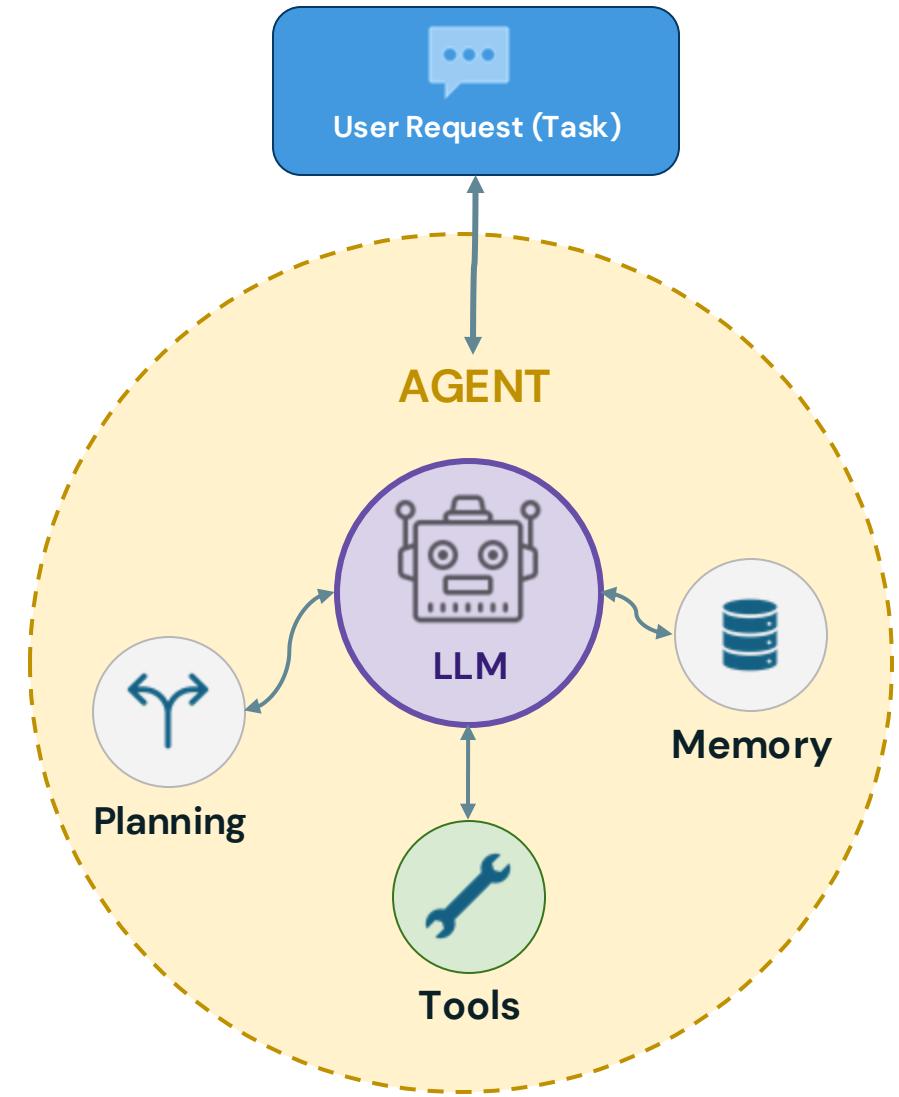
- A “**brain**” LLM to control the core logic and sequencing of actions the agent executes. It plans the work of the interacting components.
- Other LLMs or AI models as needed for sub-tasks and actions.

### Tools

External resources that the agent uses via tool use or tool calling, e.g., functions, APIs, classic ML models.

### Memory

- Short-term session and current conversational state tracking to help with planning and execution of subsequent actions.
- Long-term episodic, semantic, and procedural memory for historical state, knowledge, preferences.



# Pattern: Reasoning

## Thinking and thoughts

### Chain-of-Thought

A **prompting** technique that can be used to illicit reasoning from a instruct model.

### Reasoning model

A model that has been **trained** to automatically determine when to think about a problem.

The screenshot shows the Databricks Playground interface with three separate model instances. Each instance has a header bar with the Databricks logo, search bar, and account information. The main area displays a conversation between 'You' and the model.

- You:** How many R's are in the word Strawberry?
- Llama 4 Maverick:** There are 2 R's in the word Strawberry.  
0.20s to first token + 0.21s + 58 tokens/sec = 12 tokens
- AI Judge:** Suggested questions Preview
- Claude Sonnet 4:** To count the R's in "Strawberry", I'll go through each letter:  
S-T  
/ Thinking  
1.08s to first token + 0.16s

At the bottom of the interface, there are buttons for 'Add system prompt', 'Start typing ...', 'Sync', and a 'Stop generating' button which is highlighted in black. A note at the bottom states: "Models called in Playground may be subject to license requirements and/or use policies. [Learn more](#)".



# Pattern: ReAct (Reason + Act)

## Agent reasoning patterns

- Enables models to generate **verbal reasoning** traces and **actions**.
- Main states used in a ReAct agent are;
  - **Thought:** Reflect on the problem given and previous actions taken
  - **Act:** Choose the correct tool and input format to use.
  - **Observe:** Evaluate the result of the action and generate next thought.

Source: Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2022). ReAct: Synergizing Reasoning and Acting in Language Models. ArXiv. /abs/2210.03629

**Question:** Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

**Answer:** iPod Standard ✗

**Thought:** Let's think step by step. Apple Remote was originally designed to interact with Apple TV. Apple TV can be controlled by iPhone, iPad, and iPod Touch. So the answer is iPhone, iPad, and iPod Touch.

**Answer:** iPhone, iPad, iPod Touch ✗

**Act only**  
Act 1: `Search[Apple Remote]`  
Obs 1: The Apple Remote is a remote control introduced...  
Act 2: `Search[Front Row]`  
Obs 2: Could not find [Front Row]. Similar: ...  
Act 3: `Search[Front Row (software)]`  
Obs 3: Front Row is a discontinued media center software ...  
Act 4: `Finish[yes]` ✗

**ReAct**  
**Thought 1:** I need to search Apple Remote and find the program it was originally designed to interact with.  
Act 1: `Search[Apple Remote]`  
Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the Front Row media center program ...

**Thought 2:** Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.  
Act 2: `Search[Front Row]`  
Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

**Thought 3:** Front Row is not found. I need to search Front Row (software).  
Act 3: `Search[Front Row (software)]`  
Obs 3: Front Row is a discontinued media center software ...  
**Thought 4:** Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.  
Act 4: `Finish[keyboard function keys]` ✓

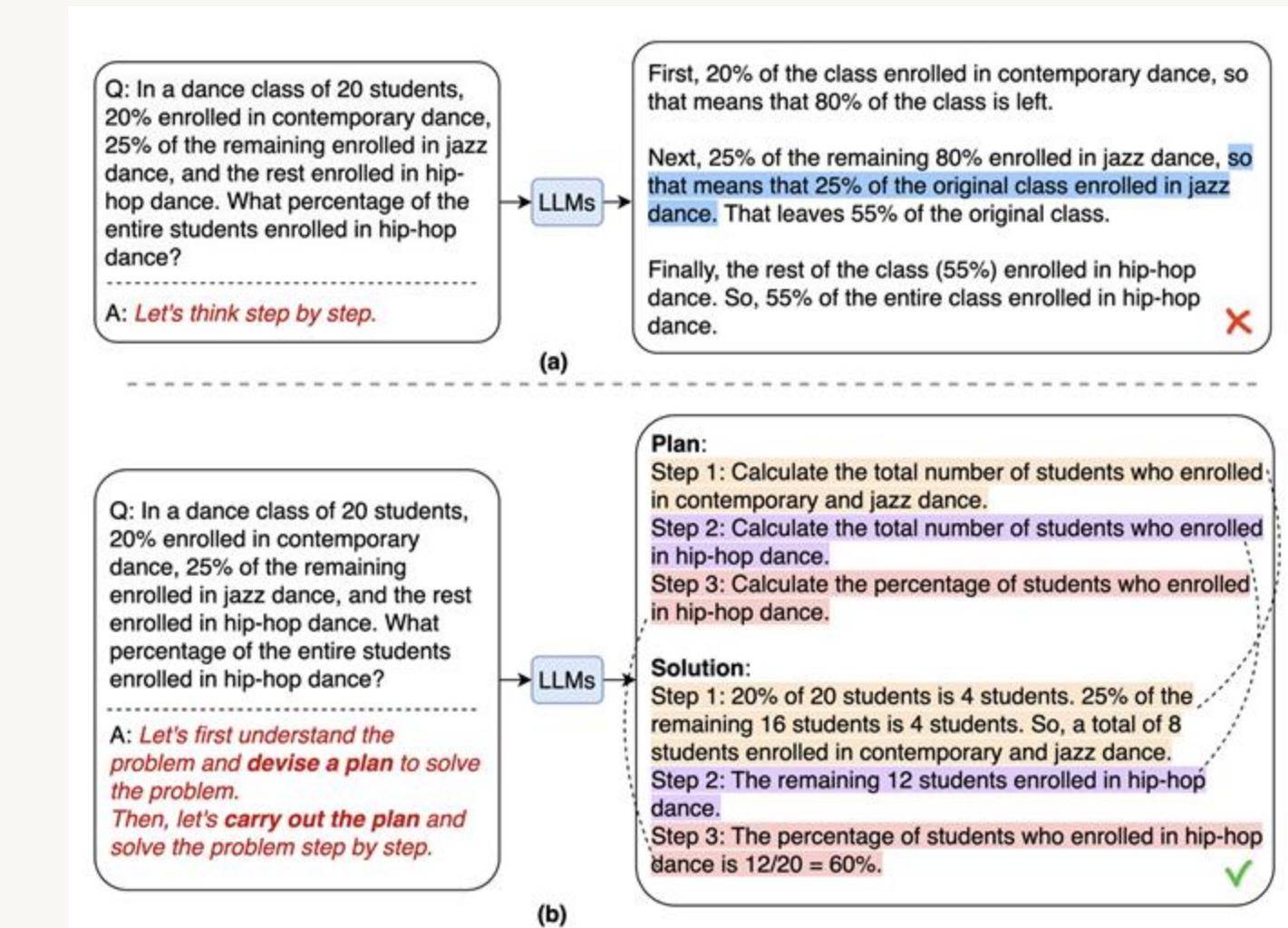


# Pattern: Plan-And-Solve

## Agent reasoning patterns

Source: Lei W., Wanyu X., Yihuai L., Zhiqiang H., Yunshi L., Roy L., Ee-Peng L. (2023). Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. ArXiv. /abs/2305.04091

- Enables models to generate **plans** and then **execute** and **observe**.
- Main components are
  - **Planner:** Reflect on the task and generate a series of steps to complete the task
  - **Execute:** Process step(s) in the generated plan using tools and observations

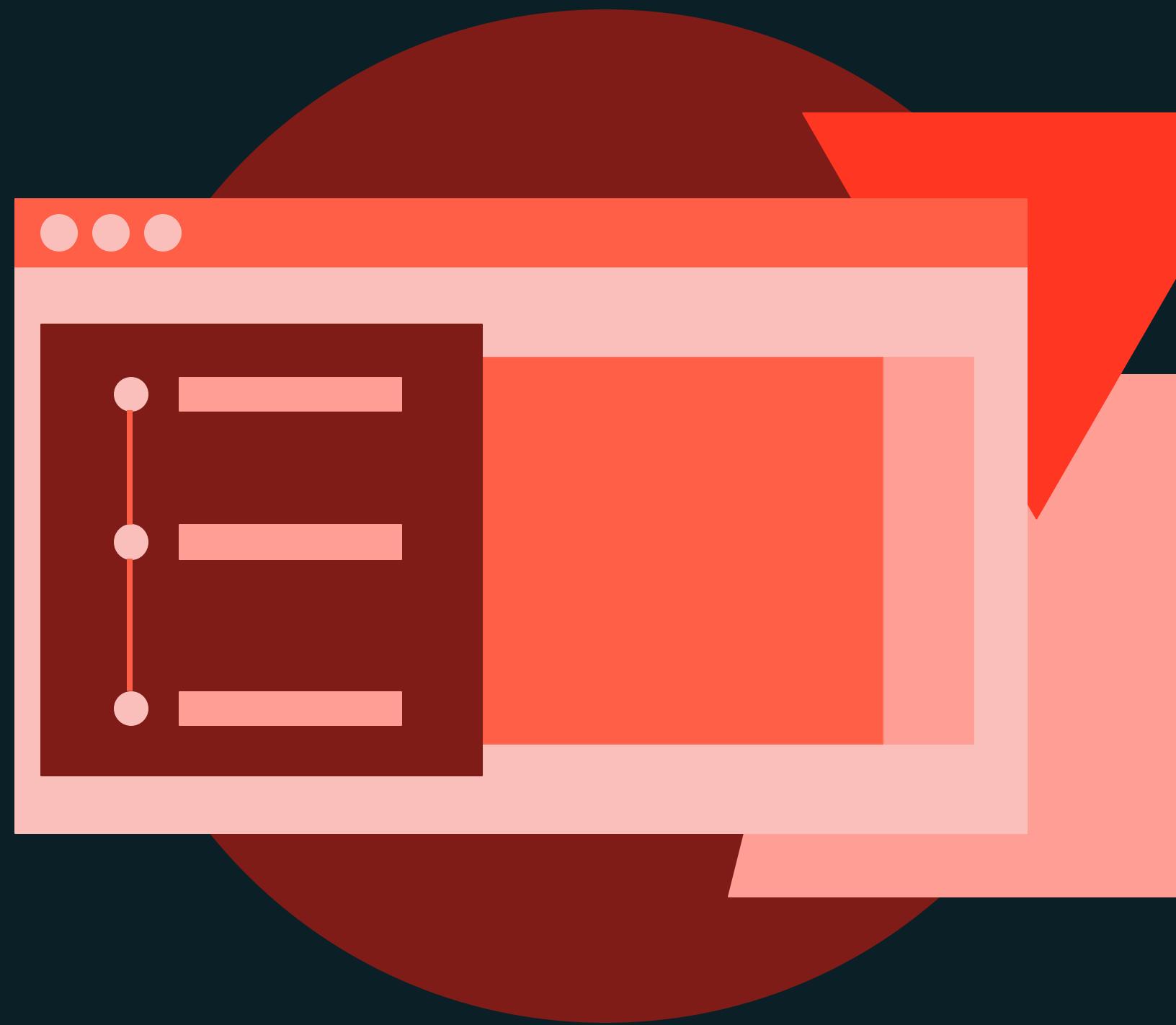




Module 4: Building Agents with Databricks

## DEMONSTRATION

# Dataset & LLM Reasoning with AI Playground



© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).



Module 1: Introduction to AI Agents

LECTURE

# Common Agentic AI Use-Cases

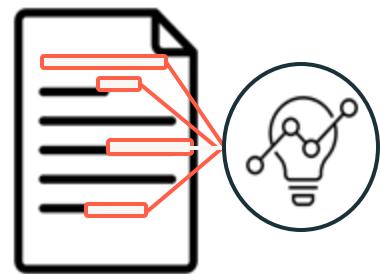


© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).

# Agentic AI Use Cases

## Key themes

### Intelligent Document Processing



Agentic AI system that extract insights from documents at scale

### Knowledge Base + Search



AI agents for search and retrieval

### Machine Learning + AI



AI system that combine classic machine learning and Generative AI



# Databricks

Some ways we use agents

## Databricks Assistant

The screenshot shows a Databricks Notebook titled "Notebook example v2" in Python. The notebook contains code for a machine learning model. On the left, there's a sidebar for the "Databricks Assistant". A red arrow points to the "Databricks Assistant" icon. The sidebar includes sections for "Requirements" (listing "Databricks Runtime for Machine Learning") and "Load and preprocess data" (mentioning the California Housing dataset). It also has a "New Assistant: ON" toggle and a "Send feedback" button.

## Customer Support

The screenshot shows a "Contact support" dialog box. It has fields for "Priority", "Issue", "Phase", "Product area", "Component", "Databricks runtime version", "Notebook url", and "Cluster id". There's a text area for "Describe your issue..." and an "Image" section with a "Capture a screenshot" checkbox. At the bottom, there are checkboxes for "I want to use Spark support hours.", "By clicking "Send message" you acknowledge that Databricks personnel may access your workspace(s) unless you have CAWL.", and "Share chat history". Buttons for "Cancel" and "Send message" are at the bottom right.

## AI/BI Genie

The screenshot shows the AI/BI Genie interface with a "Sales & Marketing Performance" dashboard. It displays various metrics like "Top brands by territory", "Customer churn by region", "Margin by business unit", "Top level business performance", "Category performance current FY", and "Explain the data set". There's a search bar at the top and a "Send feedback" button.



