

LangChain for Continuous Integration

Instructor

Kartik Nighania

MLOps Engineer at Typewise

Ex-Head of Engineering - YC-backed startup

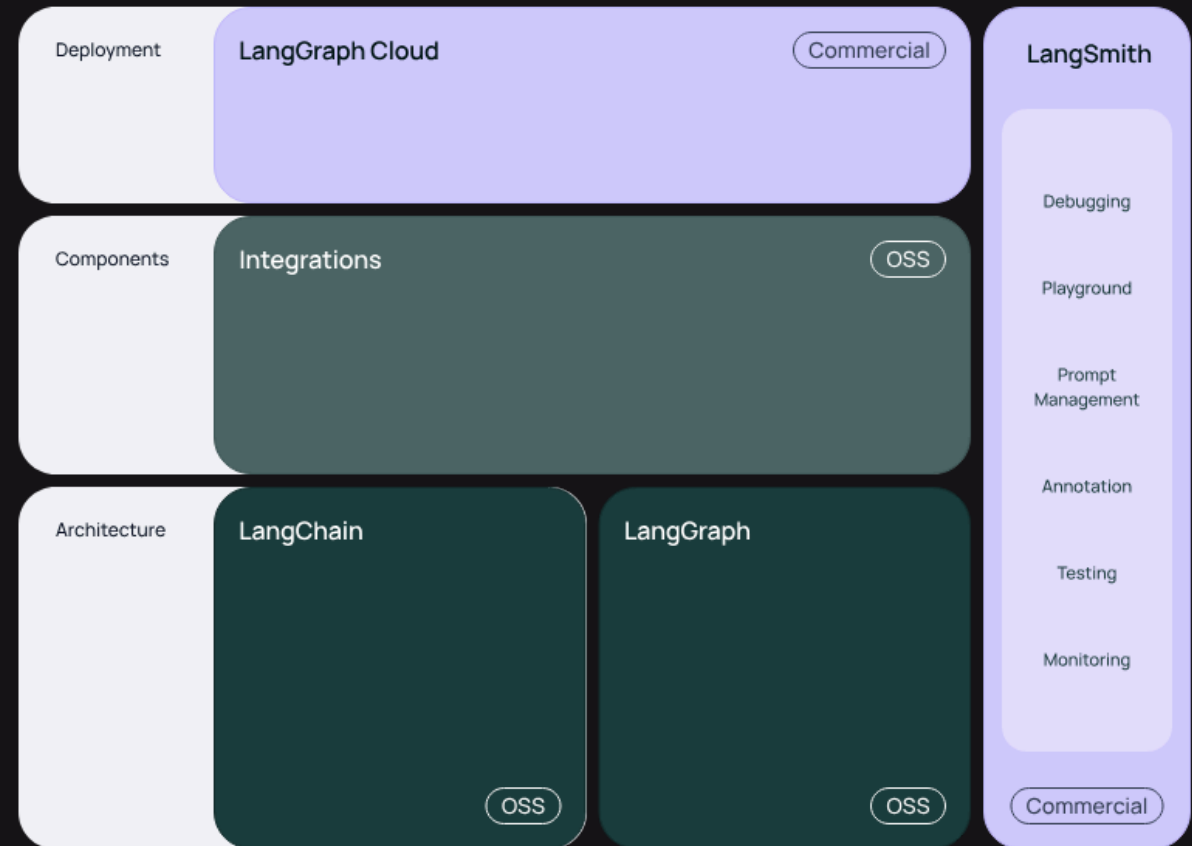
Certified AWS Cloud and Kubernetes Engineer

Open source contributor



LangChain for Continuous Integration (CI)

- **LangChain:**
 - Chains, agents, and retrieval strategies
- **Open-source libraries:**
 - core: Base abstractions and LCEL
 - community: e.g. @LangChain/openai
- **LangGraph.js:**
 - multi-actor applications
- **LangSmith:**
 - A developer platform that lets you debug, test, evaluate, and monitor LLM applications.
- **LangGraph Cloud:**
 - Deploy LangGraph agents
- **LangServe:**
 - Chains as REST APIs
- **OSS vs Commercial**



