

# Retrieval to Reasoning: RAG & AI Agents on Azure Databricks



Hemamalini Nithyanandam  
Ashwini Mahendran



# AGENDA

## **Retrieval to Reasoning: RAG & AI Agents on Azure Databricks**

- 01 | Evolution – From Retrieval to Reasoning
- 02 | Context Aware RAG on Azure databricks
- 03 | Agentic RAG
- 04 | AI Agents & RAG in Azure databricks
- 05 | RAG datapipeline
- 06 | Agentic RAG Usecases - Demo
- 07 | Best Practices & Future of RAG & AI Agents

# Who Am I ....

## Professional

- Software Designer in HP
- 12+ years IT experience
- AWS Solutions Architect
- Data Engineer  
&
- AI Enthusiast



<https://www.linkedin.com/in/hemamalini-nithyanandam/>

## Personal

- Punnagai Foundation
- Certified Yoga Trainer
- Blogs



# Who Am I ....

## Professional

- Senior Software Engineer at ATMECSAI
- AI researcher & developer
- Embedded and Cloud based full stack AI development



## Personal

- Badminton Player
- Mandala Artist

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



# Evolution – From Retrieval to Reasoning

# Evolution – From Retrieval to Reasoning

## What are LLMs?

- Large Language Models (e.g., GPT-4, BERT) are deep learning models trained on vast amounts of text data.
- They generate human-like text based on input queries or prompts.

## Key Characteristics of LLMs:

- Powerful in understanding and generating text.
- Context-dependent: rely on pre-trained knowledge, limited to training data.

## Limitations:

- Static, unable to access real-time or updated information.
- Responses are limited to what the model has learned during training.

# What is Retrieval-Augmented Generation (RAG)

## **RAG: A Hybrid AI Model**

- Combines the power of LLMs with real-time information retrieval.

## **Two Components:**

- **Retrieval:** Dynamically fetches relevant data from external sources (e.g., databases, APIs).
- **Generation:** LLM refines and generates context-aware responses using the retrieved data.

# Why RAG Matters?

## **Limitations of Traditional LLMs**

- Static and limited knowledge
- Prone to hallucinations (inaccurate responses)
- Lacks real-time context awareness

## **How RAG Enhances LLMs**

- Integrates external and real-time data
- Reduces hallucinations by grounding responses in facts
- Delivers accurate, up-to-date insights
- Improves relevance and precision by incorporating context-specific knowledge



# Enhancing LLMs with Context-Aware Data

## Static LLMs vs. Dynamic Knowledge

- LLMs alone are limited by the data they were trained on, lacking the ability to adapt to real-world, real-time situations.

## Benefits of RAG:

- **Accuracy:** By retrieving relevant information from up-to-date sources, RAG ensures responses are grounded in reality.
- **Context-Awareness:** Improves understanding of context and generates more precise and relevant responses.
- **Adaptability:** Handles a wide range of topics with ease by accessing external knowledge on demand.

# Context-Aware RAG on Azure Databricks

## Key Components:

- **Retriever:** Searches knowledge bases (Vector DBs, SQL, Delta Lake)
- **Generator (LLM):** Uses retrieved data for contextual responses
- **Feedback Loop:** Ensures accuracy & relevanceState Machine

**Data Sources:** Azure Blob, Delta Lake, SQL

**Vector DBs:** Weaviate, FAISS, ChromaDB

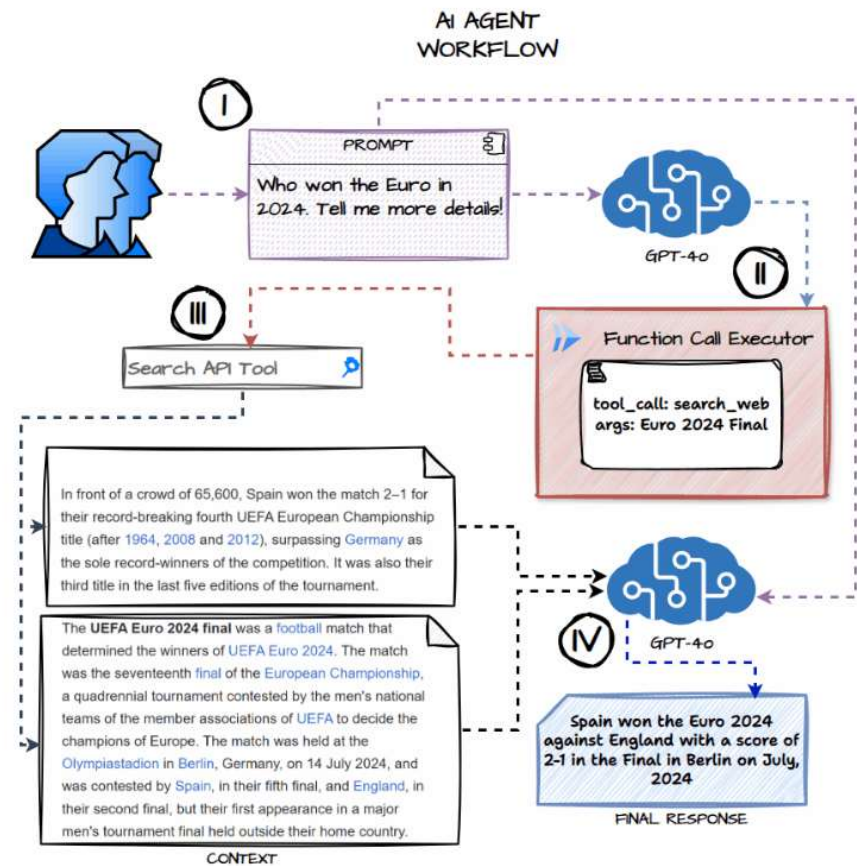
**LLMs:** OpenAI, MosaicML, Azure OpenAI