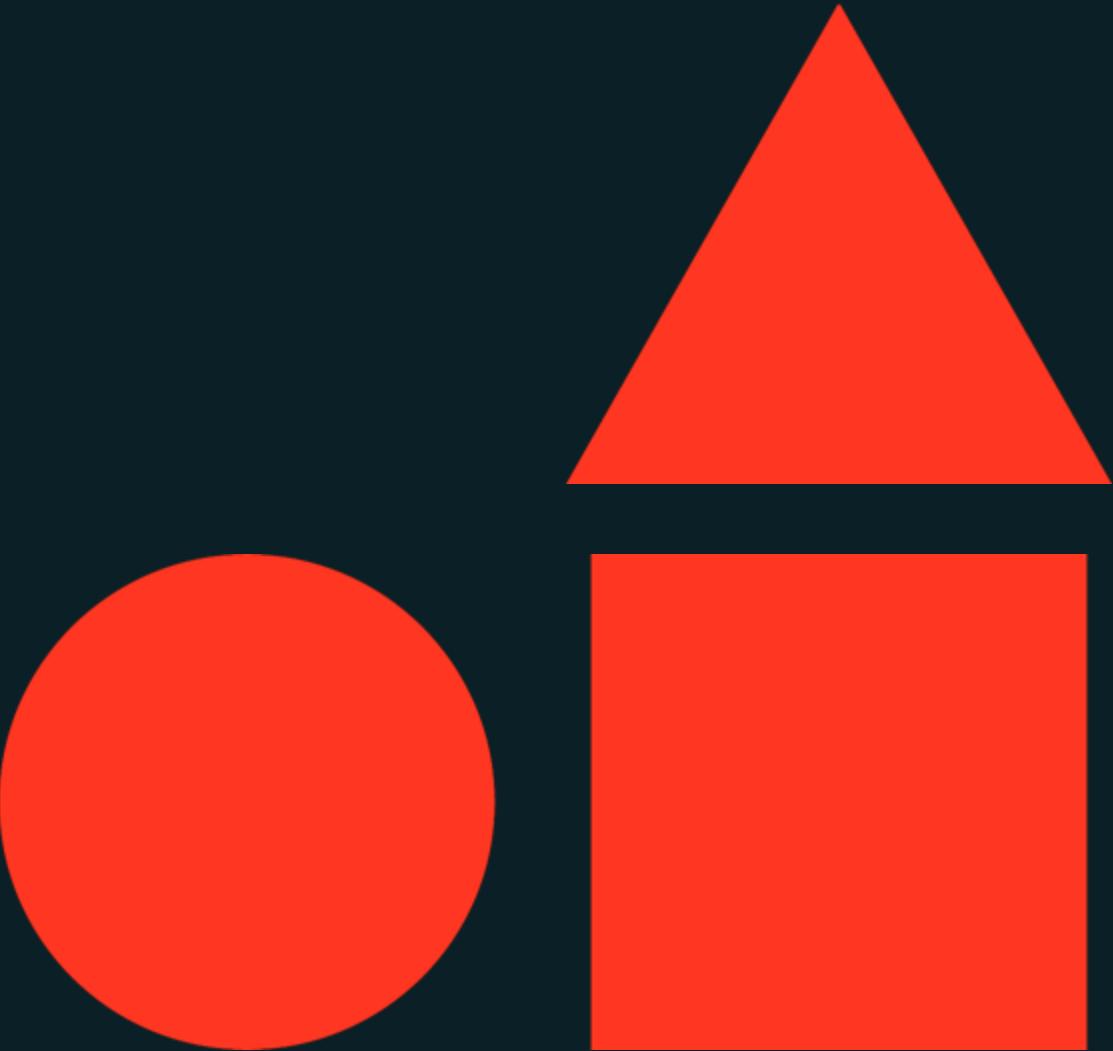


Module 2:

# Building AI Agents on Databricks

---

Get Started with AI Agents on Databricks



© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).

# Objectives

- Explain common challenges with productionizing AI Agents
- Understand how Databricks and Mosaic AI enable an AI Agent's lifecycle
- Explain the full Agent Development Lifecycle
- Determine how to choose agent components





Module 2: Building AI Agents on Databricks

LECTURE

# The Challenges with Agents in Production



© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).

# The Path to Production is Hard

LLM Ops: Control, Quality, Cost – you need more than just good models



Difficulty **controlling**  
data or models



Production-level  
**quality** is challenging



High **cost** at scale



Concern over data leakage

Determining control and ownership



Unpredictable performance

Need automation and scale



Foundation models are expensive at scale

Time consuming to build Agents

# Challenge:

## Building and deploying production-quality Gen AI solutions

90%  
of enterprises not  
confident going  
to production

# General Intelligence fails for enterprise use cases

## General Intelligence

Foundation models trained  
on a broad dataset  
**disconnected** from your  
business data

## Data Intelligence

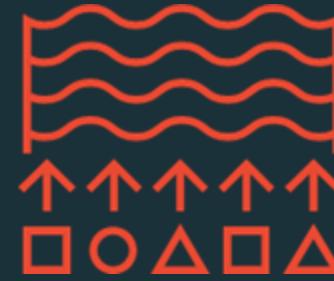
AI **connected** to your  
customer data and able  
to solve domain-  
specific problems

vs



# Mosaic AI enables you to build production-quality, enterprise-ready Agents faster

## End to end governance



Agents and tools that respect existing data and AI governance



*Reduce potential privacy & reputational risk*

## Production quality

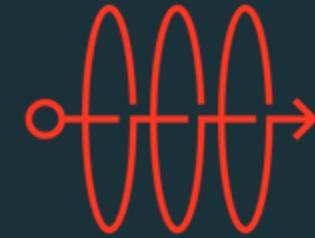


Deliver accurate agents that are evaluated and monitored



*Provides native evaluation and monitoring, grounded in our bleeding edge research*

## Rapid Development



Rapidly iterate and redeploy to improve quality



*Shave months off your time-to-market with our agent tooling*

# Databricks approach to quality

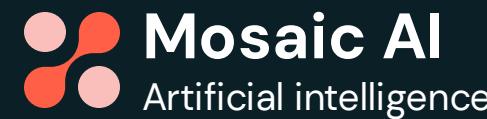
*Accurate quality measurement*

*Fast developer loop*

*Easy to collaborate with SMEs*



# Data Intelligence Platform



**Mosaic AI**  
Artificial intelligence



**DB SQL**  
Data warehousing



**Lakebase**  
Transactional database



**AI/BI**  
Business intelligence



**Lakeflow**  
Ingest, ETL, streaming



**Apps**  
Secure data & AI apps



**Marketplace**  
Data & AI marketplace



**DELTA LAKE**

**ICEBERG**



Module 2: Building AI Agents on Databricks

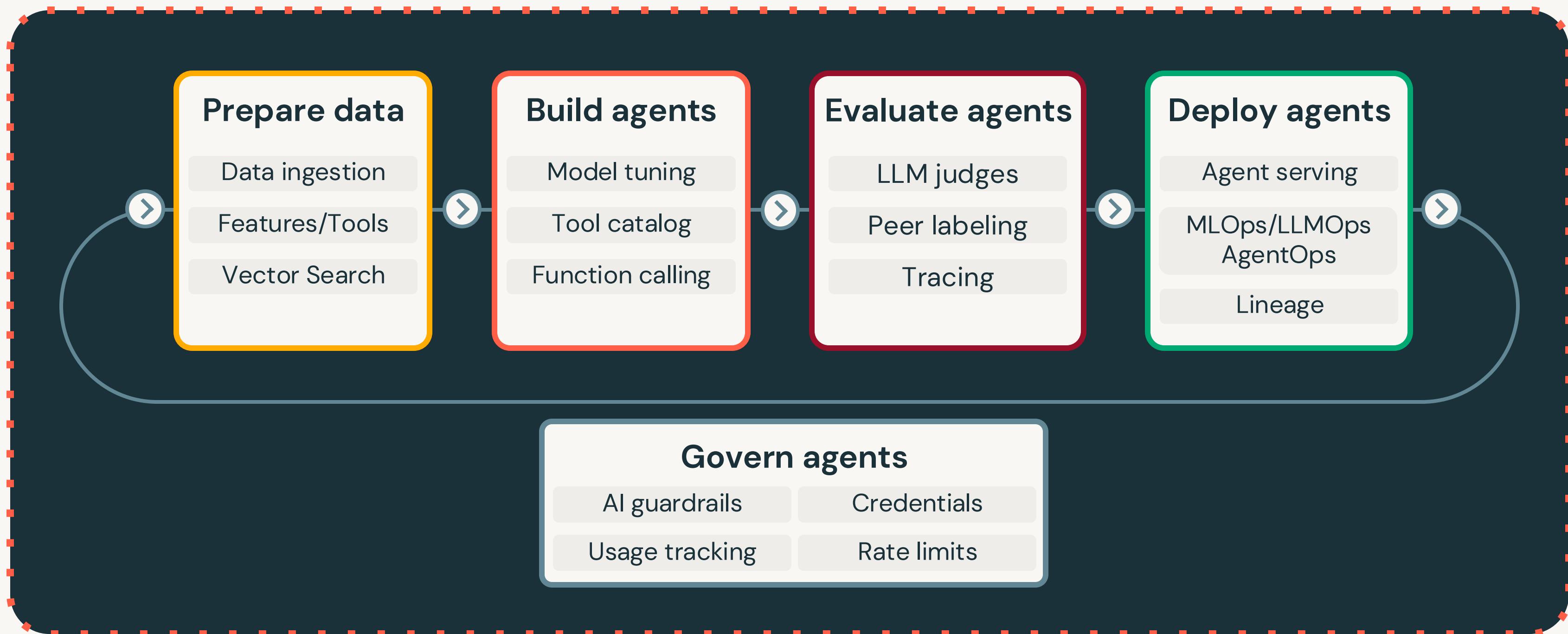
LECTURE

# Overview of Mosaic AI for Agent Development

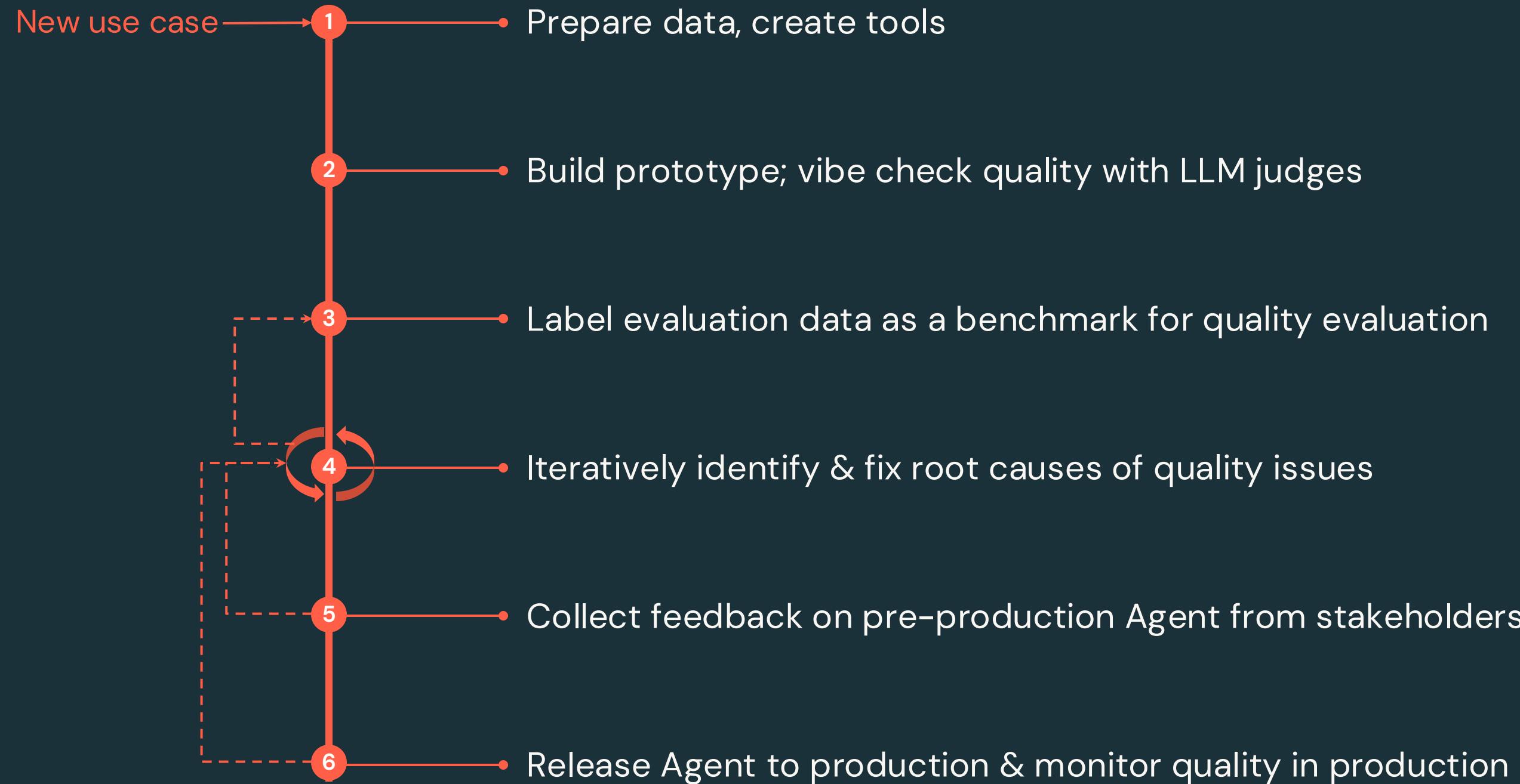


© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).

# Agent Development Lifecycle



# Let's walk through the Agent lifecycle to see how Mosaic AI helps you build high-quality Agents



# Mosaic AI Overview

Platform capability (Data + AI)

Mosaic AI capability (AI)

Asset Bundles (DABs)  
CI/CD support

MLOps Stacks

## MLOps + LLMOps

MLflow

### Data & Feature Pipelines

Lakeflow

Online Features

Vector Search

Functions and Tools  
(Including MCP)

### Building AI Systems

Agent Bricks

AI Playground

Foundation Model APIs

AutoML

### Evaluating AI Systems

MLflow 3.0+

Mosaic AI Agent Framework and Evaluation

### Deployment and Integration

Model Serving

Databricks Apps

Batch Inference and AI\_Query()

### Observability and Monitoring

Lakehouse Monitoring

AI Gateway (Usage Tracking)

Unity Catalog

## Governance

AI Gateway



# Mosaic AI offers a paved path to improve quality and deliver LLMOps with enterprise governance

Agent Evaluation & Framework complement and integrate with Unity Catalog and the rest of the Mosaic AI suite.

Each component is modular and can be used by itself as desired.

## Fast developer loop

- Rapid prototyping via AI Playground
- Iterate on Agent & Tool code in Notebooks
- Track every version of your agent's code/config with MLflow logging
- 1-line of code to deploy a scalable, production ready endpoint

## Accurate, fast quality measurement

- Synthetic evaluation dataset generation
- Proprietary, research backed LLM judges for accurate quality measurement & root cause analysis
- Online quality & usage monitoring

## Easy to collaborate with SMEs

- Stakeholder labeling UI to collect high-quality data, efficiently
- Built-in chat app to let stakeholders test your Agent

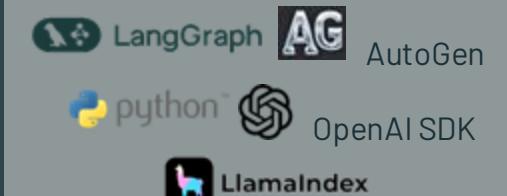
## LLMOPs for Agents

- MLflow Tracing: telemetry & observability
- Tools as Unity Catalog Functions
- Model Serving: high QPS agent hosting
- AI Gateway: Inference Tables log requests, traces, and user feedback
- Lakeguard: Sandboxed tool code execution

Tightly integrated with Unity Catalog for governance

Unity Catalog

Agent code



Tool code

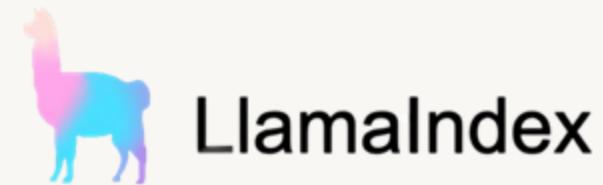


Your Data

# GenAI Frameworks & Databricks

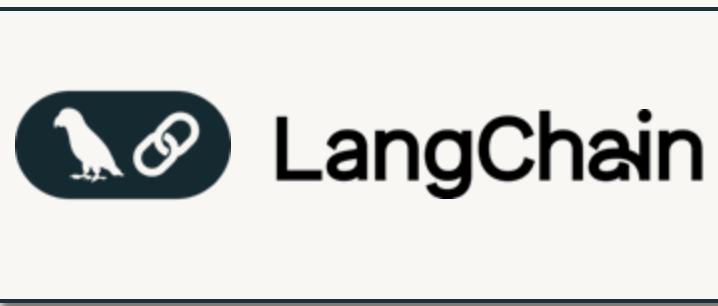
Databricks is **Framework Agnostic**

You can build your LLM solution using any framework and use all of the enterprise AI and governance features on the platform



# Top Frameworks

## Community Statistics



**LangChain**



**LangGraph**



**LlamaIndex**



**DSPy**

First PyPI Release: October 2022

PyPI Downloads per month: 78M

Stars on GitHub: 117K

- Large ecosystem of integrations and ready-made components.
- Quickly prototype with RAG and external tools.
- Imperative, step-by-step pipeline approach to LLM orchestration.



