Demonstration

Agenda

- Understand what kind of GenAI applications we will learn to create in this course
- Demonstrate end-result applications and use cases
- No need to code

Setup

Checklist

Install Python

- https://www.python.org/downloads/

Install Jupyter Notebook

- pip install notebook
- pip install jupyterlab

Install Langchain libraries

- pip install langchain
- pip install langchain-core
- pip install langchain_openai
- pip install langchain_community

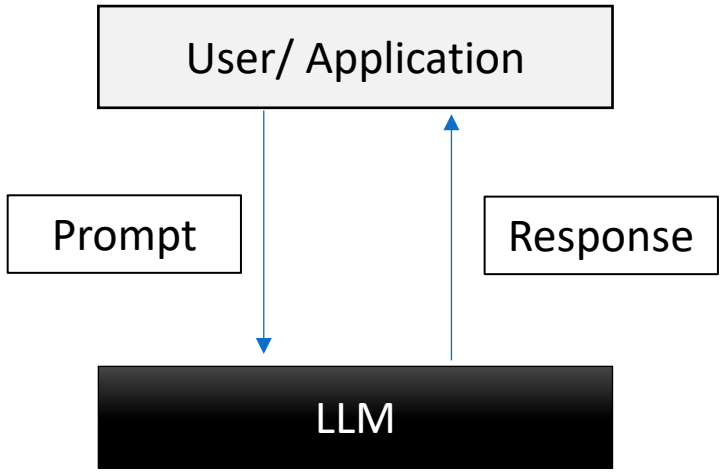
Create llm-model API keys

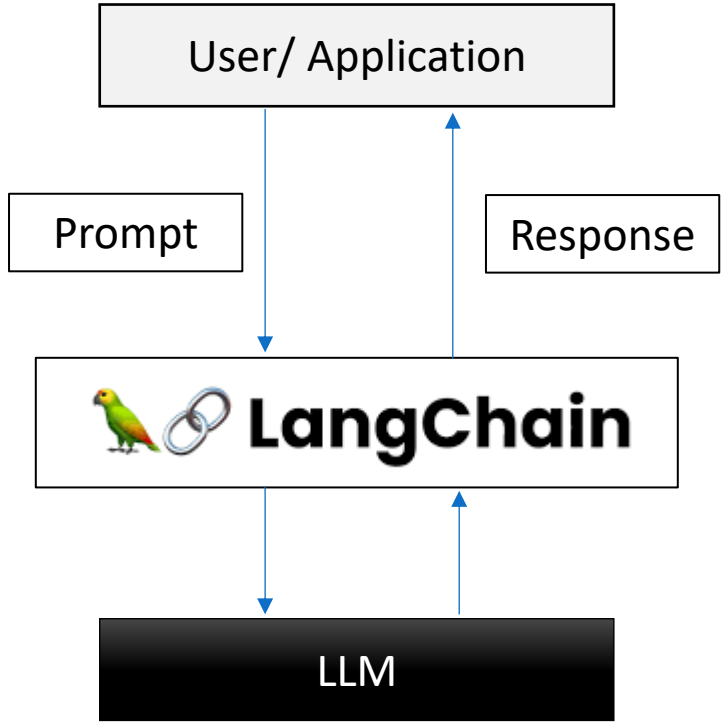
- OpenAI
- Google Generative AI

LangChain

For a simple prompt-response setup, langchain is not much useful

Why?