**Databricks Runtime:** Spark, MLFlow, AI Agents

AGENTIC RAG

# Agentic RAG: AI That Thinks and Acts

- **LLMs + Reasoning + Automation**  
→ More than just text generation
- **Context-aware AI agents** that retrieve, analyze, and take action
- Refine responses with contextual reasoning
- Automate decision-making workflows



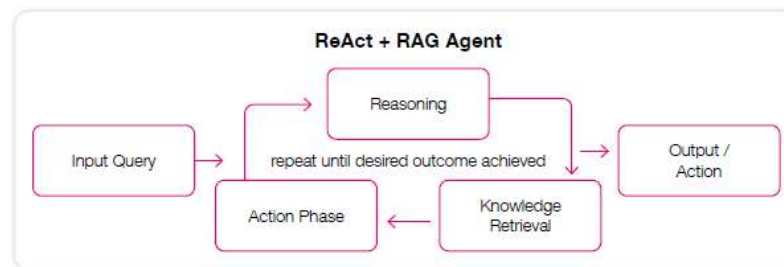
Source: [Dipanjan Sarkar](#)

# Types Of AI Agent

Name of the agent	Key Characteristics	Examples	Best For
Fixed Automation: The Digital Assembly Line	No intelligence, predictable behavior, limited scope	RPA, email autoresponders, basic scripts	Repetitive tasks, structured data, no need for adaptability
LLM-Enhanced: Smarter, but Not Einstein	Context-aware, rule-constrained, stateless	Email filters, content moderation, support ticket routing	Flexible tasks, high-volume/low-stakes, cost-sensitive scenarios
ReAct: Reasoning Meets Action	Multi-step workflows, dynamic planning, basic problem-solving	Travel planners, AI dungeon masters, project planning tools	Strategic planning, multi-stage queries, dynamic adjustments
ReAct + RAG: Grounded Intelligence	External knowledge access, low hallucinations, real-time data	Legal research tools, medical assistants, technical support	High-stakes decisions, domain-specific tasks, real-time knowledge needs
Tool-Enhanced: The Multi-Taskers	Multi-tool integration, dynamic execution, high automation	Code generation tools, data analysis bots	Complex workflows requiring multiple tools and APIs
Self-Reflecting: The Philosophers	Meta-cognition, explainability, self-improvement	Self-evaluating systems, QA agents	Tasks requiring accountability and improvement

# ReAct RAG –Reasoning + Action+ knowledge

Feature	Description
Intelligence	Employs a RAG workflow, combining LLMs with external knowledge sources (databases, APIs, documentation) for enhanced context and accuracy.
Behavior	Uses ReAct-style reasoning to break down tasks, dynamically retrieving information as needed. Grounded in real-time or domain-specific knowledge.
Scope	Designed for scenarios requiring high accuracy and relevance, minimizing hallucinations.
Best Use Cases	High-stakes decision-making, domain-specific applications, tasks with dynamic knowledge needs (e.g., real-time updates).
Examples	Legal research tools, medical assistants referencing clinical studies, technical troubleshooting agents.

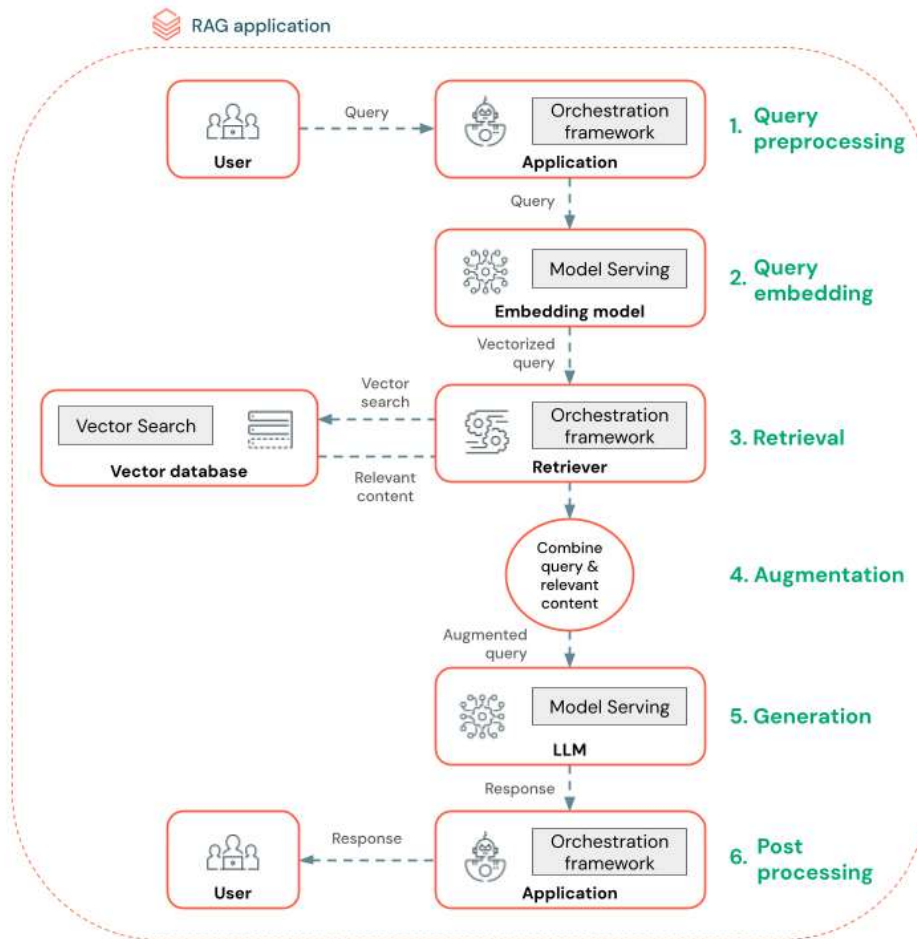


**Fig 1.5:** Workflow of a ReAct + RAG agent

Source:<https://www.galileo.ai/ebook-mastering-agents>



# AI Agents & RAG – Working Together



- 1. Query Preprocessing** – The user query is formatted, templated, or keyword-extracted for vector search.
- 2. Query Vectorization** – Model Serving converts the query into embeddings, aligning with the indexed data.
- 3. Retrieval Phase** – A vector similarity search fetches and ranks the most relevant data chunks.
- 4. Prompt Augmentation** – Retrieved chunks are merged with the query to enhance context before LLM processing.
- 5. LLM Generation** – The LLM generates a response using the enriched prompt.
- 6. Post-processing** – The output is refined with business logic, citations, or formatting adjustments.