# Top Frameworks

## Community Statistics



LangChain



LangGraph



LlamaIndex



DSPy

First PyPI Release: January 2024

PyPI Downloads per month: 10M

Stars on GitHub: 19.4K

- Persistent state across interactions or sessions.
- Visualizable, graph-like structure for debugging for design.
- Complex flows instead of linear chains in LangChain.



# Top Frameworks

## Community Statistics



LangChain



LangGraph



LlamaIndex



DSPy

First PyPI Release: February 2023

PyPI Downloads per month: 5.3M

Stars on GitHub: 44.6K

- Connecting an LLM to custom/private data.
- Data ingestion pipelines and indexing strategies.
- Focused on modular retrieval and query engines vs. general orchestration.



# Top Frameworks

## Community Statistics



LangChain



LangGraph



LlamaIndex



DSPy

First PyPI Release: April 2024

PyPI Downloads per month: 2.2M

Stars on GitHub: 28.9K

- Systematic prompt engineering and reproducibility.
- Learn and refine prompts automatically based on your goals.
- Focused on prompt quality and task accuracy.
- Production systems needing consistent outputs.





Module 2: Building AI Agents on Databricks

LECTURE

# Custom Agent Development with Mosaic AI



© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).

# Select Your Model

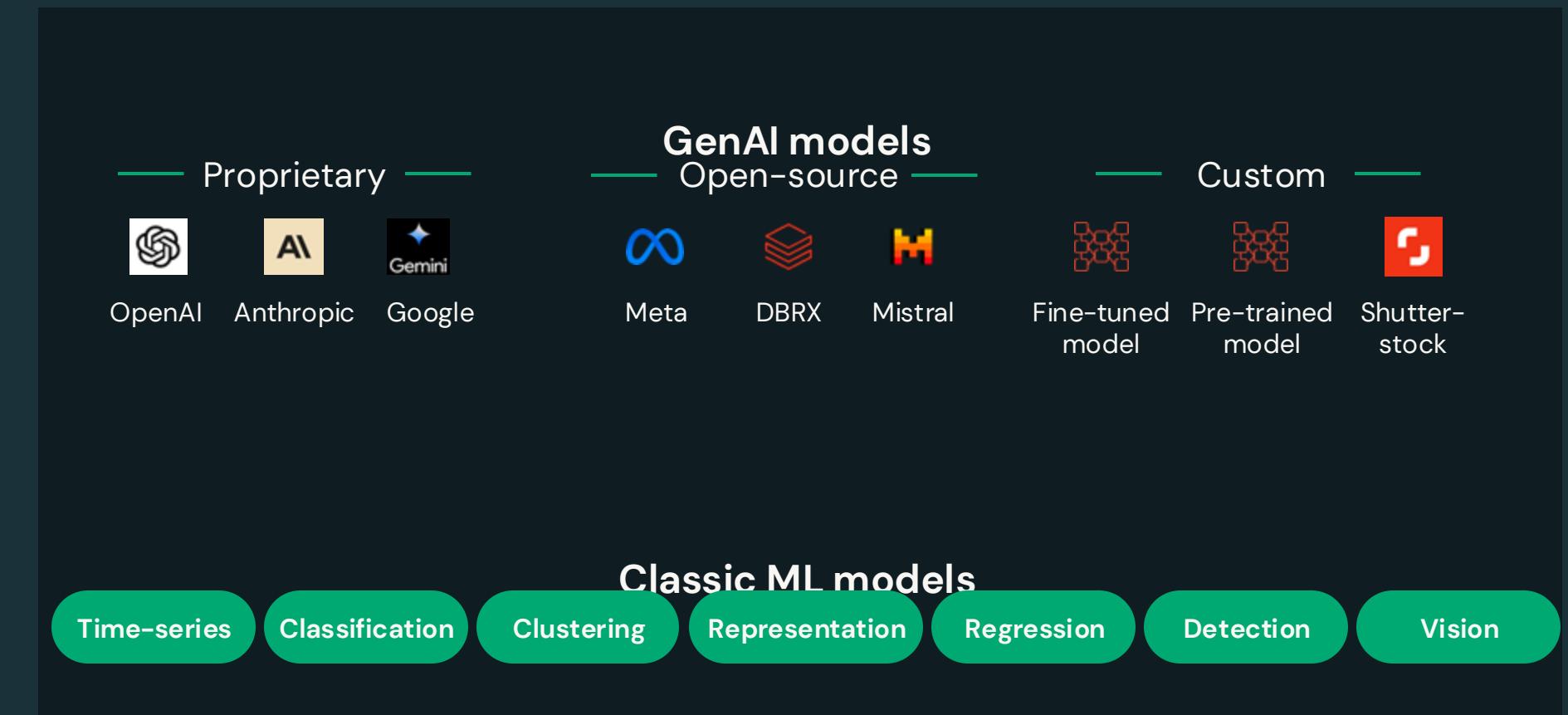
Use any model, from LLMs to Classic ML models specific to the task

Query, govern, & manage any model

- External (proprietary) models
- Open Source models
- Custom models

Available in multiple ways

- Endpoints
- Playground
- Batch/SQL



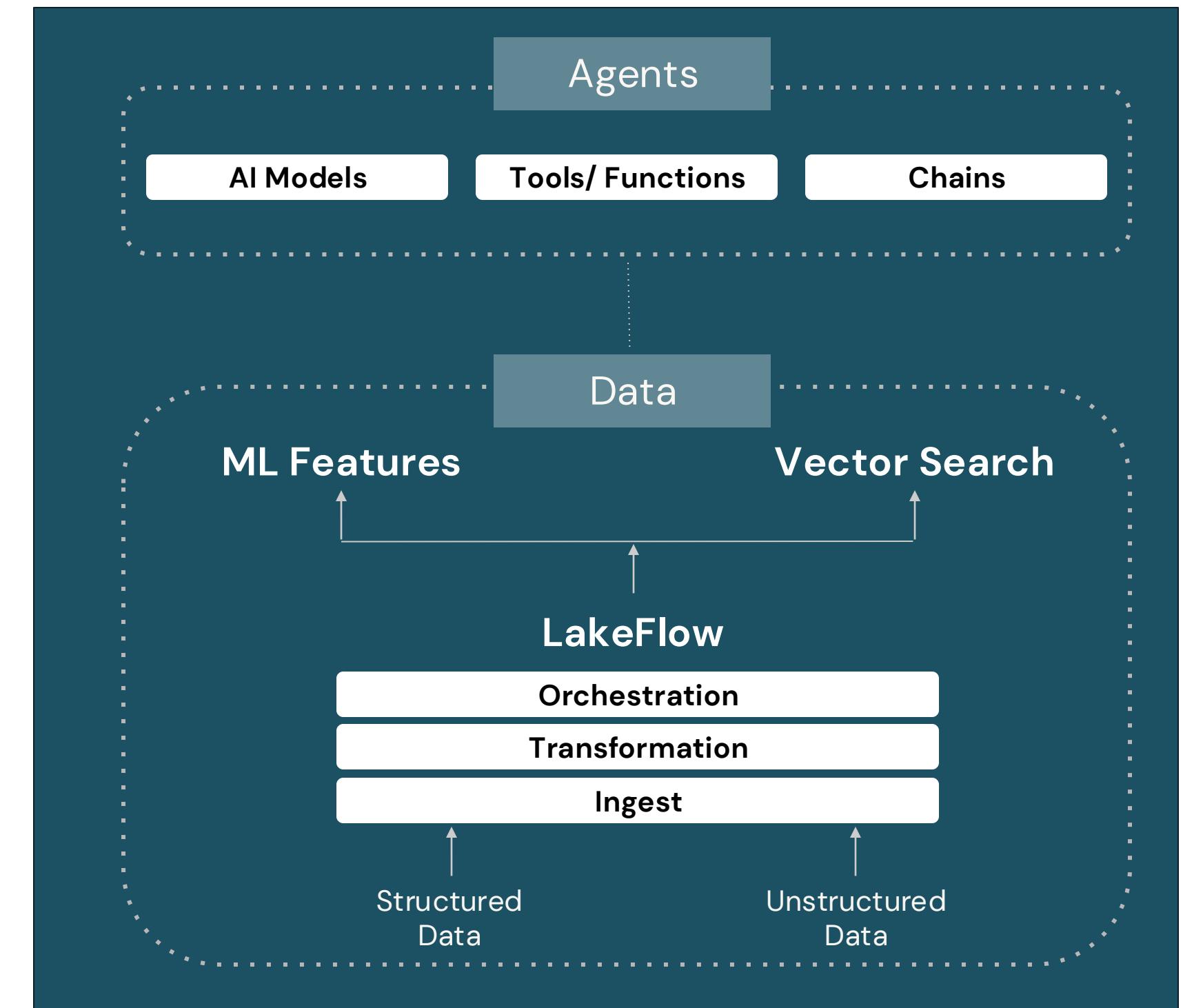
# Agents that reason over your data and leverage tools

Securely reason over your data with the only AI platform built on the lakehouse

Leverage tools and functions that provide correct answers, without hallucination

Automatically generate vector indexes and features from your data

Easily build and use agent components — customized models, functions, and chains



# Build Agents with Tools

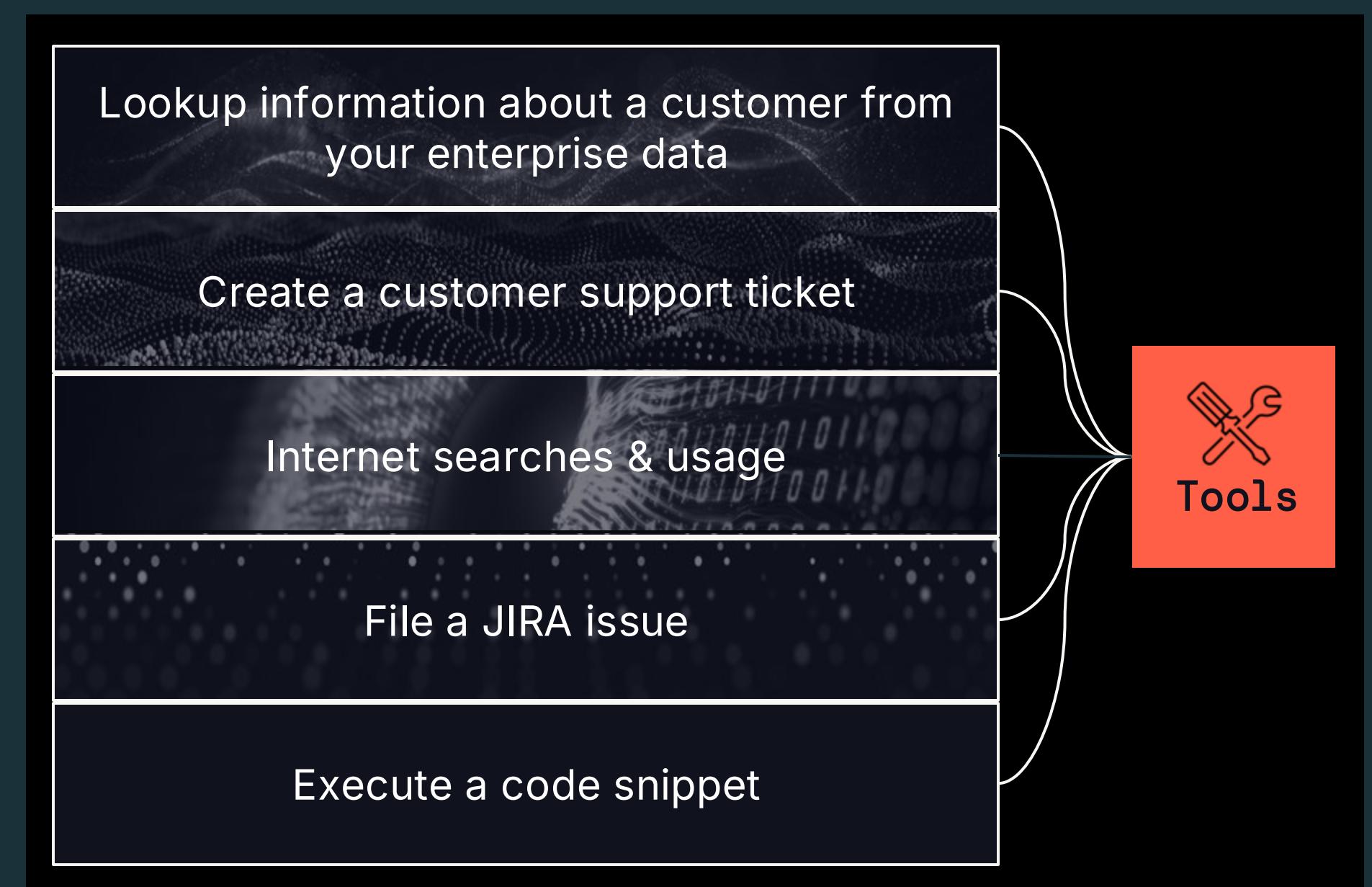
**Super power your agents with tools**

Author, publish, and share enterprise AI tools

Leverage trusted compute and credential management

Fully integrated into Unity Catalog

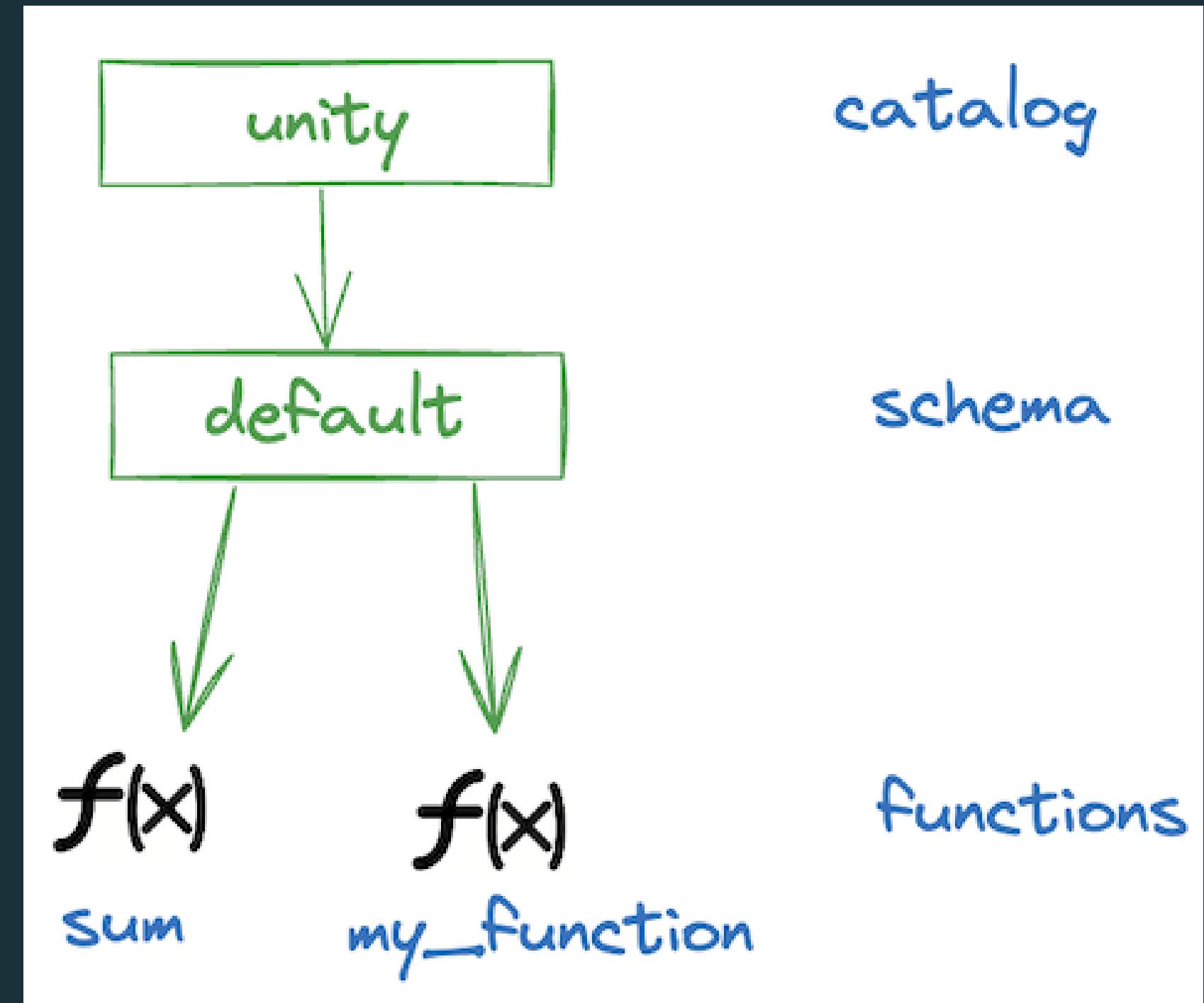
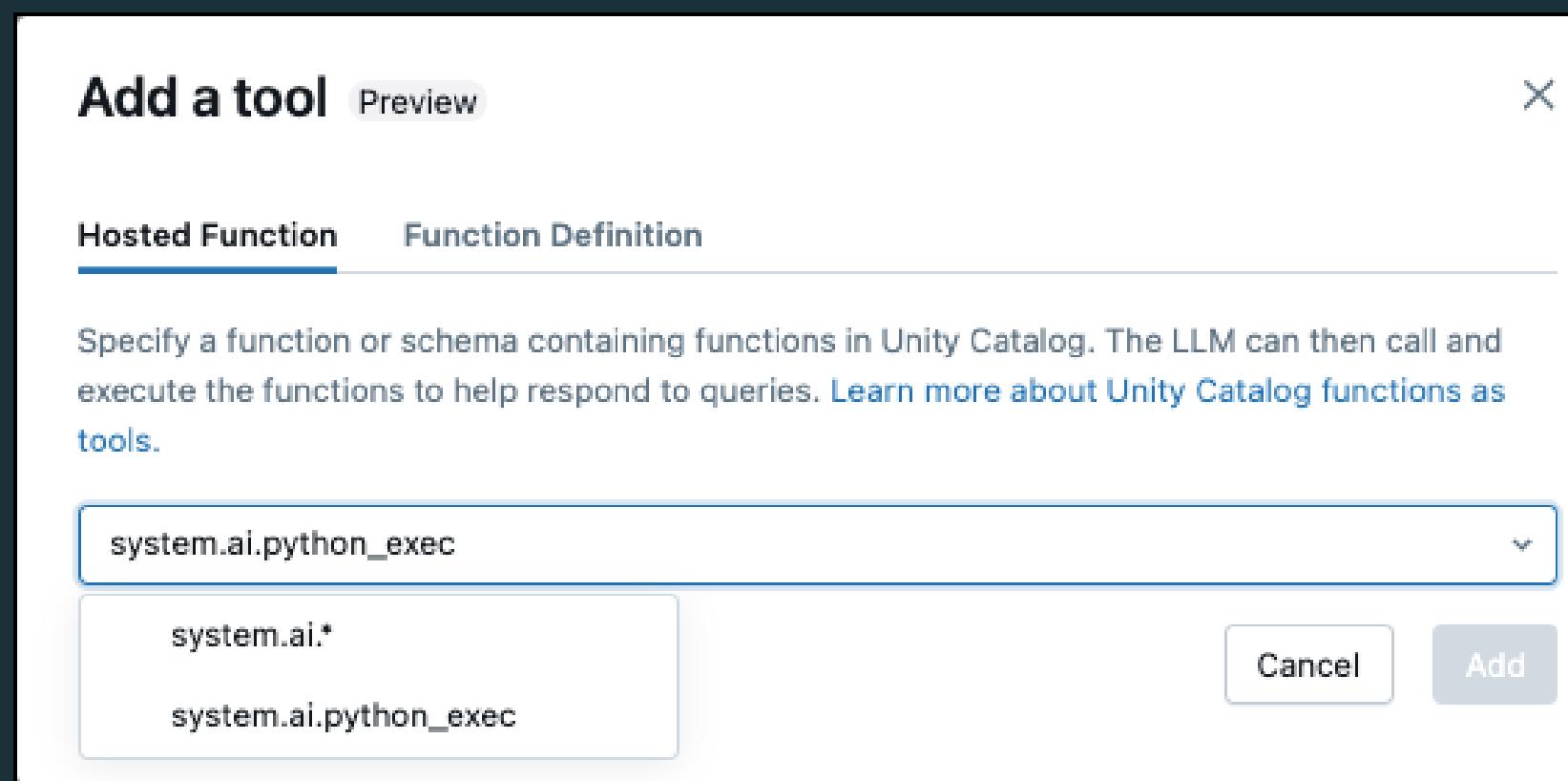
Supported across the stack



