

# Introduction to LangChain

## Instructor

Dipanjan Sarkar

Head of Community & Principal AI Scientist at Analytics Vidhya

Google Developer Expert - ML & Cloud Champion Innovator

Published Author



# Outline

- What is LangChain
- The LangChain Core Framework
- LLM Input / Output with LangChain
- Retrieval with LangChain
- Tools and Agents with LangChain
- Building Chains with LangChain
- Other features

# What is LangChain

- An open framework & ecosystem to develop applications powered by LLMs using a suite of tools



**LangChain**

# What is LangChain

- Its core framework consists of libraries to connect with LLMs using prompts and format outputs using output parsers.
- Multiple steps can be chained and complex conversational applications, RAG systems, and agents can be built.



**LangChain**

# What is LangChain

- LangChain features LangServe and LangSmith for deploying, evaluating, and monitoring LLMs.
- LangChain includes LangGraph, which allows for combining multiple chains to build complex agents and more.

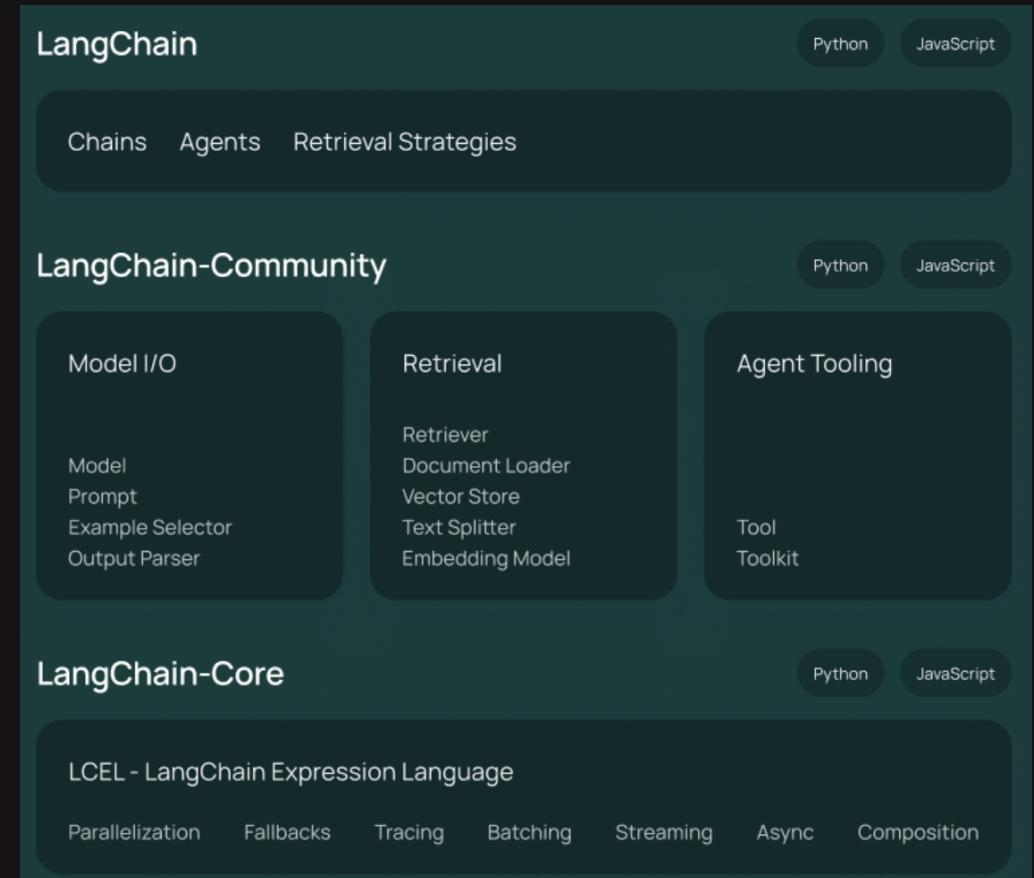


**LangChain**

# LangChain Core Framework

It is a framework that gives developers and data scientists a suite of components to build LLM applications easily.

- As the name suggests, one can easily chain multiple operations together to build simple as well as complex conversational LLM applications.
- Compatible with almost all popular large language models.