Source: <https://www.galileo.ai/ebook-mastering-agents>

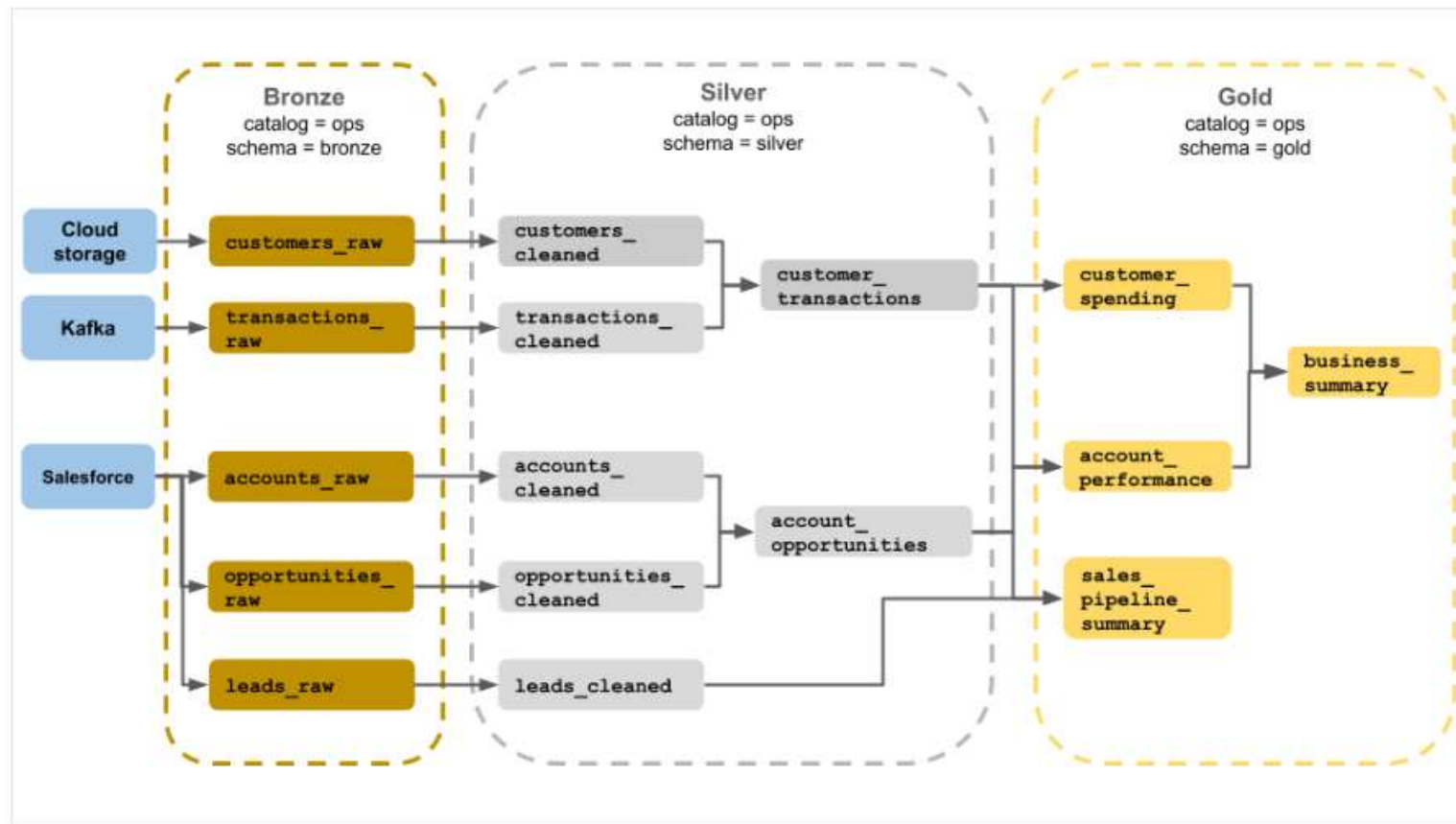
# Implementing RAG Pipelines & Agents on Azure Databricks