# Mosaic AI Tools in Unity Catalog

Allows users to register and govern Python and/or SQL functions in Unity Catalog

Simplifies tool calling for building AI Agent systems

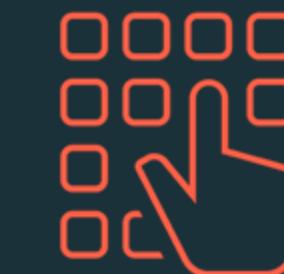


## Unity Catalog support for Model Context Protocol (MCP)

Open standard for  
connecting AI agents to  
data and tools



First-Party Servers for  
UC Tools



Easy Hosting for  
Servers through  
Databricks Apps



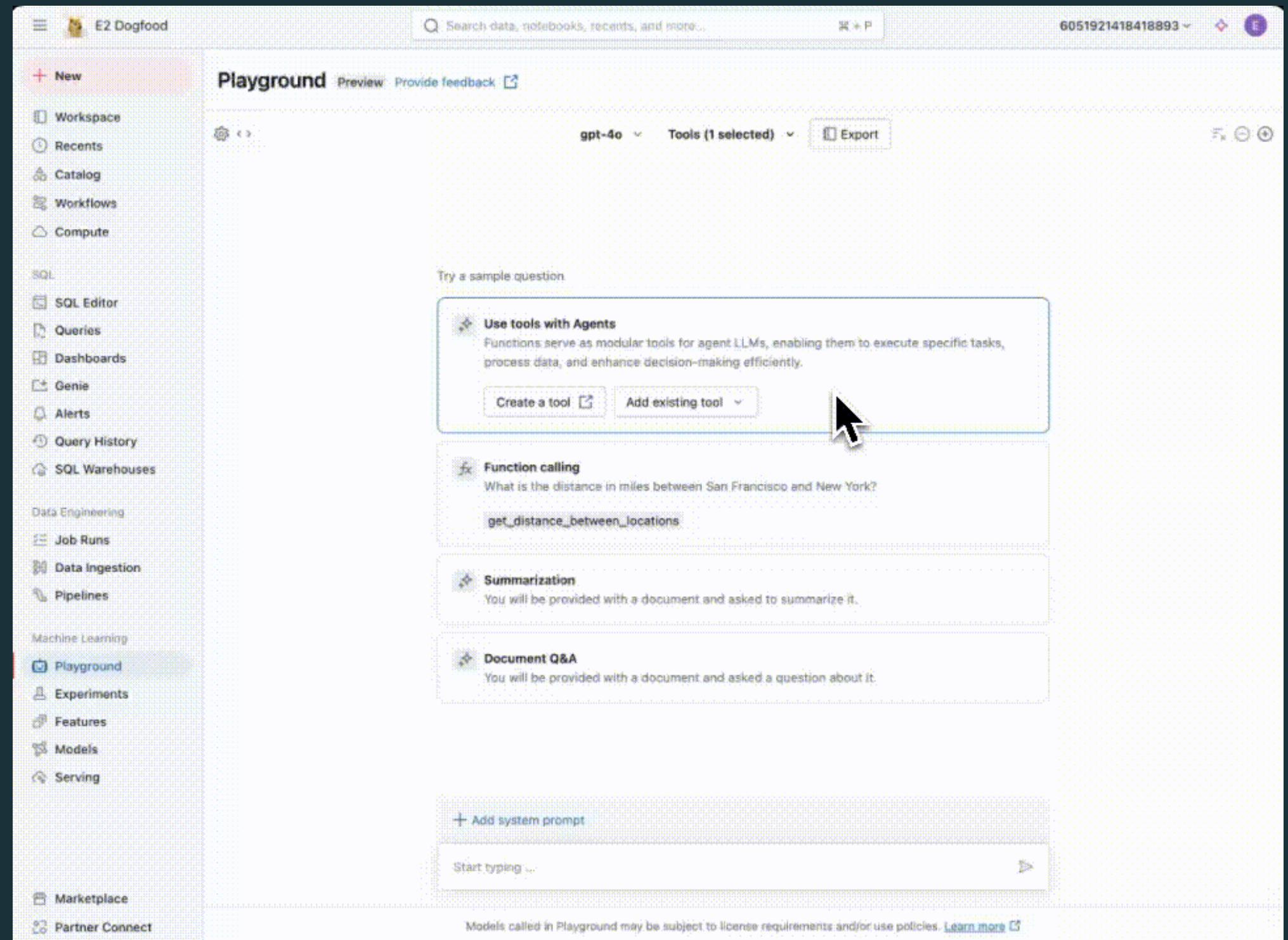
Use MCP in Playground  
& Agent Bricks

# Prototyping

**AI Playground** lets you quickly create and iterate on an Agent prototype without any code

Quickly assess initial quality with built-in LLM judges from **Agent Evaluation**

When finished, export production-ready code to further customize, evaluate, and deploy

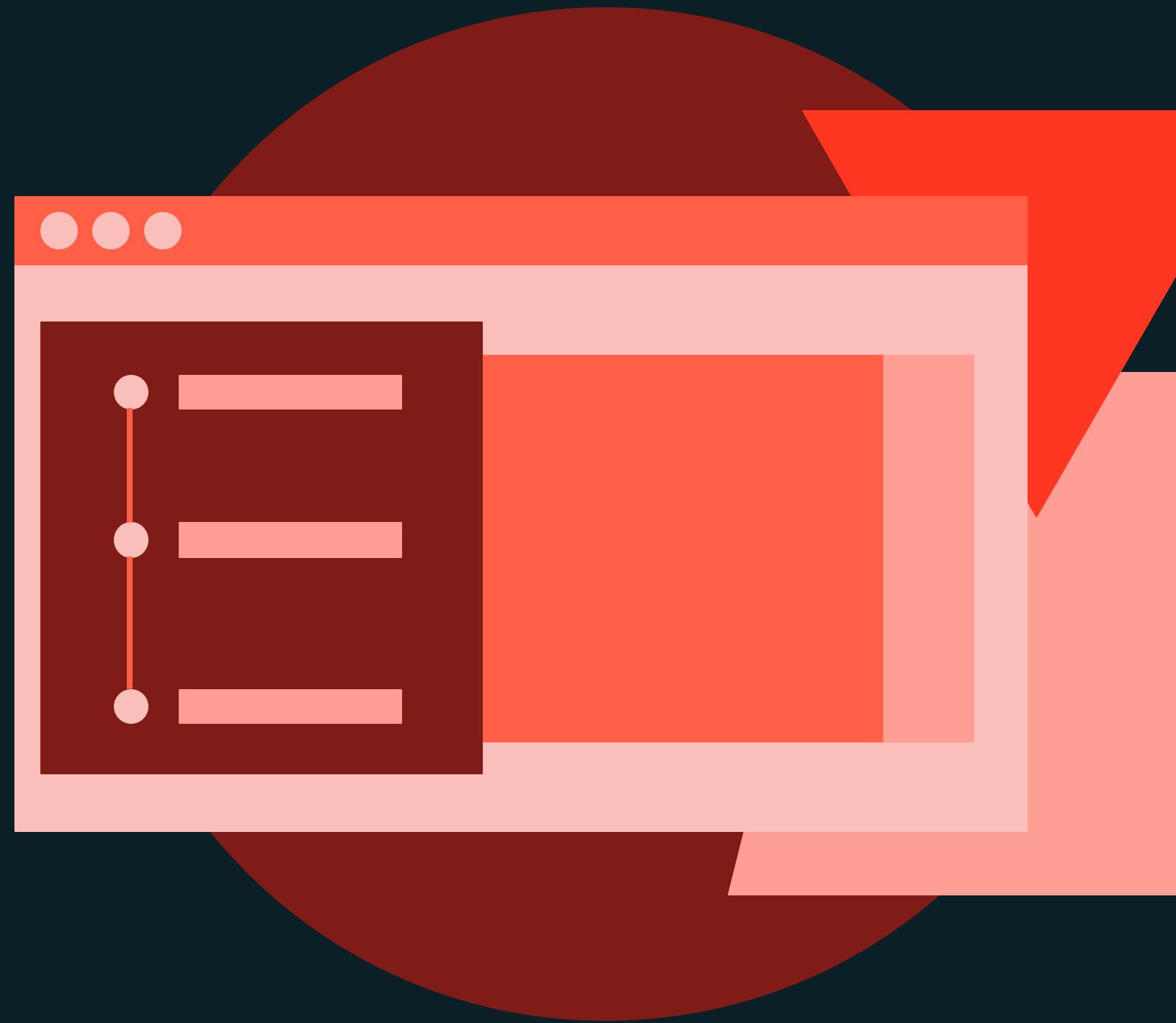




Module 2: Building AI Agents on Databricks

## DEMONSTRATION

# Build Agent Tools and Prototype with AI Playground



© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).

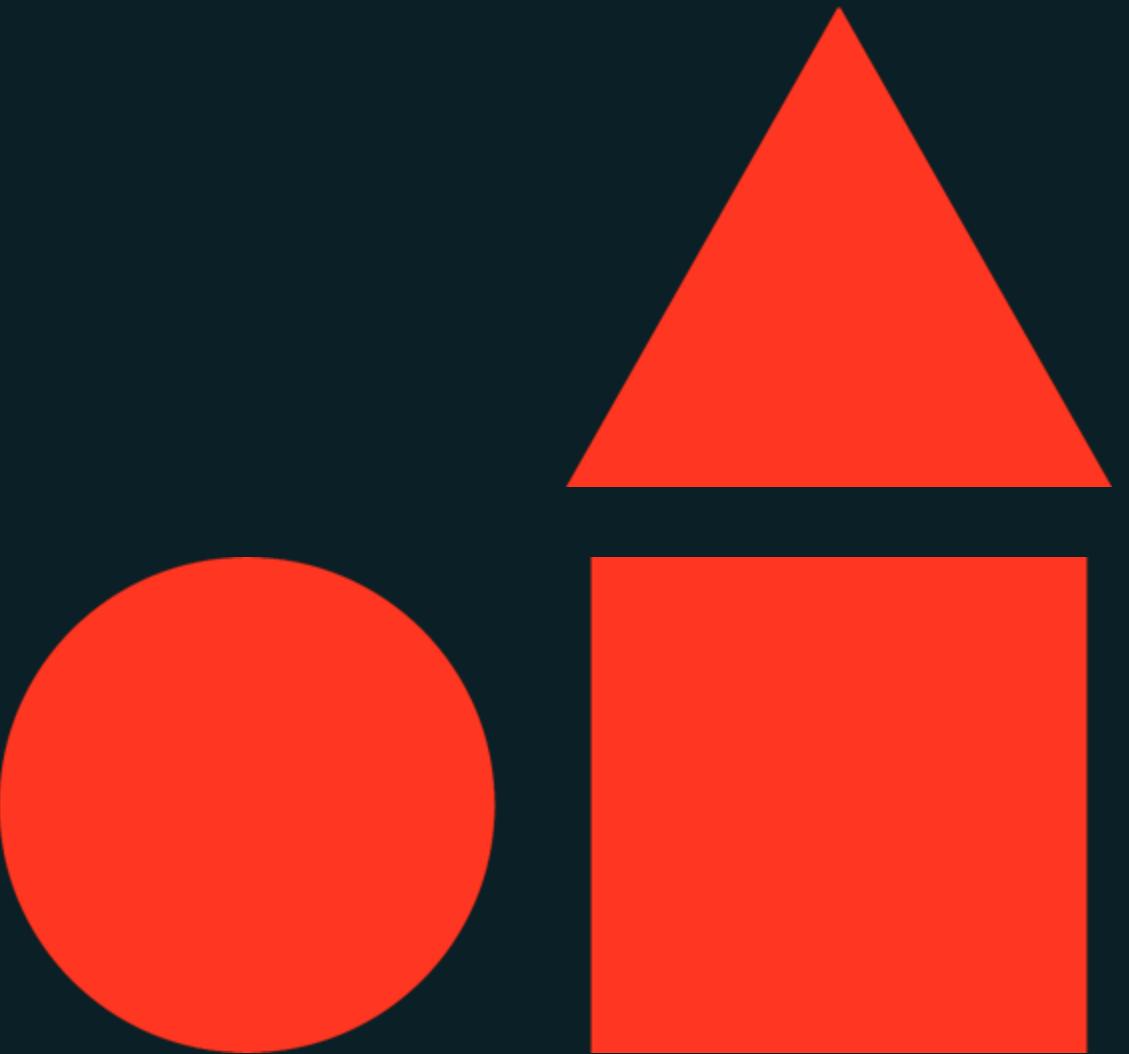


Module 3:

# Agent Evaluation and Deployment with Mosaic AI

---

Get Started with AI Agents on Databricks



# Objectives

- Explain the role of evaluation and deployment in the Agent Development Lifecycle, ensuring readiness for production.
- Use MLflow for agent evaluation, including logging, tracing, observability, and comparing multiple agent versions.
- Differentiate between deployment paradigms: batch, streaming, and real-time serving.
- Deploy agents and models to scalable production endpoints using Databricks Model Serving (UI, API, SDK).
- Recognize how governance and monitoring are enforced via Unity Catalog and integrated observability.
- Apply Lakehouse-unified serving features like automatic lookups and error reduction to streamline production deployments.