LangChain Components



LLMs

Raw foundation models

Chat Models

Fine-tuned for chat instructions

Agents

Self guiding LLMs



Document Loaders

Data ingestion

Chat Loaders

Data ingestion in chat format

Document Transformers

Convert raw data

Text Embedding Models

Embeddings for search



Memory

Chat messages

Graphs

Graph databases

Vector Stores

Database for search

Model Cache

Output caching

Stores

Storage solutions



Retrievers

All data level transformations

Tools and Toolkit

Functionalities for LLMs and Agents



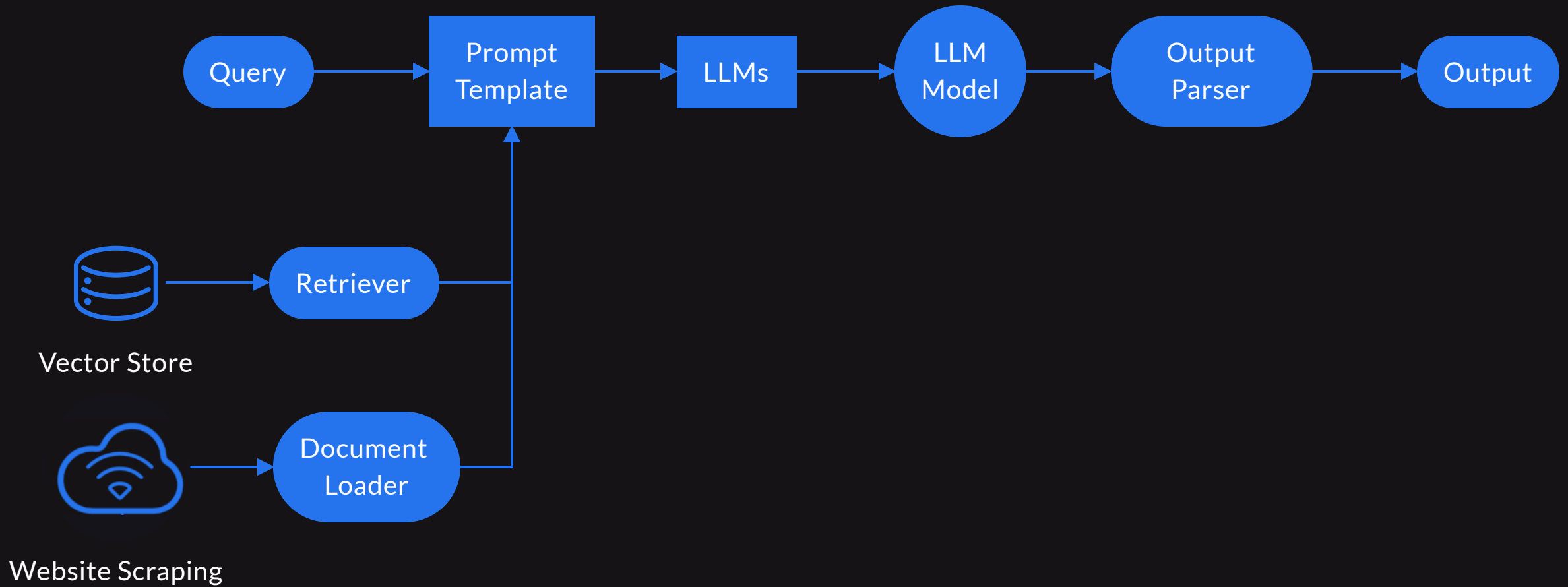
Callbacks

Deeper integration

Adapters

Copy interface

RAG pipeline with LangChain



LCEL

- LCEL stands for **LangChain Expression Language**
- Orchestration framework that is declarative and composable in nature



```
chain = prompt | llm | parser
chain.invoke({
    "name": "Bob",
    "user_input": "What is your name?"
})
```

Advantages of LCEL

- Streaming support
- Async and multi chain parallel execution support
- Retries and fallback support
- Intermediate steps and callback support
- Streaming for different output types with partial schema
- LangSmith for monitoring
- LangServe for deployment

Thank You