# AZURE DATABRICKS

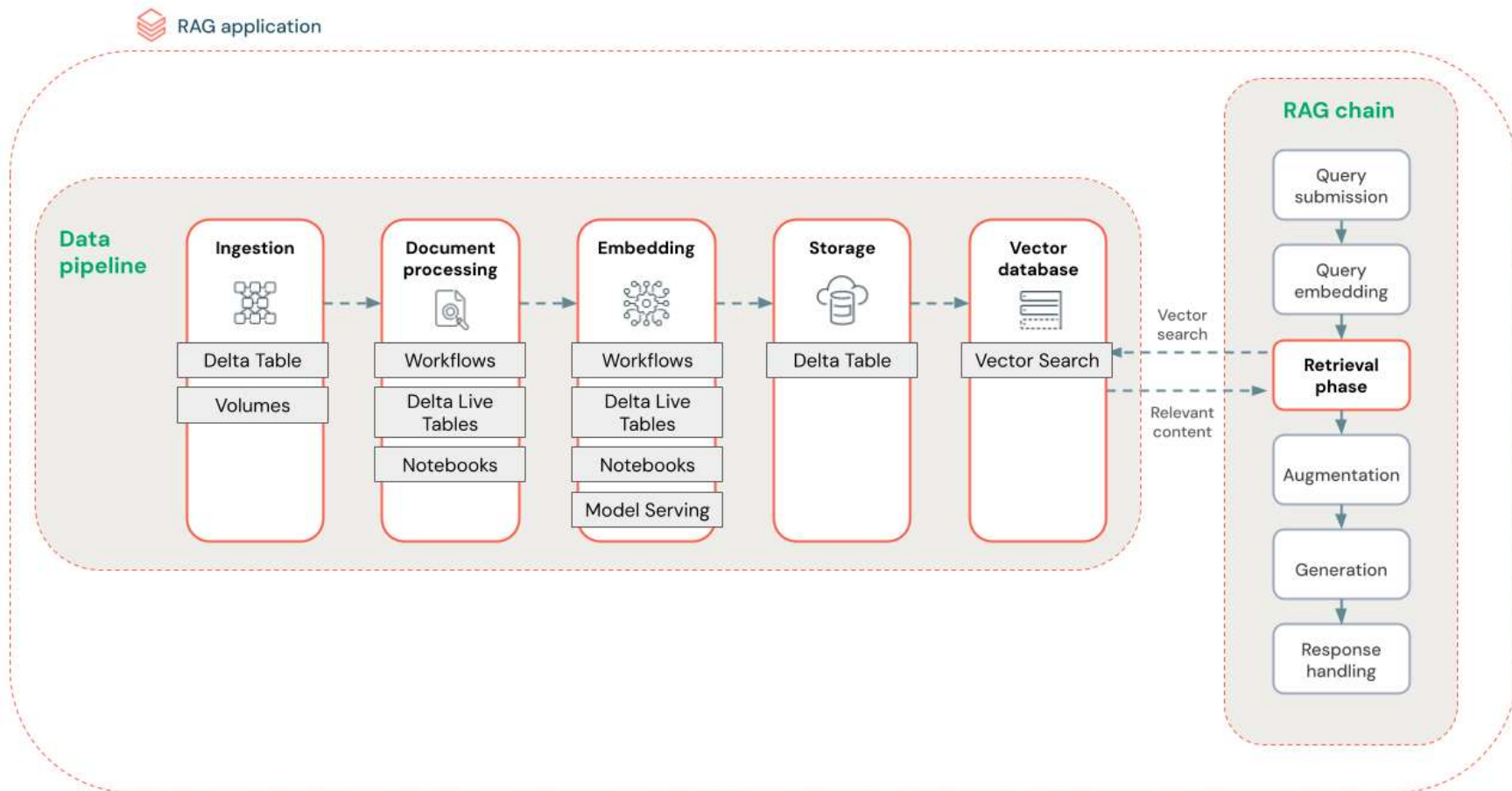
- Unified, **open analytics platform** for data, AI, and ML for data-driven decision-making
- Built on **Apache Spark**, optimized for Azure and integrated with cloud storage and security
- Uses **Data Lakehouse + AI** for optimized performance
- Supports ETL, ML, BI, & Generative AI.
- **Data Governance & Security** – Manage access and compliance effortlessly with Unity Catalog for secure data control.
- **Streaming & Real-time Analytics** – Process live data streams with Structured Streaming for instant insights and automation.