Module 3: Agent Evaluation and Deployment with Mosaic AI

LECTURE

# Agent Evaluation and MLflow



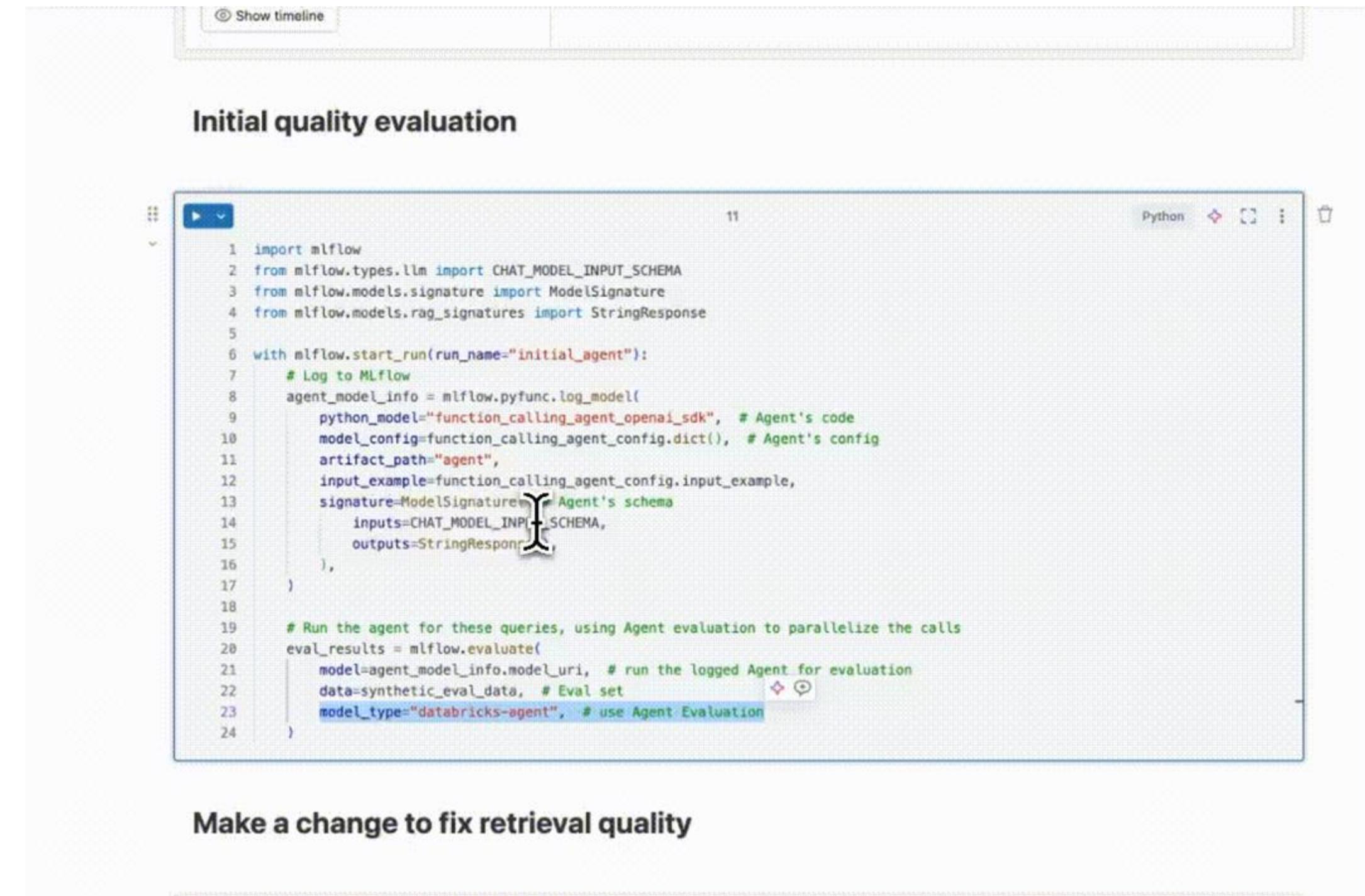
© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).

# Prepare for Production

Databricks LLM judges, created by Mosaic AI Research, provide best-in-class quality measurement and speed

**Agent Evaluation** automatically identifies the most likely root cause of quality issues

**Simple UIs** help you review Agent outputs and compare multiple versions of your Agent



The screenshot shows a Databricks notebook interface with a title bar 'Initial quality evaluation'. Below the title, there is a code editor containing Python code for agent evaluation. The code imports mlflow, CHAT\_MODEL\_INPUT\_SCHEMA, ModelSignature, and StringResponse from mlflow types, models, and rag\_signatures respectively. It then starts an MLflow run named 'initial\_agent', logs the model info, and runs the agent. Finally, it evaluates the agent using the logged model URI, synthetic eval data, and specifying the model type as 'databricks-agent'. A callout box points to the 'model\_type' parameter with the text 'Make a change to fix retrieval quality'.

```
1 import mlflow
2 from mlflow.types.llm import CHAT_MODEL_INPUT_SCHEMA
3 from mlflow.models.signature import ModelSignature
4 from mlflow.models.rag_signatures import StringResponse
5
6 with mlflow.start_run(run_name="initial_agent"):
7     # Log to MLflow
8     agent_model_info = mlflow.pyfunc.log_model(
9         python_model="function_calling_agent_openai_sdk", # Agent's code
10        model_config=function_calling_agent_config.dict(), # Agent's config
11        artifact_path="agent",
12        input_example=function_calling_agent_config.input_example,
13        signature=ModelSignature(), # Agent's schema
14        inputs=CHAT_MODEL_INPUT_SCHEMA,
15        outputs=StringResponse(),
16    ),
17
18
19     # Run the agent for these queries, using Agent evaluation to parallelize the calls
20     eval_results = mlflow.evaluate(
21         model=agent_model_info.model_uri, # run the logged Agent for evaluation
22         data=synthetic_eval_data, # Eval set
23         model_type="databricks-agent", # use Agent Evaluation
24     )
```

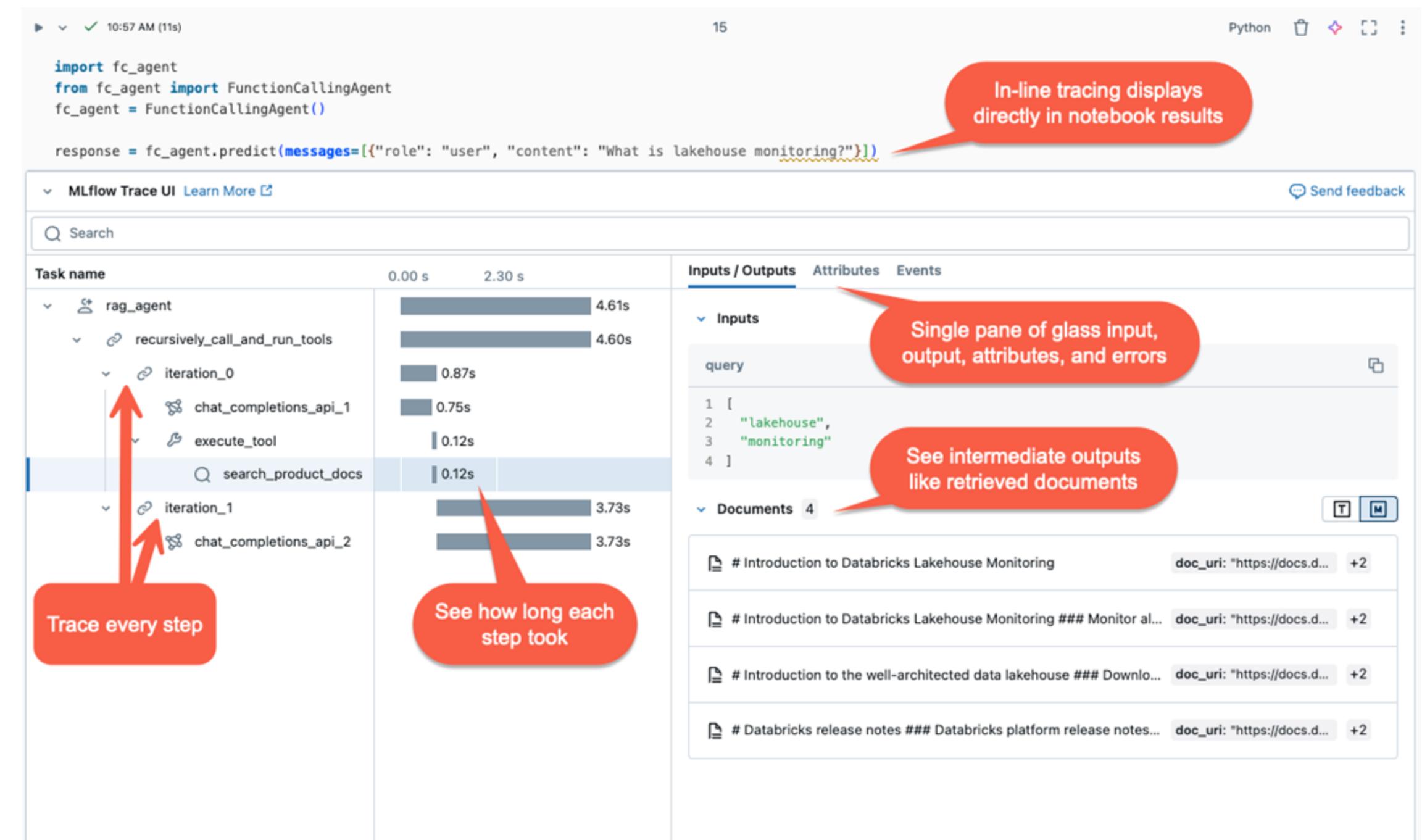
Make a change to fix retrieval quality



# Prepare for Production

**MLflow Tracing** enables you to debug complex Agent logic quickly

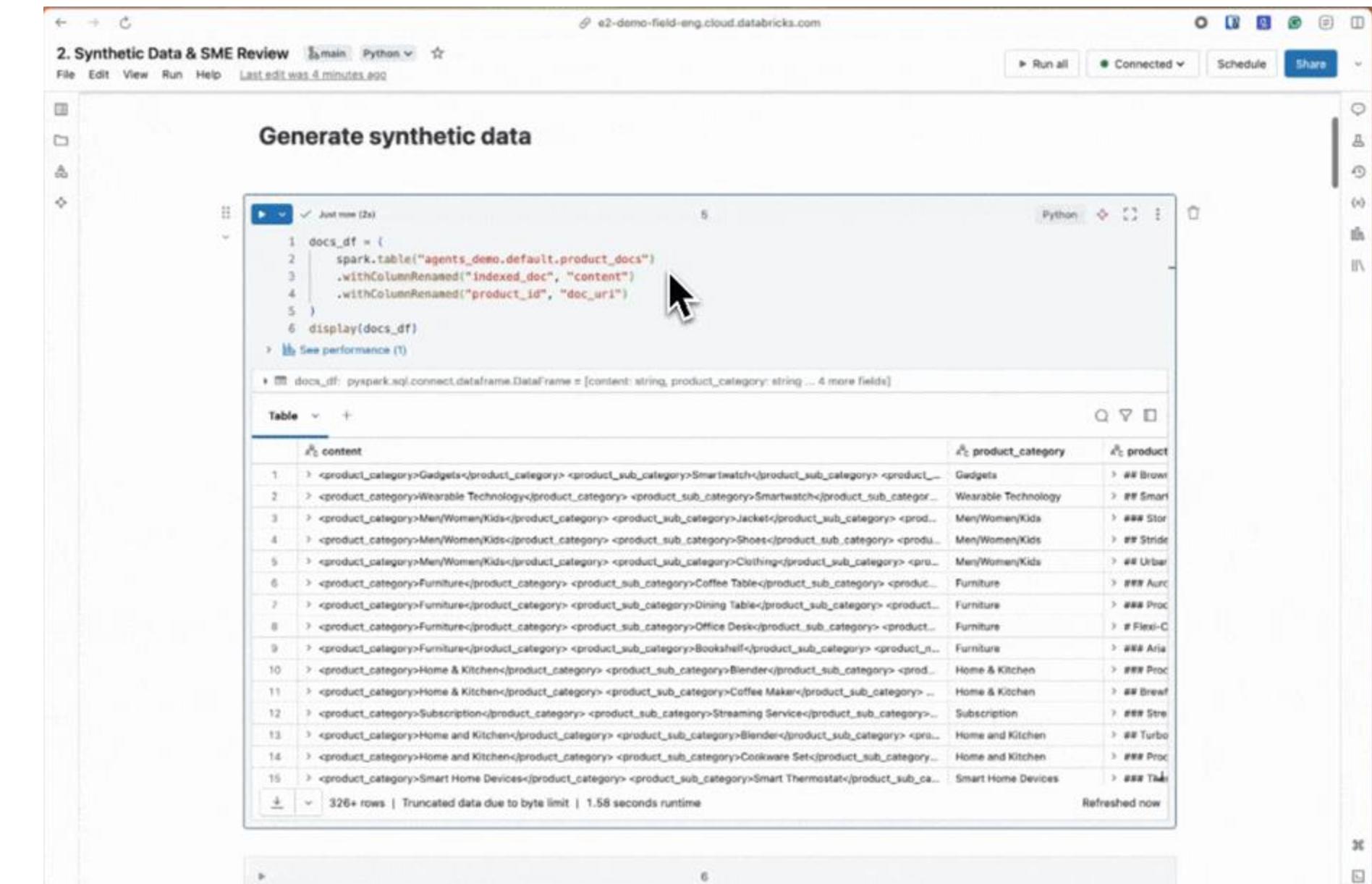
**MLflow logging** lets you track versions of Agent code & config in Delta Tables



# Label Evaluation Datasets

**High quality, research-backed synthetic data** generation API immediately unblocks quality evaluation *without* SMEs

Generates a ground truth list of facts, rather than a verbose written response, which is more efficient for SMEs to review



The screenshot shows a Databricks notebook interface with the following details:

- Title:** 2. Synthetic Data & SME Review
- Code:**

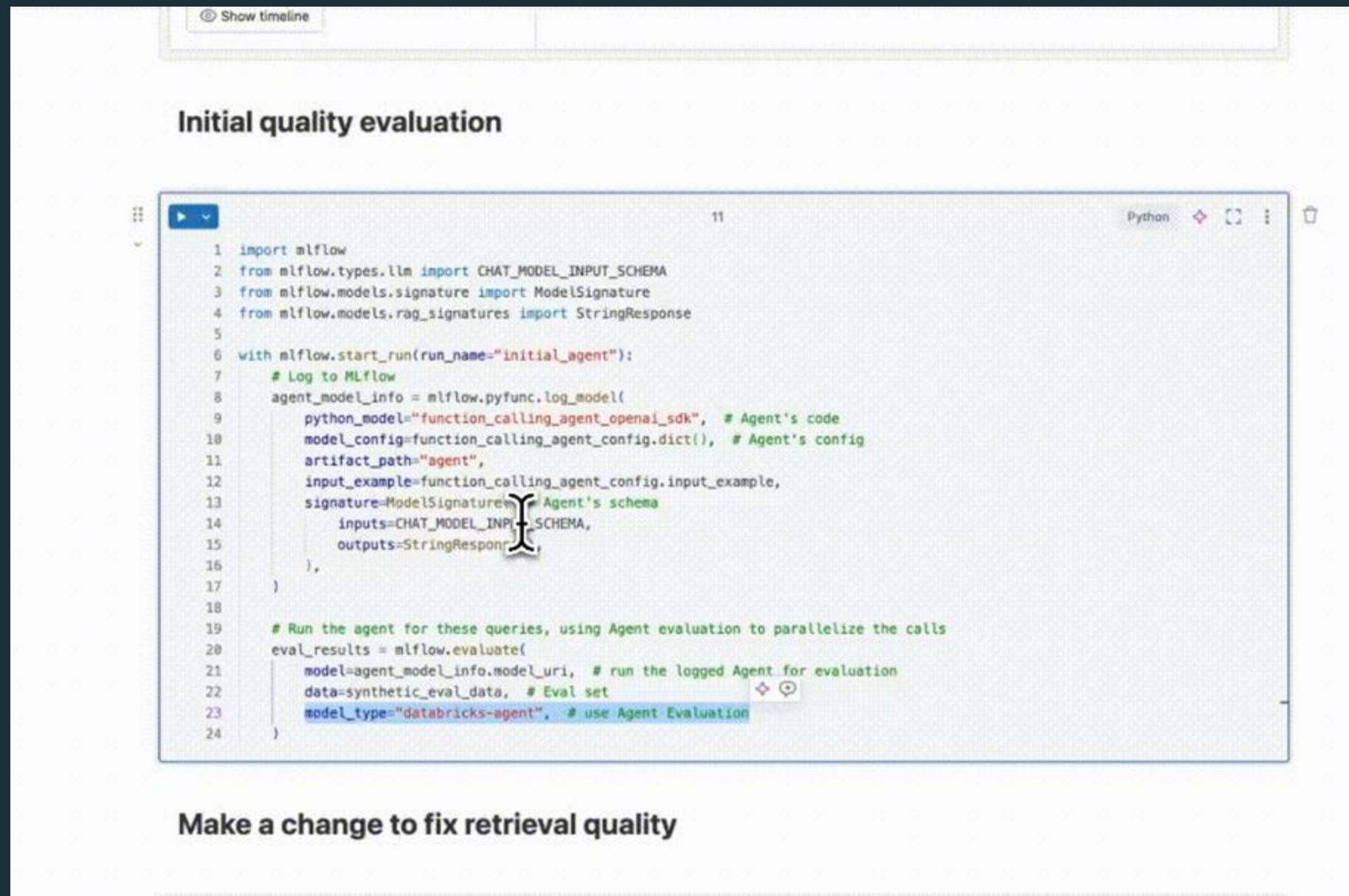
```
1 docs_df = (
2     spark.table("agents_demo.default.product_docs")
3     .withColumnRenamed("indexed_doc", "content")
4     .withColumnRenamed("product_id", "doc_uri")
5 )
6 display(docs_df)
> See performance (1)
```
- Data Preview:** A table titled "docs\_df" showing columns "content" and "product\_category". The data consists of 326 rows of product descriptions and their categories.
- Runtime:** 1.58 seconds