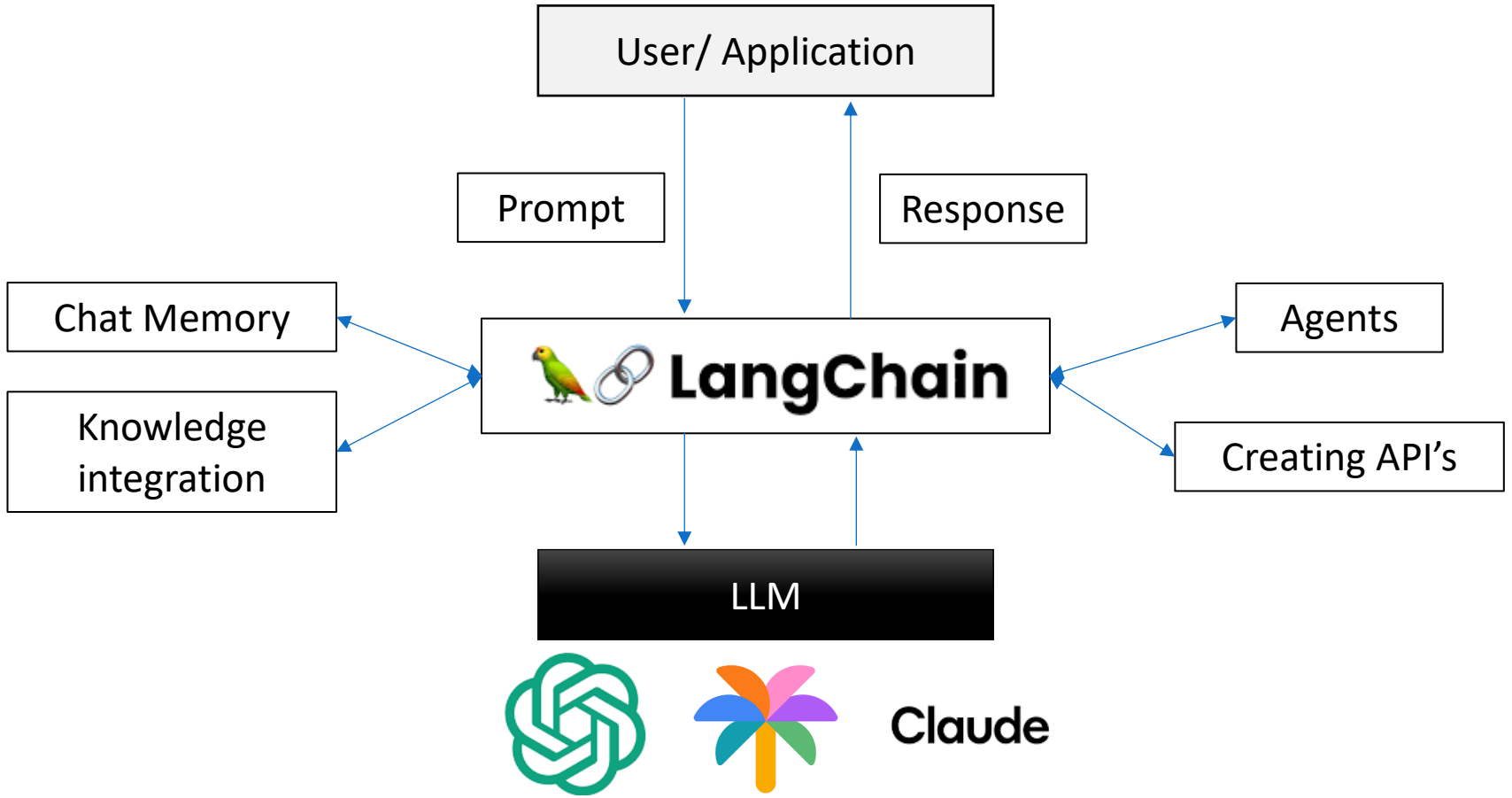




LangChain

For advanced applications, langchain simplifies the coding

Why?



LangChain

What & Why?

What is LangChain?
LangChain is a framework designed to simplify the creation of applications using LLMs.

Why use LangChain?
It makes it easy to add and replace components that bring advanced capabilities like agency, knowledge integration, memory etc.

LLM vs Chat input

Definition

LLMs are the older models which take in a plain text string as a prompt and give back a response.
Chat is based on newer chat models which take in three different type of messages and is richer in context

Types of chat messages

Human message – used to input user's prompt
AI message – used to provide AI's previous responses
System message – used to set the context

Example:
System – You are a helpful AI that helps the user make travel plans. Respond only in a single line.
Human – I want to go skiing. Which city should I go to?
AI – You should consider visiting Aspen, Colorado for a great skiing experience.
Human – What else can I do in that city?

ChatOpenAI Model

Important parameters

model - Name of OpenAI model to use.
Refer - https://platform.openai.com/docs/models

temperature – controls the randomness/ creativity. Between 0 and 1.
0 – less creative, 1 – more creative

max_tokens - Max no. of tokens to generate. Used to control the cost of response.

timeout - Timeout for requests**.**

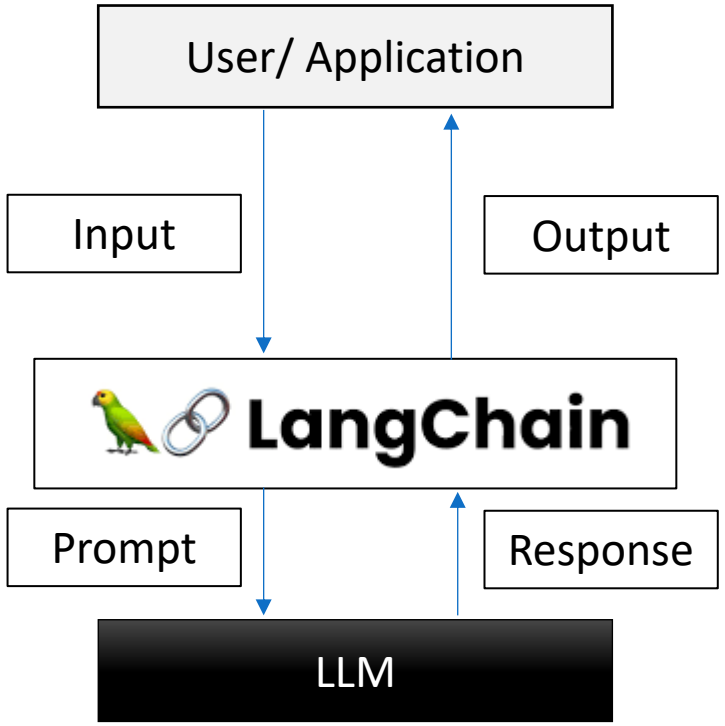
max_retries - Max number of retries.