# Medallion Architecture



Source: <https://learn.microsoft.com/en-us/azure/databricks/lakehouse/medallion>

# RAG DATA Pipeline

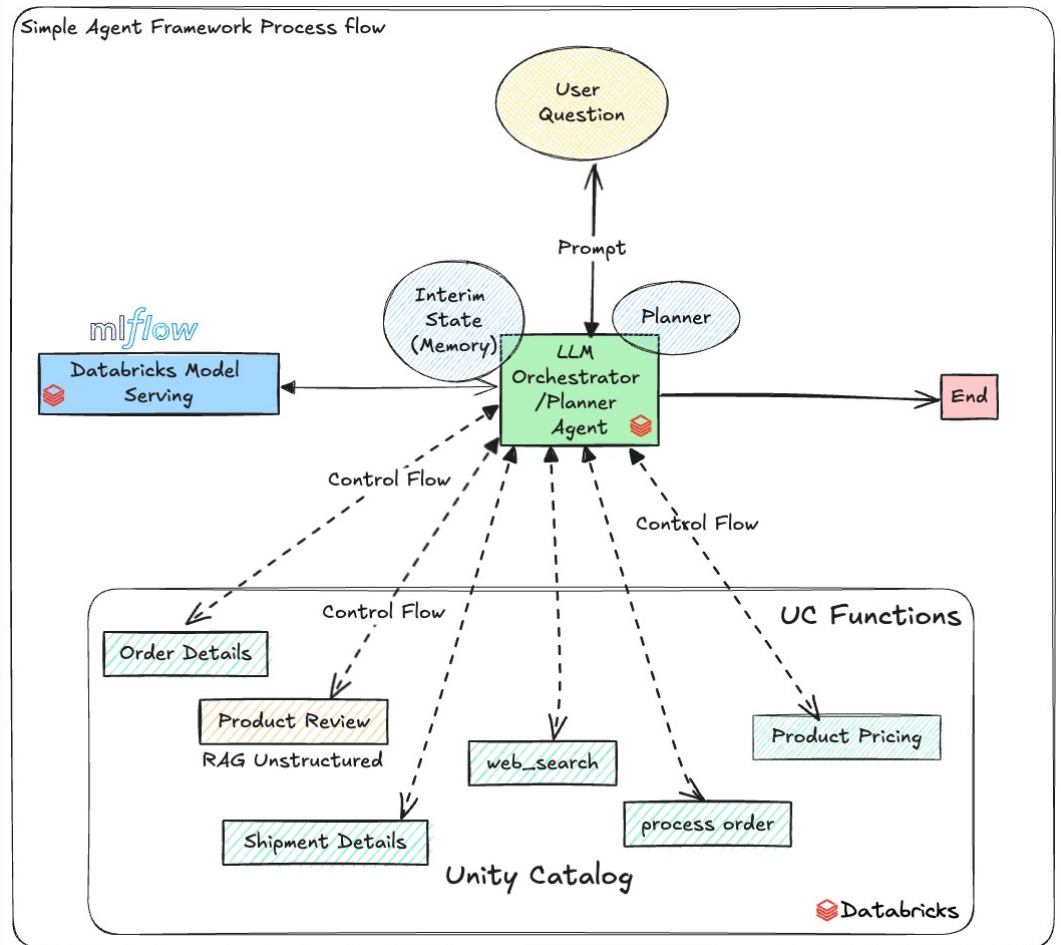


Source: <https://www.databricks.com/glossary/retrieval-augmented-generation-rag>

# Real-time Use Cases & Best Practices

# Implementing RAG in Databricks

1. Ingest & Prepare Data: Store in Delta Lake or Azure Blob
2. Generate Embeddings: Use FAISS/ChromaDB for vector indexing
3. Retrieve Relevant Context: Query vector DB for relevant documents
4. Generate Responses: Pass retrieved data to LLM
5. Agent Execution: Automate insights & decision-making

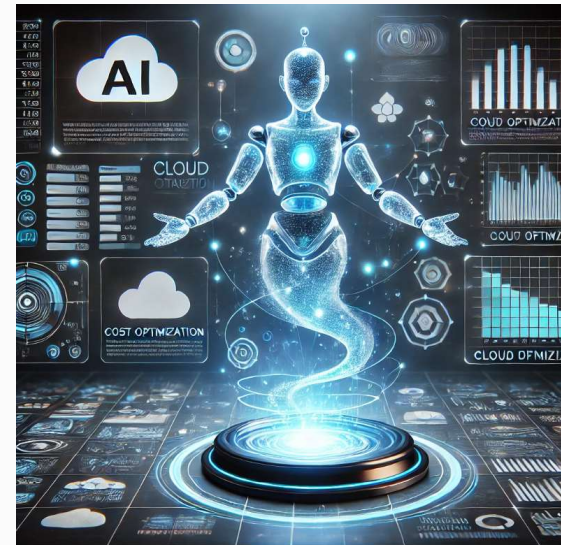


Source: <https://www.databricks.com/blog/announcing-mosaic-ai-agent-framework-and-agent-evaluation>