# LLM Input/Output with LangChain

langchain-openai

langchain-google-genai

langchain-aws

langchain-google-vertexai

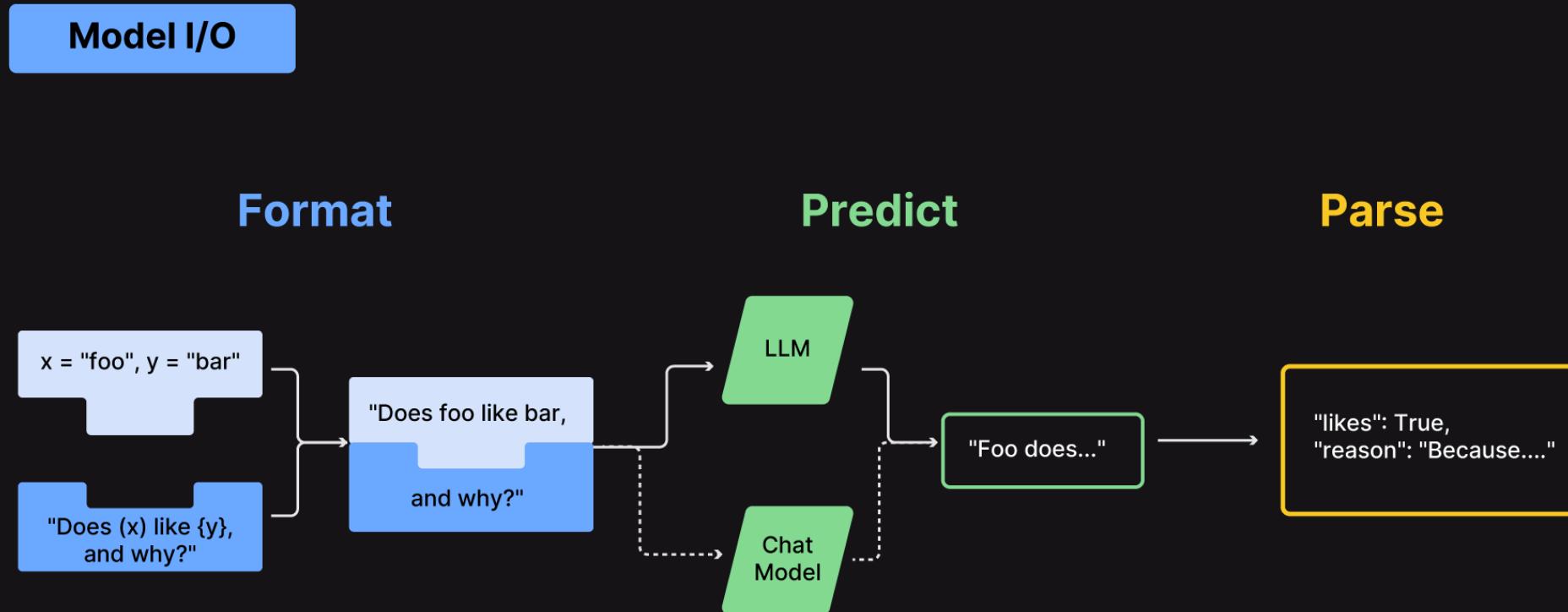
langchain-anthropic

langchain-community

LangChain has the necessary libraries and functions to connect with all major closed LLMs like ChatGPT, Gemini, Claude and popular models from cloud providers.

LangChain has a community library which is a one-stop shop to connect with all open LLMs including popular models hosted on HuggingFace, Vector DBs and more

# LLM Input/Output with LangChain



# LLM Input/Output with LangChain

- LangChain provides interfaces to talk to LLMs using its popular `ChatModel` construct and `predict` responses based on input prompts
- Constructs like `PromptTemplate` and `ChatPromptTemplate` allow for dynamic, data-driven prompt `formatting` and creation at runtime.

# LLM Input/Output with LangChain

- Allows access to output parsers to **parse** LLM responses into formats like JSON, CSV, and more
- Enables us to do iterative prompt engineering easily

# Why Retrieval with LangChain

- Many enterprise applications use LLMs to access **custom enterprise data** for answering questions beyond their original training data.