api_key - OpenAI API key. If not passed in will be read from env var OPENAI_API_KEY.

References - https://api.python.langchain.com/en/latest/chat_models/langchain_openai.chat_models.base.ChatOpenAI.html
https://python.langchain.com/v0.2/docs/integrations/llms/openai/

Prompt Template

What?

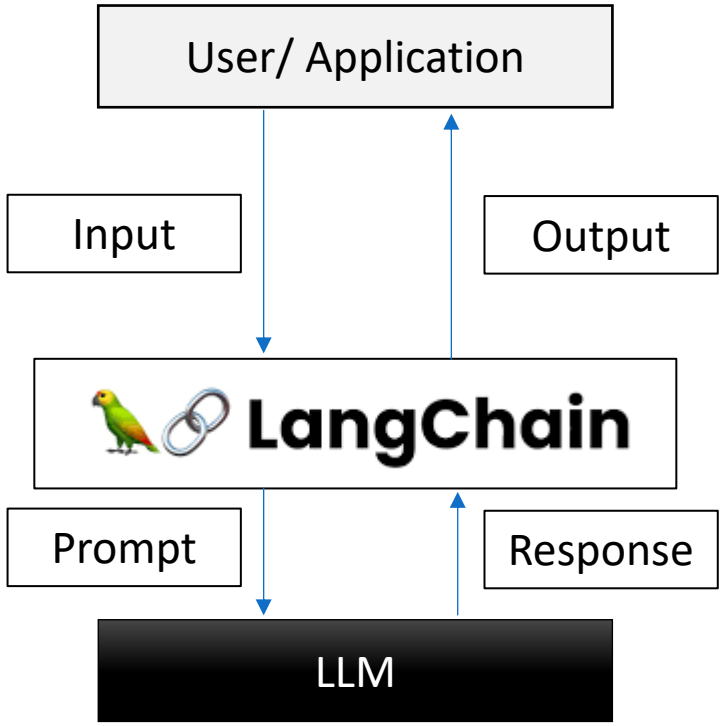


Prompt
Design a logo for a modern tech startup. The logo should feature a minimalist design with clean lines and a geometric shape. Use a color scheme of blue and white.

Prompt Template
Design a logo for a [type of company]. The logo should feature a minimalist design with clean lines and a geometric shape. Use a color scheme of [Color scheme].

Prompt Template

Why?

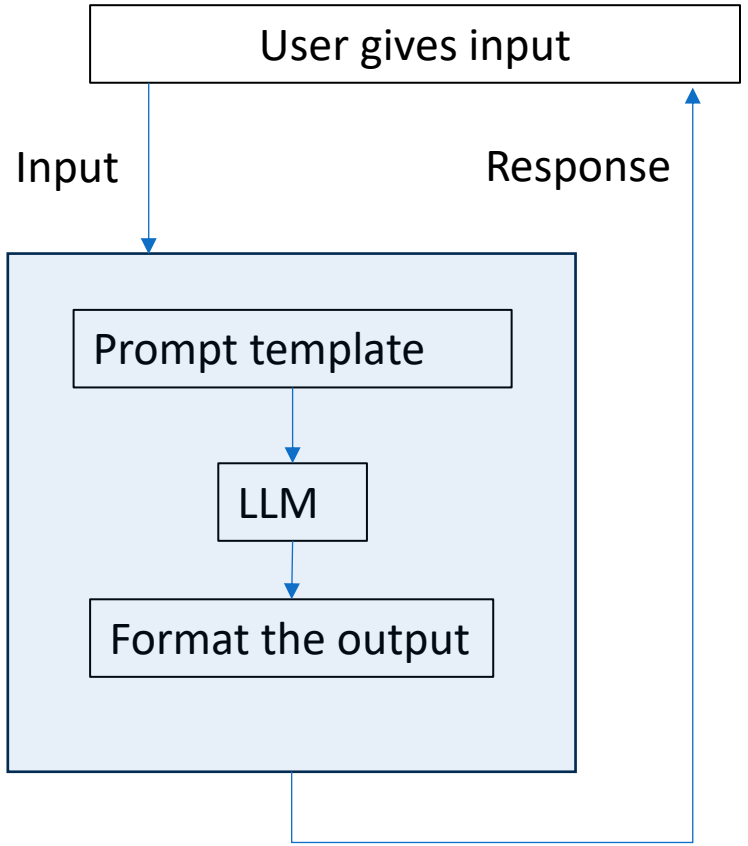


What
Prompt templates convert user input to a useful Prompt

Why
Saves time and effort
Enrich input by adding instructions, examples and questions

Chains

What?