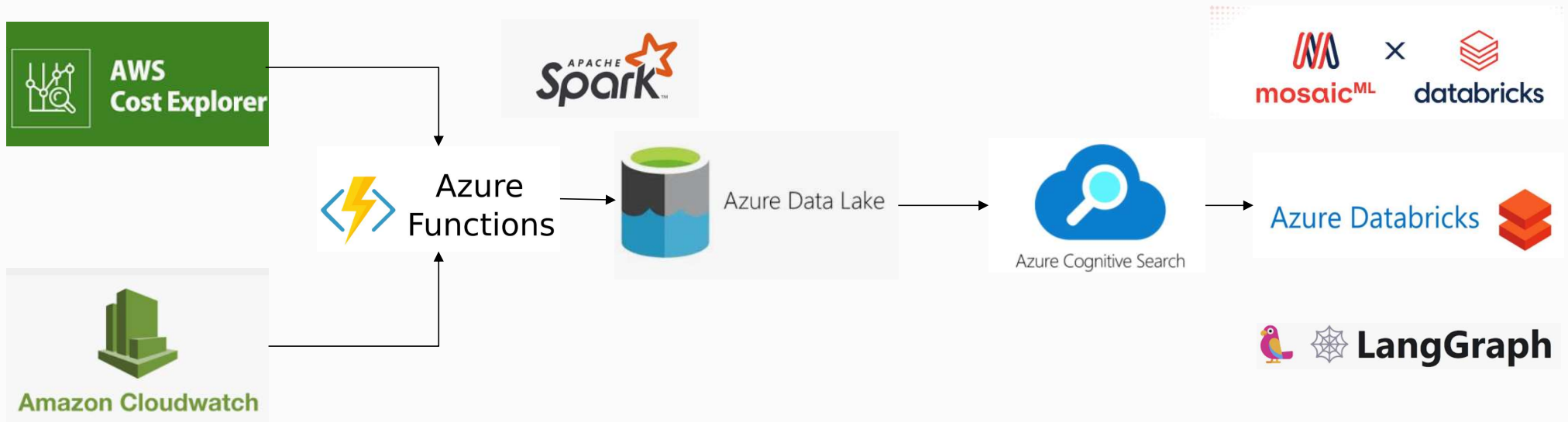
# FinOps Genie in Action

## Use Case:

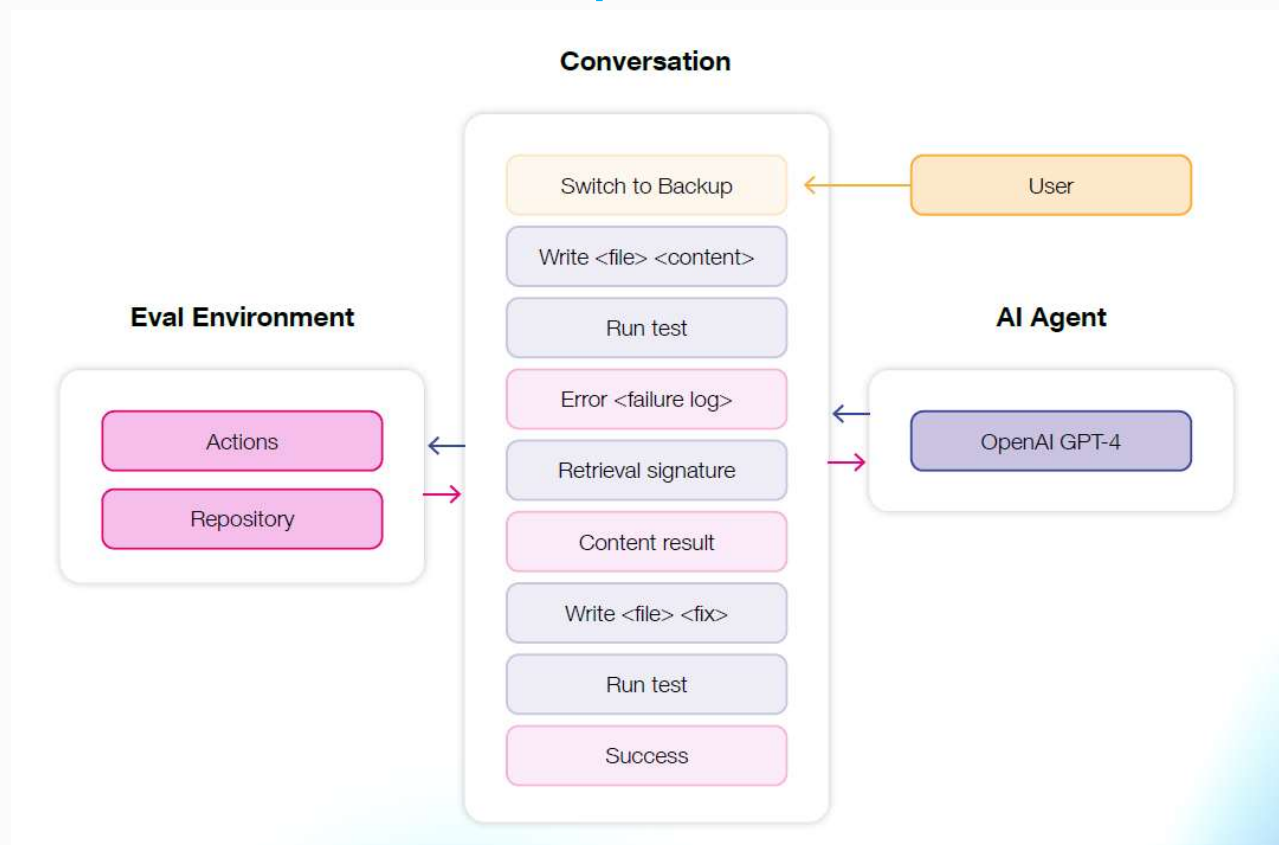
- AWS Cost Analysis & Optimization using RAG-based LLM Agent
- Ingest AWS cost data into Databricks
- Retrieve relevant insights using RAG
- AI Agent suggests cost-saving strategies



# Implementing RAG & AI Agents in Databricks



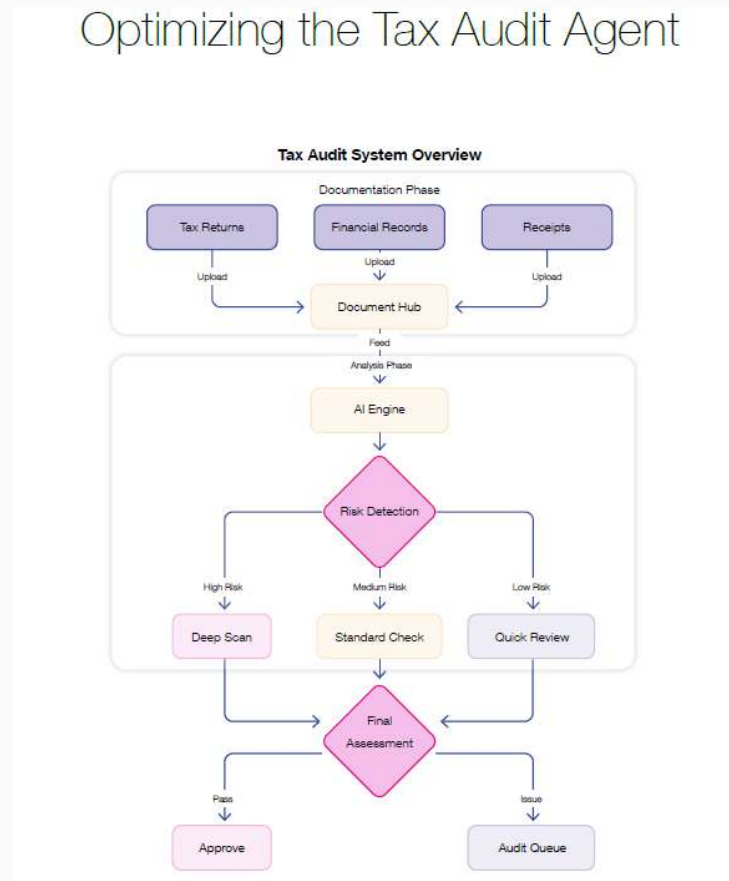
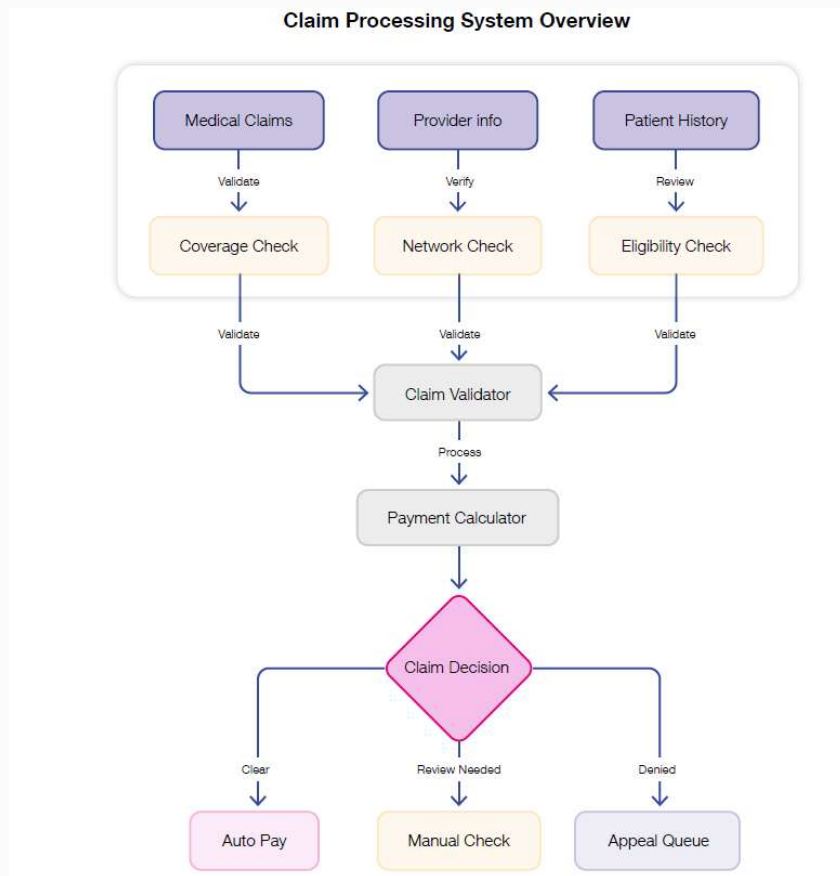
# Automated AI Agent driven Development