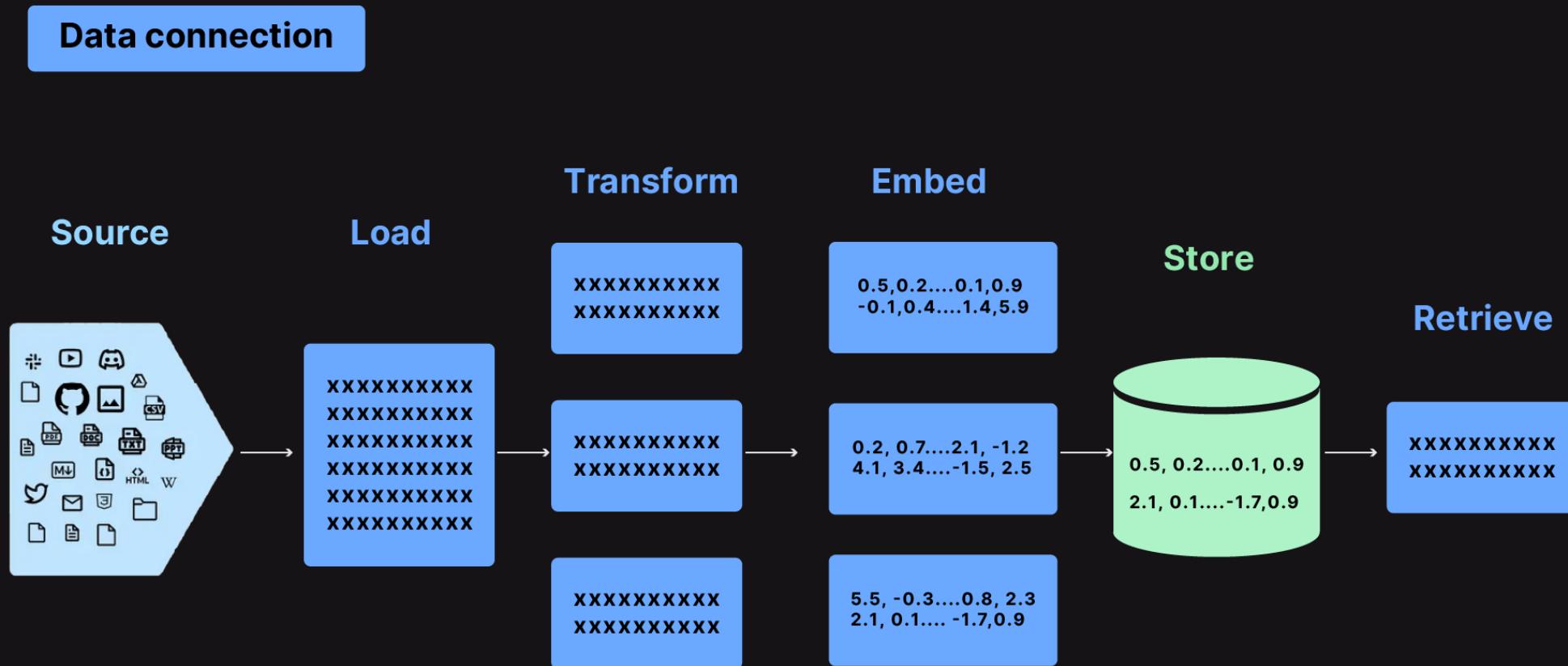
# Retrieval Augmented Generation (RAG) System

A system which integrates custom data and augments an LLM's knowledge-base to generate contextual responses

# Why Retrieval with LangChain

- LangChain helps in building both the retrieval and the response generation pipeline

# Retrieval with LangChain



# Retrieval with LangChain

- LangChain provides over 100 different document loaders:
  - as well as integrations with other major providers like Unstructured [to load data](#) from all types of documents (HTML, PDF, code)
- Provides several transformation algorithms:
  - [to split, chunk and transform](#) larger documents into more manageable chunks for retrieval

# Retrieval with LangChain

- Offers integrations with more than 25 embedding model providers:
  - ranging from open-source solutions like HuggingFace to proprietary ones like OpenAI
  - enabling the embedding of document chunks into vector embeddings.