content	product_category
> <product_category>Gadgets</product_category> <product_sub_category>Smartwatch</product_sub_category> <product_sub_sub_category>Gadgets	> ## Brown
> <product_category>Wearable Technology</product_category> <product_sub_category>Smartwatch</product_sub_category> <product_sub_sub_category>Wearable Technology	> ## Smart
> <product_category>Men/Women/Kids</product_category> <product_sub_category>Jacket</product_sub_category> <product_sub_sub_category>Men/Women/Kids	> ## Stor
> <product_category>Men/Women/Kids</product_category> <product_sub_category>Shoes</product_sub_category> <product_sub_sub_category>Men/Women/Kids	> ## Stride
> <product_category>Men/Women/Kids</product_category> <product_sub_category>Clothing</product_sub_category> <product_sub_sub_category>Men/Women/Kids	> ## Urban
> <product_category>Furniture</product_category> <product_sub_category>Coffee Table</product_sub_category> <product_sub_sub_category>Furniture	> ## Auct
> <product_category>Furniture</product_category> <product_sub_category>Dining Table</product_sub_category> <product_sub_sub_category>Furniture	> ## Proc
> <product_category>Furniture</product_category> <product_sub_category>Office Desk</product_sub_category> <product_sub_sub_category>Furniture	> # Flexi-C
> <product_category>Furniture</product_category> <product_sub_category>Bookshelf</product_sub_category> <product_sub_sub_category>Furniture	> ## Aria
> <product_category>Home & Kitchen</product_category> <product_sub_category>Blender</product_sub_category> <product_sub_sub_category>Home & Kitchen	> ## Proc
> <product_category>Home & Kitchen</product_category> <product_sub_category>Coffee Maker</product_sub_category> <product_sub_sub_category>Home & Kitchen	> ## Breakfast
> <product_category>Subscription</product_category> <product_sub_category>Streaming Service</product_sub_category> <product_sub_sub_category>Subscription	> ## Stre
> <product_category>Home and Kitchen</product_category> <product_sub_category>Blender</product_sub_category> <product_sub_sub_category>Home and Kitchen	> ## Turbo
> <product_category>Home and Kitchen</product_category> <product_sub_category>Cookware Set</product_sub_category> <product_sub_sub_category>Home and Kitchen	> ## Proc
> <product_category>Smart Home Devices</product_category> <product_sub_category>Smart Thermostat</product_sub_category> <product_sub_sub_category>Smart Home Devices	> ## Tech

**Easy-to-use** to quickly validate synthetic data vs. starting from scratch



# Offline evaluation



The screenshot shows a Jupyter Notebook cell with the title "Initial quality evaluation". The cell contains the following Python code:

```
1 import mlflow
2 from mlflow.types.llm import CHAT_MODEL_INPUT_SCHEMA
3 from mlflow.models.signature import ModelSignature
4 from mlflow.models.rag_signatures import StringResponse
5
6 with mlflow.start_run(run_name="initial_agent"):
7     # Log to MLflow
8     agent_model_info = mlflow.pyfunc.log_model(
9         python_model="function_calling_agent_openai_sdk", # Agent's code
10        model_config=function_calling_agent_config.dict(), # Agent's config
11        artifact_path="agent",
12        input_example=function_calling_agent_config.input_example,
13        signature=ModelSignature( # Agent's schema
14            inputs=CHAT_MODEL_INPUT_SCHEMA,
15            outputs=StringResponse(),
16        ),
17    )
18
19    # Run the agent for these queries, using Agent evaluation to parallelize the calls
20    eval_results = mlflow.evaluate(
21        model=agent_model_info.model_uri, # run the logged Agent for evaluation
22        data=synthetic_eval_data, # Eval set
23        model_type="databricks-agent", # use Agent Evaluation
24    )
```

Below the code cell, the text "Make a change to fix retrieval quality" is displayed.

Validate agents meet business requirements

Test and validate new versions

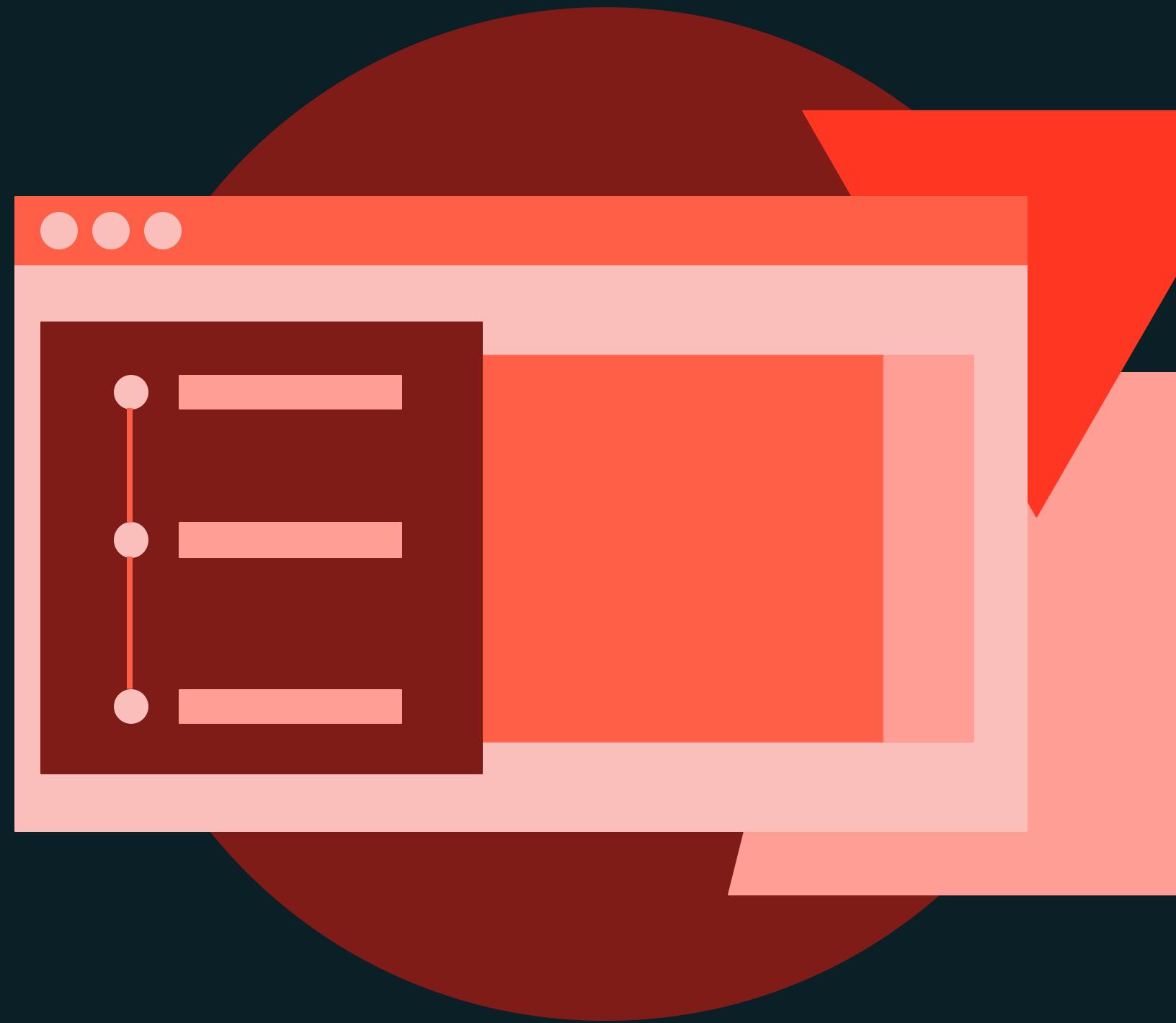




Module 3: Agent Evaluation and Deployment with Mosaic AI

## DEMONSTRATION

# From AI Playground to Deployment



© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).



Module 3: Agent Evaluation and Deployment with Mosaic AI

## LAB EXERCISE

# Creating Tools and Testing in the Playground



© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).



Module 3: Agent Evaluation and Deployment with Mosaic AI

LECTURE

# Deploying Agents & Model Serving



© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).

# Deployment Paradigms

Similar to traditional machine learning extended to GenAI

Latency Decreases

Throughput Decreases

## Batch

Generate (and store) completions on a table of (text) inputs/prompts

## Streaming

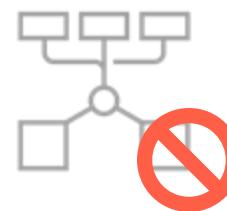
Generate (and store) completions on micro-batches of (text) inputs/prompts as they're processed

## Real-time

Generate completions asynchronously in real-time on individual input/prompt

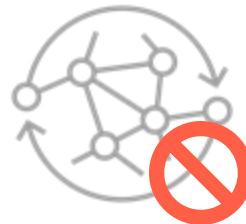
# Challenges with Deploying Real-time AI

Most AI systems don't get into production



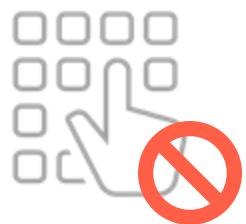
## Infrastructure is hard

Real-time AI systems require fast and scalable serving infrastructure, which is costly to build and maintain



## Deploying real time models needs disparate tools

Data teams use diverse tools to develop models  
Customers use separate platforms for data, LLMs and Serving, adding complexity and cost



## Operating production AI requires expert resources

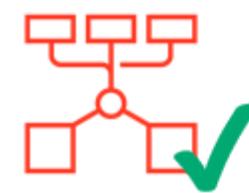
Steep learning curve of deployment tools.

Model deployment is bottlenecked by limited engineering resources, limiting the ability to scale AI



# Mosaic AI Model Serving

Integrate your model into your websites and applications as an API



## Production-Grade Serving

**Highly available**, low latency, scalable serving that works for small and large workloads



## Accelerate deployments with Lakehouse-Unified Serving

Automatic feature lookups, monitoring and unified governance that **automates deployment** and reduce errors



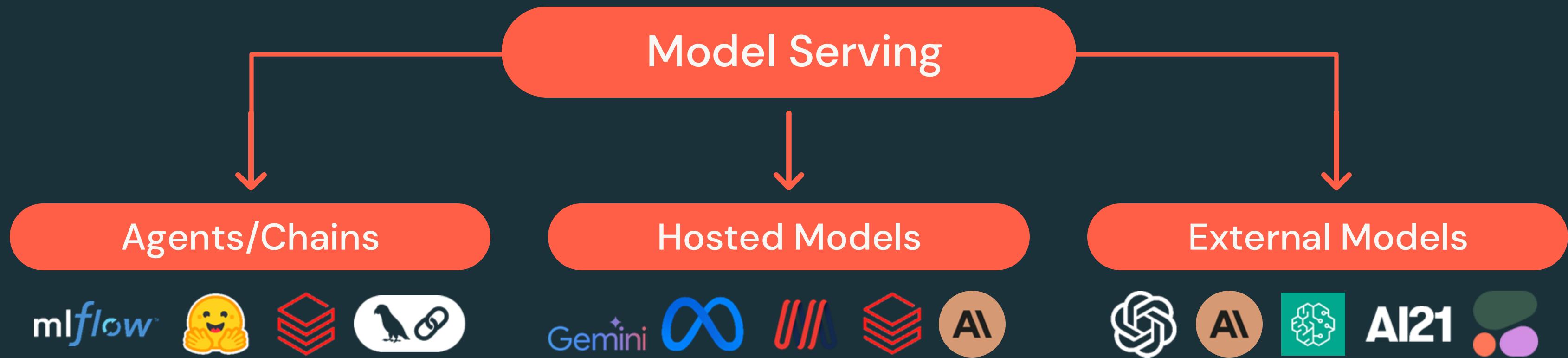
## Simplified Deployment

Simple and flexible deployment through **UI or API**



# Databricks Model Serving

Unified UI, API & SDK for managing all types of AI Models



Access any AI Model, be it a fully custom model, an agent/chain model, a Databricks-managed Foundation Model, or a third-party Foundation Model

Govern and Monitor all Models in one place

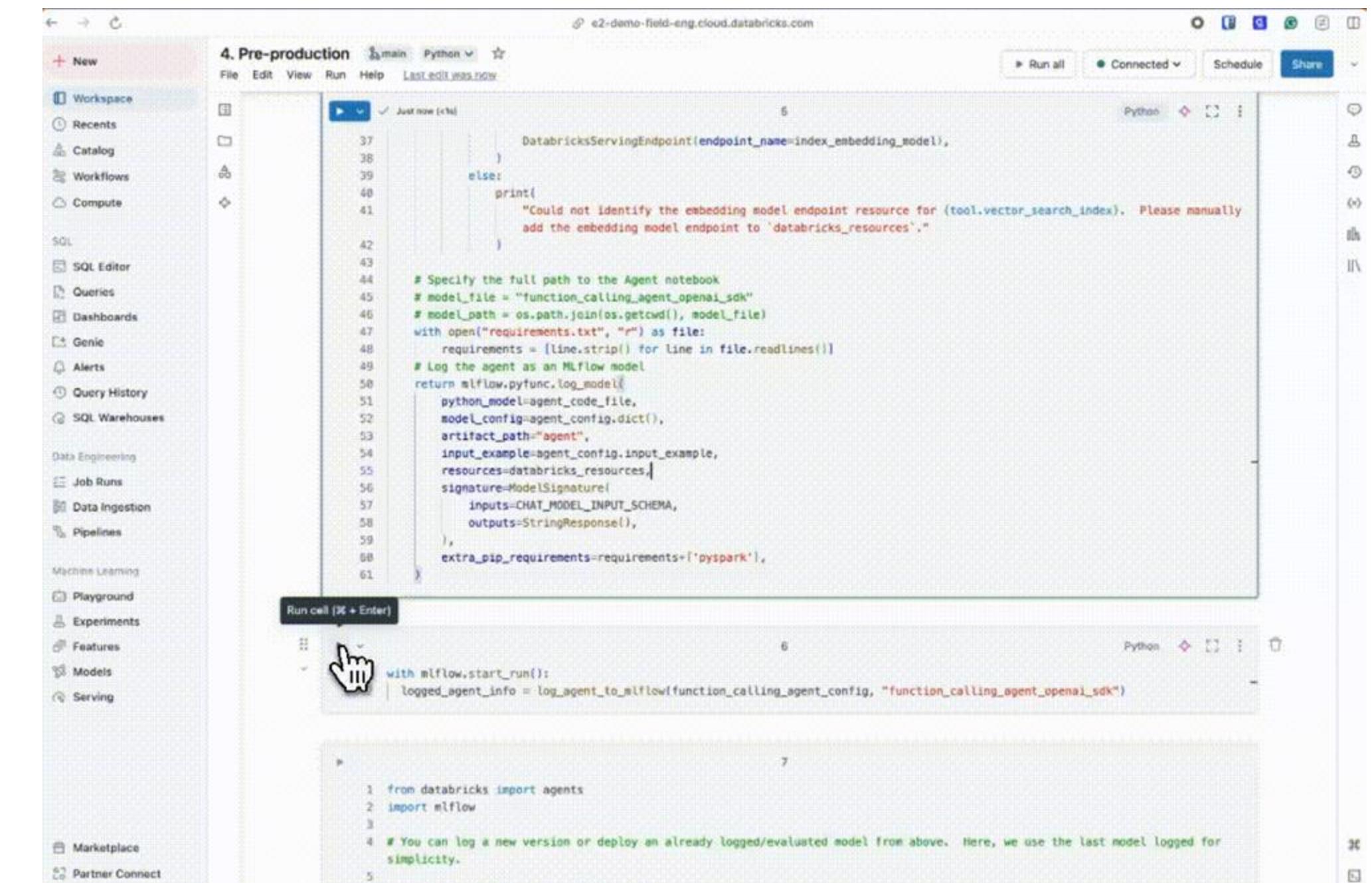
Query Models via unified interface (Single API, SDK and UI)

# Collect Feedback From Stakeholders

Human in the loop feedback logged to Unity Catalog for building a golden, labeled dataset and fine-tuning

**Review App** provides a pre-built chat app that makes it easy to collect feedback from SMEs

Use **Agent Evaluation + AI/BI dashboard** to pinpoint quality issues based on LLM judges and user feedback



The screenshot shows a Databricks notebook interface with the title "4. Pre-production". The left sidebar contains navigation links for Workspace, Recents, Catalog, Workflows, Compute, SQL, Data Engineering, Machine Learning, and Marketplace. The main area displays two code cells. The top cell contains Python code for setting up a Databricks Serving Endpoint and logging agent configuration to MLflow:

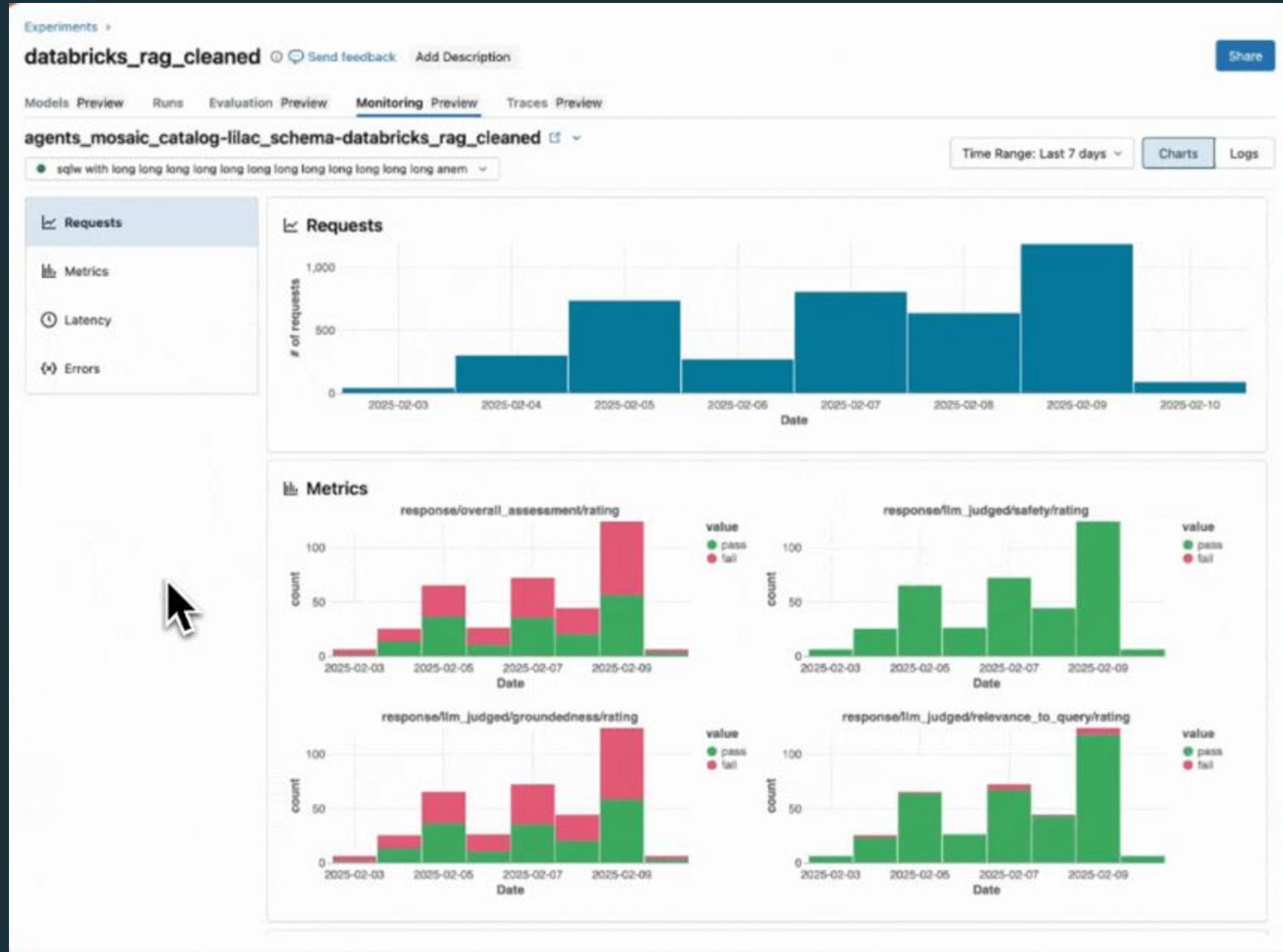
```
37     DatabricksServingEndpoint(endpoint_name=index_embedding_model),
38
39     else:
40         print(
41             "Could not identify the embedding model endpoint resource for (tool.vector_search_index). Please manually
42             add the embedding model endpoint to 'databricks_resources'."
43
44 # Specify the full path to the Agent notebook
45 # model_file = "function_calling_agent_openai_sdk"
46 # model_path = os.path.join(os.getcwd(), model_file)
47 with open("requirements.txt", "r") as file:
48     requirements = [line.strip() for line in file.readlines()]
49 # Log the agent as an MLflow model
50 return mlflow.pyfunc.log_model(
51     python_model=agent_code_file,
52     model_config=agent_config.dict(),
53     artifact_path="agent",
54     input_example=agent_config.input_example,
55     resources=databricks_resources,
56     signature=ModelSignature(
57         inputs=CHAT_MODEL_INPUT_SCHEMA,
58         outputs=StringResponse(),
59     ),
60     extra_pip_requirements=requirements+[pyspark],
61 )
```

A "Run cell ⌘+Enter" button is located below this cell.

The bottom cell shows a cursor hovering over the first line of code, which starts with "with mlflow.start\_run():". The code continues to define a function for logging agent info to MLflow.



# Online monitoring



Continuously validate quality / performance

Quickly fix issues

Unified observability



Module 4:

# Production-Ready Agents with Agent Bricks

---

Get Started with AI Agents on Databricks



© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).

# Objectives

- Explain why building and evaluating agents at scale is challenging, including trade-offs between cost and quality and the need for automation
- Describe how Agent Bricks simplify agent development by providing pre-built, research-backed components for optimization, evaluation, and deployment
- Leverage advanced AI techniques integrated into Agent Bricks (e.g., prompt optimization, critique filtering, adaptive LLM judges, RLHF, synthetic data generation)
- Recognize how Agent Bricks fit into the overall Mosaic AI ecosystem for enterprise-grade governance, monitoring, and deployment.





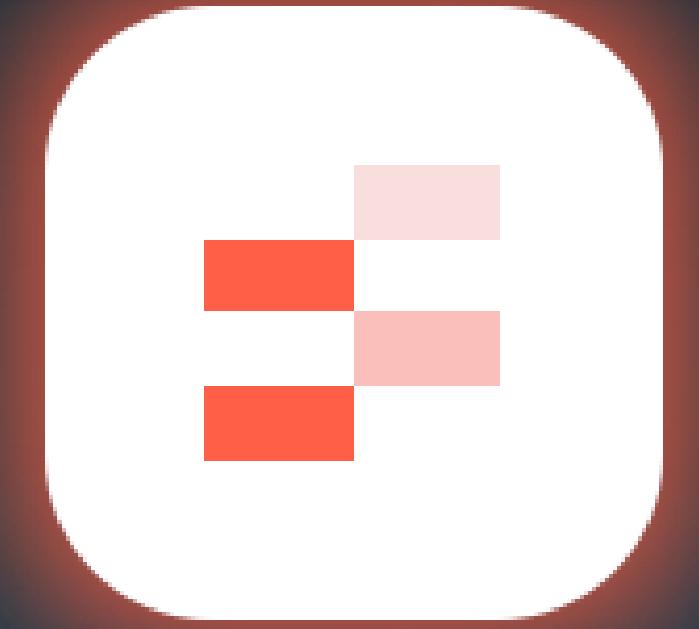
Module 4: Production-ready Agents with Agent Bricks

LECTURE

# Introduction to Agent Bricks



© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).



# Agent Bricks

Build and optimize agents on your data

# Building Agents today is challenging



**Evaluation is difficult**



**Too many knobs**



**Trade-off cost versus quality**

# Building Agents today is challenging

85%

of users rely only on  
manual checking of  
agents



Too many  
knobs

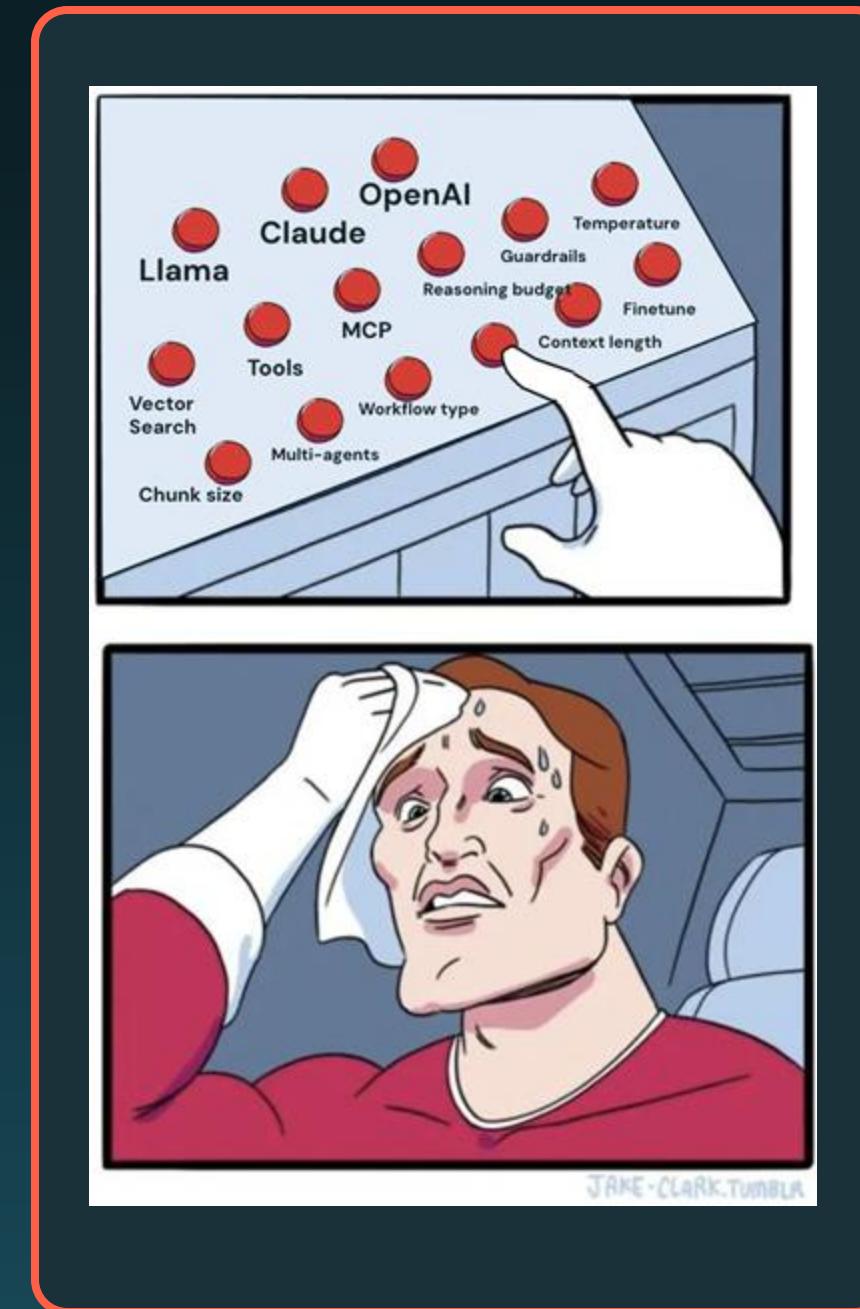


Trade-off cost  
versus quality

# Building Agents today is challenging



Evaluation is difficult



Trade-off cost  
versus quality

# Building Agents today is challenging

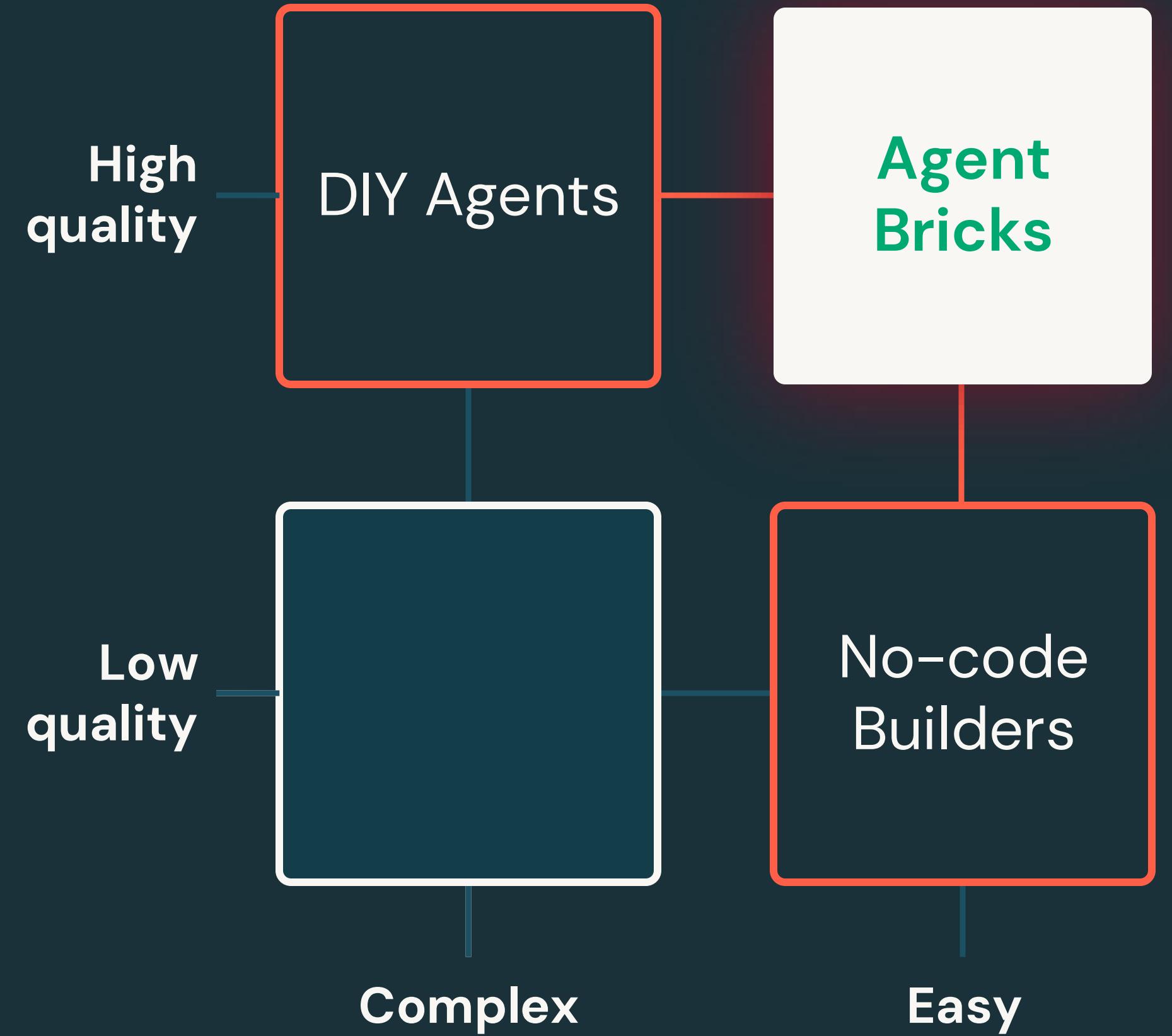


**Evaluation is difficult**



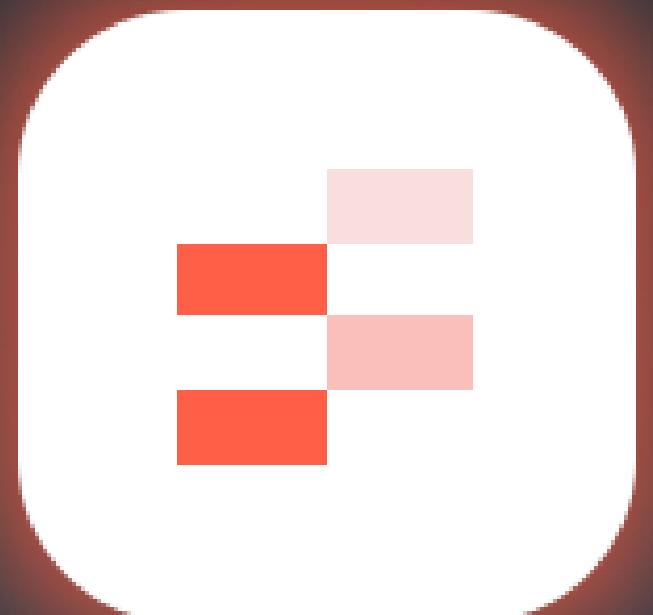
**Too many knobs**





# Agent Bricks

Powered by Databricks Mosaic AI Research



Best-of-N

Test-time adaptive optimization

Deep research agents

Fine-tuning

Critique filtering

Prompt optimization

Automated evaluation

Agentic knowledge graphs

o1-style reasoning

Contextual retrieval

RLVR

Adaptive LLM judges

Constitutional AI

MixAttention

Value-guided decoding

RLHF

Synthetic data generation

Multi-agentic patterns

# Agent Bricks

1

Select a task, declare a high-level description of the agent

**Agent Bricks** Beta

Provide feedback  Use PDFs in Agent Bricks 

Choose your use case

 **Information Extraction**  
Extract key information, classify content, or summarize text from documents into a structured JSON.

 **Custom LLM**  
Specialize an LLM endpoint to perform custom text tasks (i.e. content generation, chat) aligned within your domain-specific guidelines.