Source: <https://www.galileo.ai/ebook-mastering-agents>

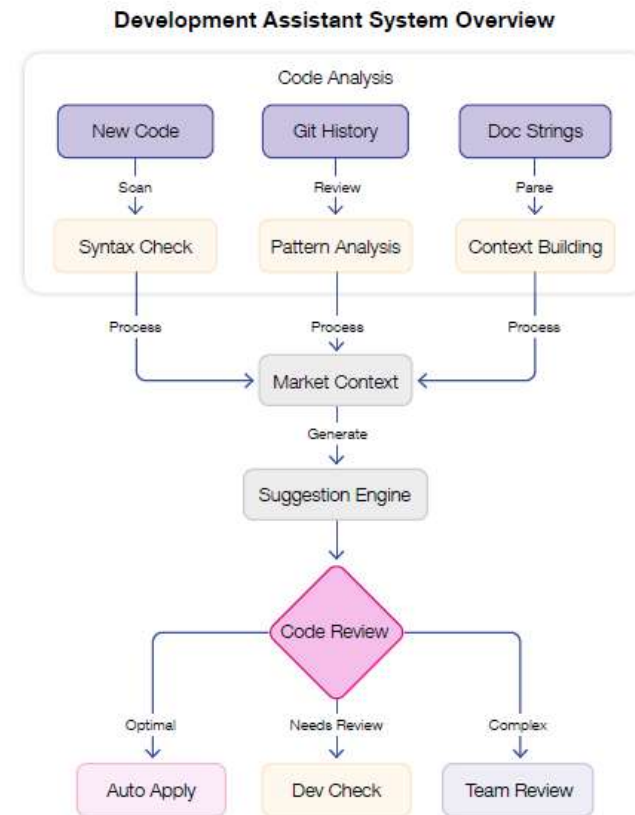
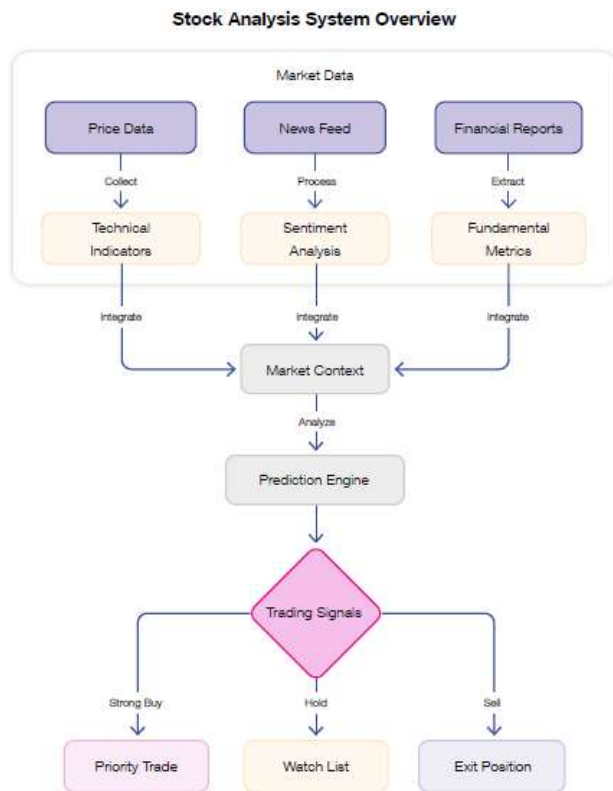


# AI Agents Usecases : Claim Processing



Source: <https://www.galileo.ai/ebook-mastering-agents>

# AI Agents Usecases :Stock Analysis






Source:<https://www.galileo.ai/ebook-mastering-agents>

# RAG & AI Agents in Enterprise

- **Financial Services:** Risk analysis & fraud detection
- **Cloud Cost Optimization:** AI-powered cost-saving recommendations
- **Customer Support AI:** Context-aware chatbot interactions
- **Healthcare AI:** Clinical data retrieval & summarization