HuggingFace

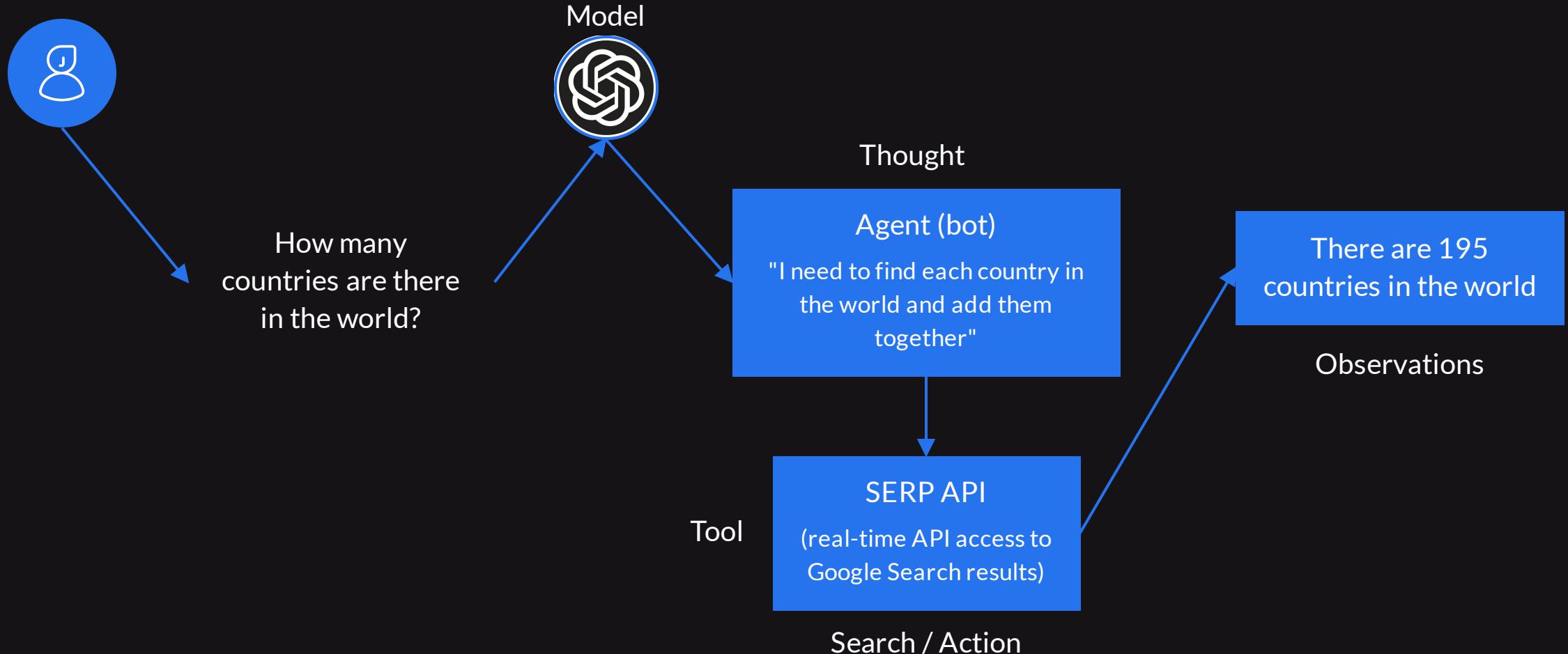


OpenAI

# Retrieval with LangChain

- Integrates with over 50 vector database providers:
  - to store document chunks and embeddings.
- Supports many different retrieval and search algorithms:
  - to retrieve relevant document chunks based on user queries

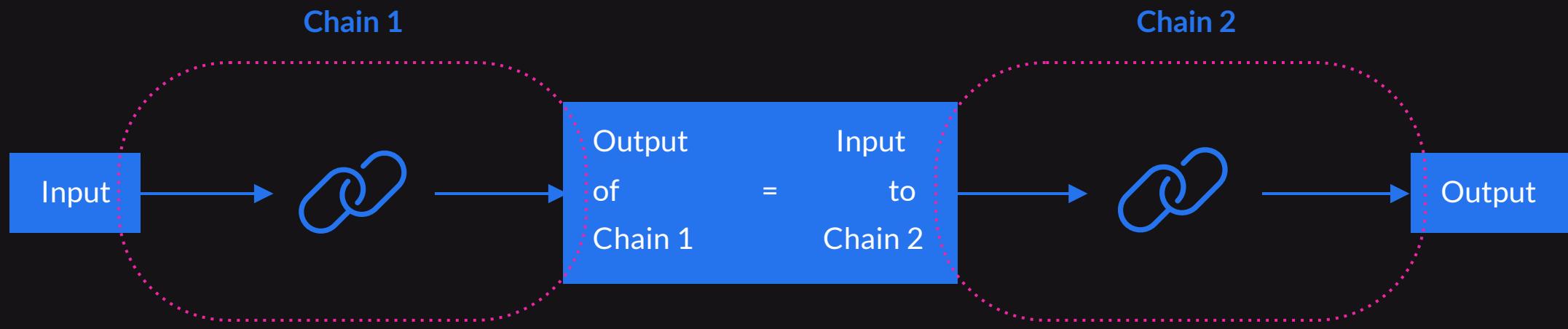
# Tools and Agents with LangChain



# Tools and Agents with LangChain

- LangChain enables to build tools that give an interface for LLMs and other components to interact with other systems
  - Example tools include Wikipedia, calculator, search engines, and more
- Allows for the creation of agents that use LLMs to determine actions, which are executed using various tools.
  - An executor serves as the runtime environment that facilitates the execution of these agents.

# Building Chains with LangChain



# Building Chains with LangChain

- LangChain allows the use of chains, which are sequences of calls or execution steps.
  - These calls can be made to an LLM, a tool, or a data preprocessing step.

# Building Chains with LangChain

LangChain supports two major types of chains:

---

- **Legacy Chains**

- These are derived classes constructed by subclassing a legacy Chain class
- These chains do not use LCEL under the hood but are the standalone classes

- **LCEL Chains**

- It uses the newer LangChain Expression Language (LCEL) to build a chain of steps by using the overloaded vertical bar or pipe ‘|’ operator

## Other features

- Enables to connect to almost any third-party LLM, Vector DB, Embedding models, and Data loaders
- Supports storing LLM conversation history using memory constructs, either in-memory, on disk, or in a data store.
  - It integrates with all popular data stores for effective memory management.
- Has capabilities for callbacks which are useful for logging, monitoring, streaming, and other tasks

# Thank You

---