 **Knowledge Assistant**  
Turn your docs into an expert AI chatbot.

 **Multi-Agent Supervisor**  
Design an AI system that brings Genie, agents, and tools together.

# Agent Bricks

1 Select a task, declare a high-level description of the agent

2 Agent Bricks auto-magically:



Creates evaluation benchmarks

Automatically create quality benchmarks on **your task**



Accuracy

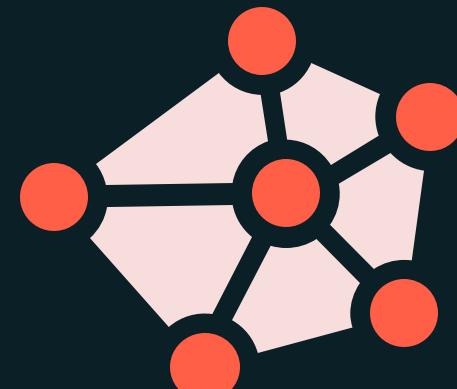


Product Relevance



Customer Churn

Generate new training data



# Agent Bricks

1 Select a task, declare a high-level description of the agent

2 Agent Bricks auto-magically:



Creates evaluation benchmarks



Auto-optimizes the agent

Searches and compose different techniques for quality

Prompt optimization

Fine-tuning

Tool choices

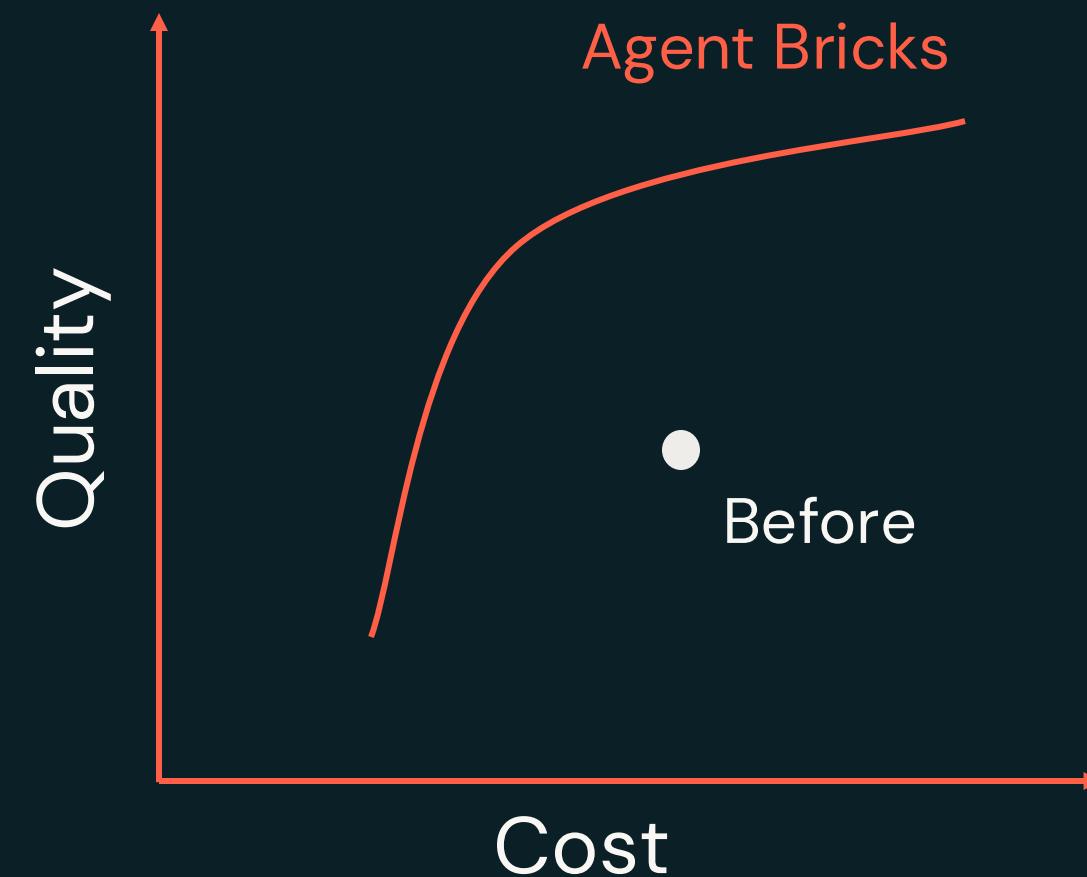
Reward model filtering

Custom LLM Judges

RLHF

# Agent Bricks

- 1 Select a task, declare a high-level description of the agent
- 2 Agent Bricks auto-magically:
  - Creates evaluation benchmarks
  - Auto-optimizes the agent
  - Cost *and* Quality
- 3 Deploy and iterate on quality



# Agent Bricks

Agent Bricks Beta

[Provide feedback ↗](#)

[Use PDFs in Agent Bricks](#)

Choose your use case

## Information Extraction

Extract key information, classify content, or summarize text from documents into a structured JSON.

## Knowledge Assistant

Turn your docs into an expert AI chatbot.

## Multi-Agent Supervisor

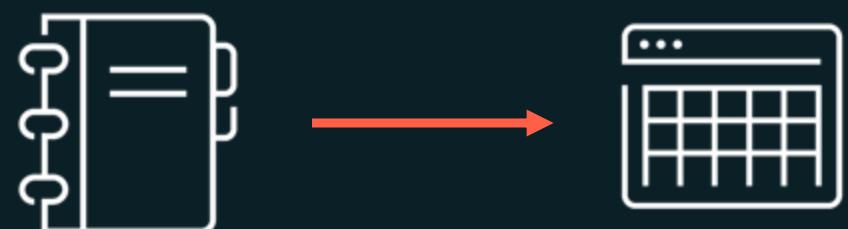
Design an AI system that brings Genie, agents, and tools together.

## Custom LLM

Specialize an LLM endpoint to perform custom text tasks (i.e. content generation, chat) aligned within your domain-specific guidelines.

## Information Extraction

*Extract structured data from documents into tables*



# Agent Bricks

Agent Bricks Beta

[Provide feedback ↗](#)

[Use PDFs in Agent Bricks](#)

Choose your use case

## Information Extraction

Extract key information, classify content, or summarize text from documents into a structured JSON.

## Knowledge Assistant

Turn your docs into an expert AI chatbot.

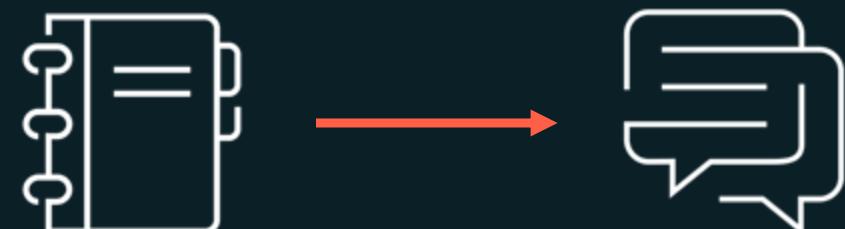
## Multi-Agent Supervisor

Design an AI system that brings Genie, agents, and tools together.

## Custom LLM

Specialize an LLM endpoint to perform custom text tasks (i.e. content generation, chat) aligned within your domain-specific guidelines.

**Knowledge Assistant**  
*From documents to conversational agents*



# Agent Bricks

Agent Bricks Beta

[Provide feedback ↗](#)

[Use PDFs in Agent Bricks](#)

Choose your use case

## Information Extraction

Extract key information, classify content, or summarize text from documents into a structured JSON.

## Knowledge Assistant

Turn your docs into an expert AI chatbot.

## Multi-Agent Supervisor

Design an AI system that brings Genie, agents, and tools together.

## Custom LLM

Specialize an LLM endpoint to perform custom text tasks (i.e. content generation, chat) aligned within your domain-specific guidelines.

## Multi-Agent Supervisor

*Combine Agents and Tools into a multi-agent system*



Agents



MCP Servers



Genie

# Agent Bricks

Agent Bricks Beta

[Provide feedback ↗](#)

[Use PDFs in Agent Bricks](#)

Choose your use case

## Information Extraction

Extract key information, classify content, or summarize text from documents into a structured JSON.

## Knowledge Assistant

Turn your docs into an expert AI chatbot.

## Multi-Agent Supervisor

Design an AI system that brings Genie, agents, and tools together.

## Custom LLM

Specialize an LLM endpoint to perform custom text tasks (i.e. content generation, chat) aligned within your domain-specific guidelines.

## Custom LLM

*Optimized LLM for custom tasks and guidelines*



Content  
Generation



Custom  
Chat

# Use Cases

## Agent Bricks

### Information Extraction Agent

**Problem:** Data is buried

**Value:** Turn unstructured text into structured fields like names, dates, and entities—no labeled data needed.

### Custom LLM Agent

**Problem:** Teams rely on generic models or manual effort to tailor text for their domain.

**Value:** Transform text for tasks like summarization, classification, or rewriting—optimized for your industry.

### Knowledge Assistant Agent

**Problem:** Chatbots give vague or incorrect answers

**Value:** Deliver fast, accurate answers grounded in your enterprise data.

### Multi-Agent Supervisor

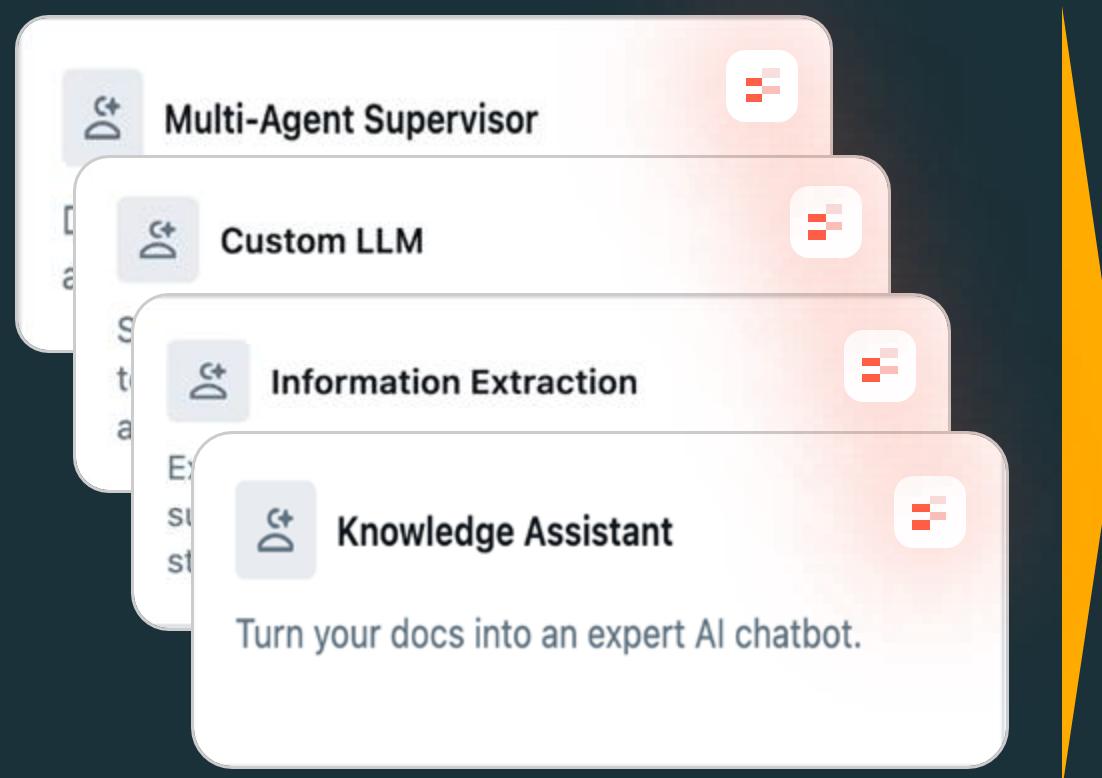
**Problem:** Complex tasks break down when a single model tries to do everything at once.

**Value:** multi-agent system with higher-quality results with less manual debugging or retraining



# Agent Bricks

## Production AI Agents Optimized on Your Data



### Existing Approach



### Agent Bricks



*Time to Production,  
up to 8x faster*

"Leveraging Agent Bricks, Analytics8 achieved a **40% increase in answer accuracy** with **8x faster implementation times** for our use cases

Post launch, we've also observed that answer quality continues to climb.

– Patrick Vinton, CTO  
Analytics8



Module 4: Production-ready Agents with Agent Bricks

## LAB EXERCISE

# Build a Knowledge Assistant with Agent Bricks



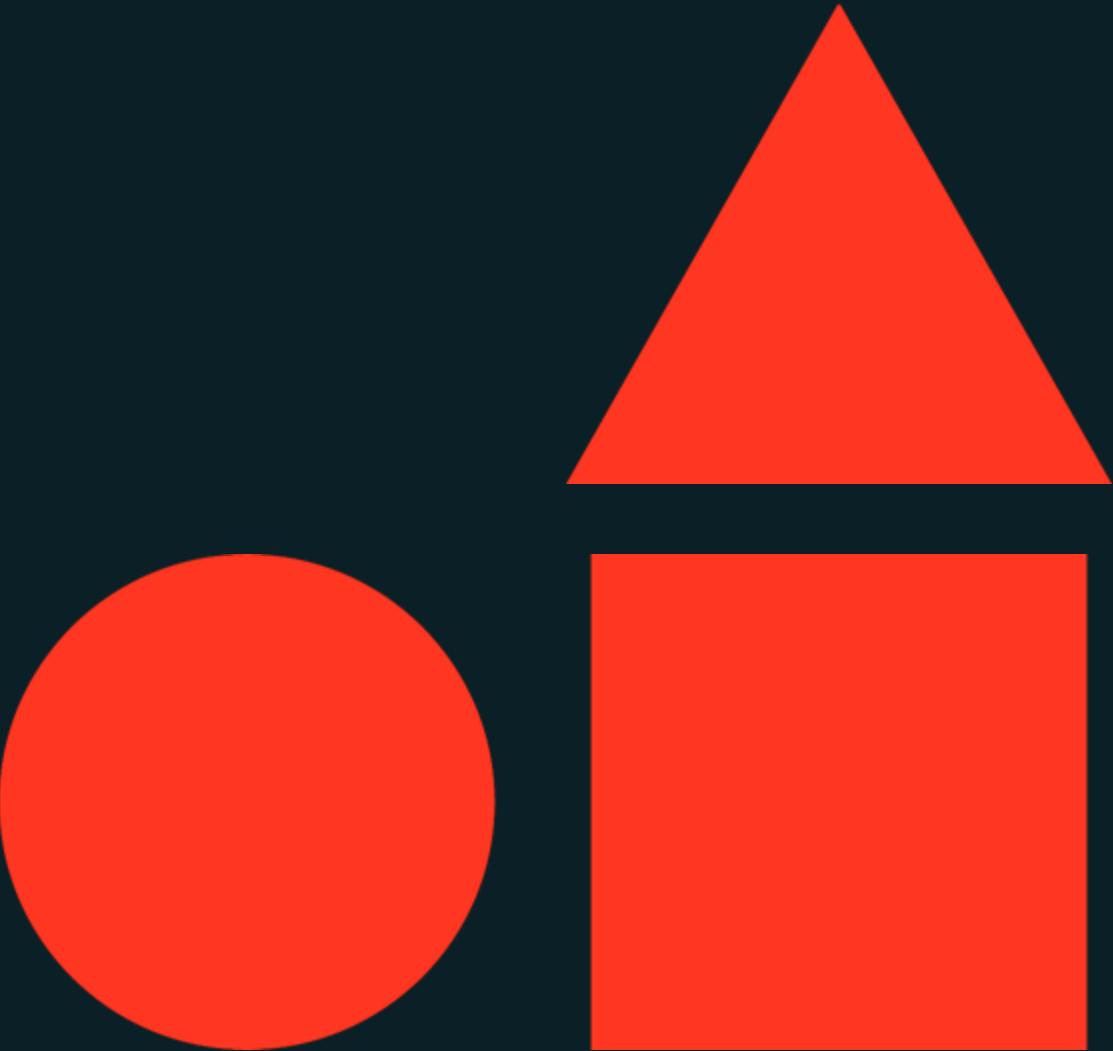
© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).



# Summary and Next Steps

---

[Get Started with AI Agents on Databricks](#)



# Next Steps

## Additional resources for continuing the learning journey

### Generative AI Engineering with Databricks

- Continue your learning through [self-paced](#) or [instructor-led](#) offerings
- These learning pathways include following courses; ([AI Agents Update - Coming Soon](#))
  - Generative AI Solution Development
  - **Generative AI Application Development**
  - Generative AI Application Evaluation and Governance
  - Generative AI Application Deployment and Monitoring

### Generative AI Engineer Associate Certification

- Validate your generative AI skills on Databricks by earning a Databricks credential
- [Exam information](#) and [exam guide](#)
- The exam covers:
  - Design Applications – 14%
  - Data Preparation – 14%
  - Application Development – 30%
  - Assembling and Deploying Apps – 22%
  - Governance – 8%
  - Evaluation and Monitoring – 12%





# databricks



© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the [Apache Software Foundation](#).