# Databricks, Fabric & Synapse: UseCases

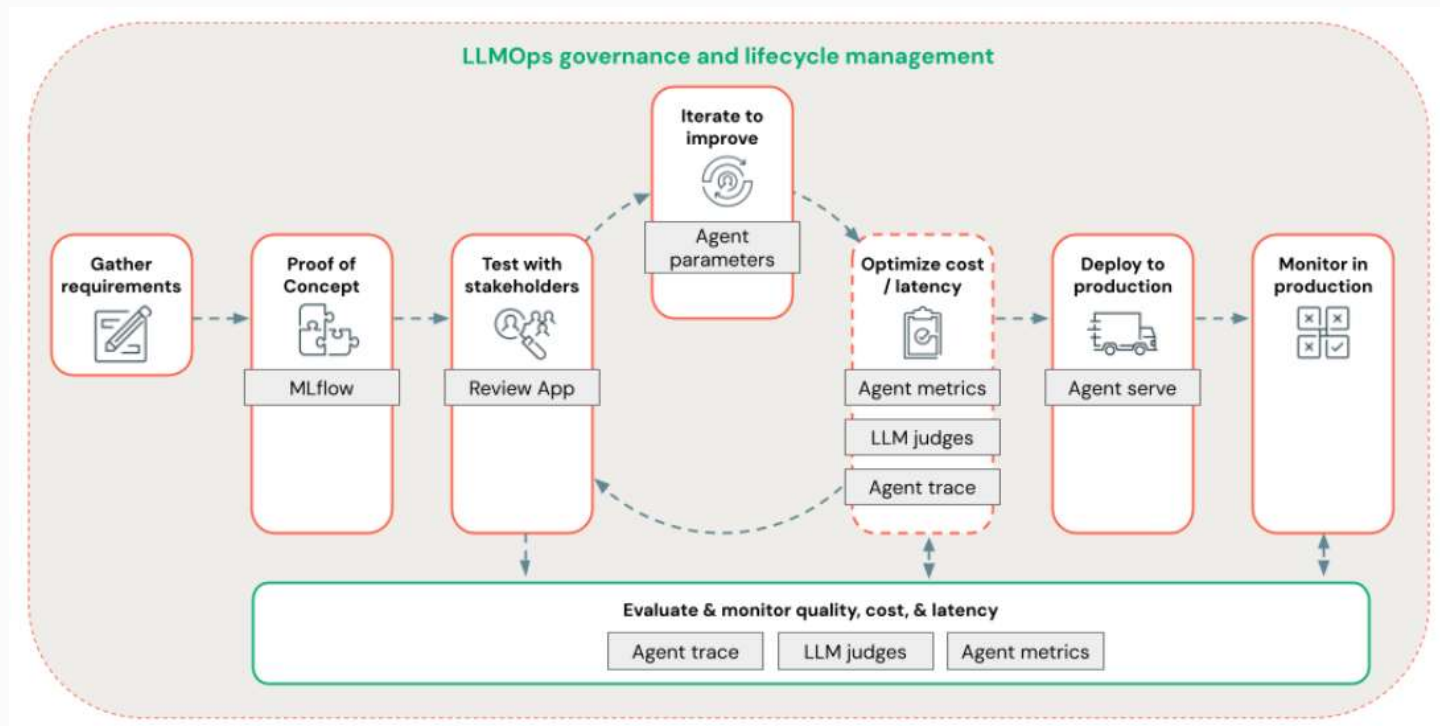
## Comparison: Databricks vs. Fabric vs. Synapse

Feature	Azure Databricks 	Microsoft Fabric 	Azure Synapse Analytics 
Core Focus	AI, ML, Data Engineering & Analytics	Unified Data & AI Platform	Data Warehousing & Big Data Analytics
Architecture	Open Lakehouse (Delta Lake)	SaaS-based Lakehouse	SQL Data Warehouse & Data Lake
Best For	Big Data, ML, AI, Real-time Streaming	End-to-end Data & AI (BI, AI, Governance)	BI, SQL Analytics, ETL
Compute Engine	Apache Spark, Photon	Power BI, Spark	SQL Pools, Apache Spark
Data Processing	Batch, Streaming, ML, AI	Low-code/no-code, AI-powered automation	SQL-based ETL, Data Pipelines
Storage	Delta Lake (open format)	OneLake (Fabric's data lake)	Azure Data Lake Storage (ADLS)
Governance	Unity Catalog (fine-grained access)	Microsoft Purview	Role-based Access Control (RBAC)
BI & Reporting	Connects to Power BI	Deep Power BI integration	Power BI & SQL Reporting
Use Cases	AI/ML, Real-time Analytics, Data Science	Business Intelligence, AI Automation, Governance	Data Warehousing, Structured Data Analytics

# Best Practices for Scalable RAG & AI Agents

- **Optimize Retrieval Efficiency:** Hybrid search (semantic + keyword)
- **Fine-tune models** for domain-specific knowledge using Azure OpenAI, MosaicML, or Databricks Foundation Models.
- **Implement Cohere Reranking** for improved relevance in retrieval results.
- **Optimize embedding model selection** (e.g., OpenAI's ada-002, Cohere embeddings).
- **Reduce Hallucinations:** Reinforcement learning for feedback loops
- **Scalability Considerations:** Deploy RAG with Databricks & Azure AI
- Implement **AI Agent Evaluation Framework** to measure response accuracy and quality.

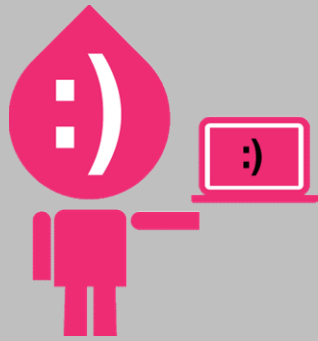
# AI Agent Evaluation



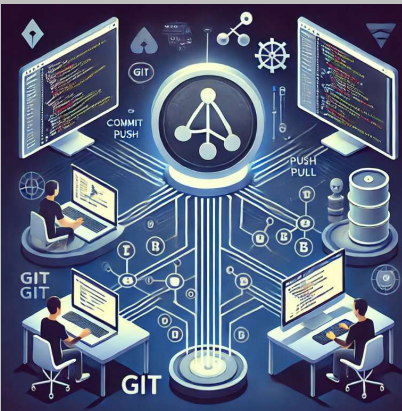
Source: <https://docs.databricks.com/aws/en/generative-ai/agent-evaluation/>

# Future of RAG & AI Agents

- **Memory-augmented Agents** (Retain past interactions)
- **Autonomous AI Decision-Making** (Self-improving models)
- **Advanced Multi-Agent Systems** (Collaboration between AI agents)



# ANY QUESTIONS...



<https://github.com/hemsush/GlobalAIBootcamp2025Talk>