Why

- Specify and control the sequence of actions
- Input processing and output formatting
- Makes the code modular and clearer to understand and modify
- Add memory

Chain

What will be covered

Conventional Chains

- Generic chain - LLM Chain
- Specific chain - LLM Math Chain
- Sequential Chains

Easier Prompt and Output formatting

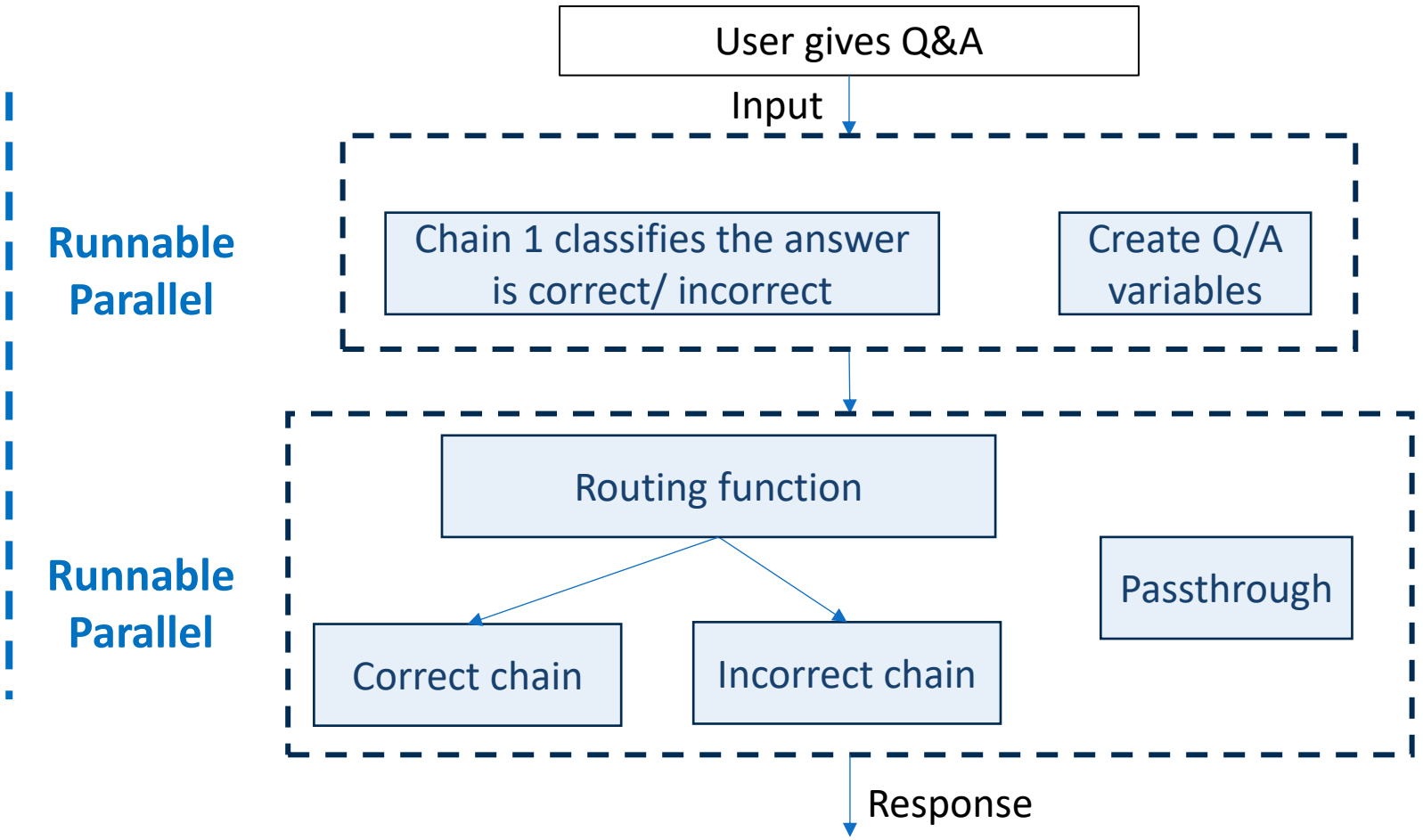
LangChain Expressions Language (LCEL) based chaining

- Pipe Operator |
- Runnable Primitives
- Controlling flow of execution
- Dynamic Routing

Easier and more feature rich in controlling the sequence of action

Chains

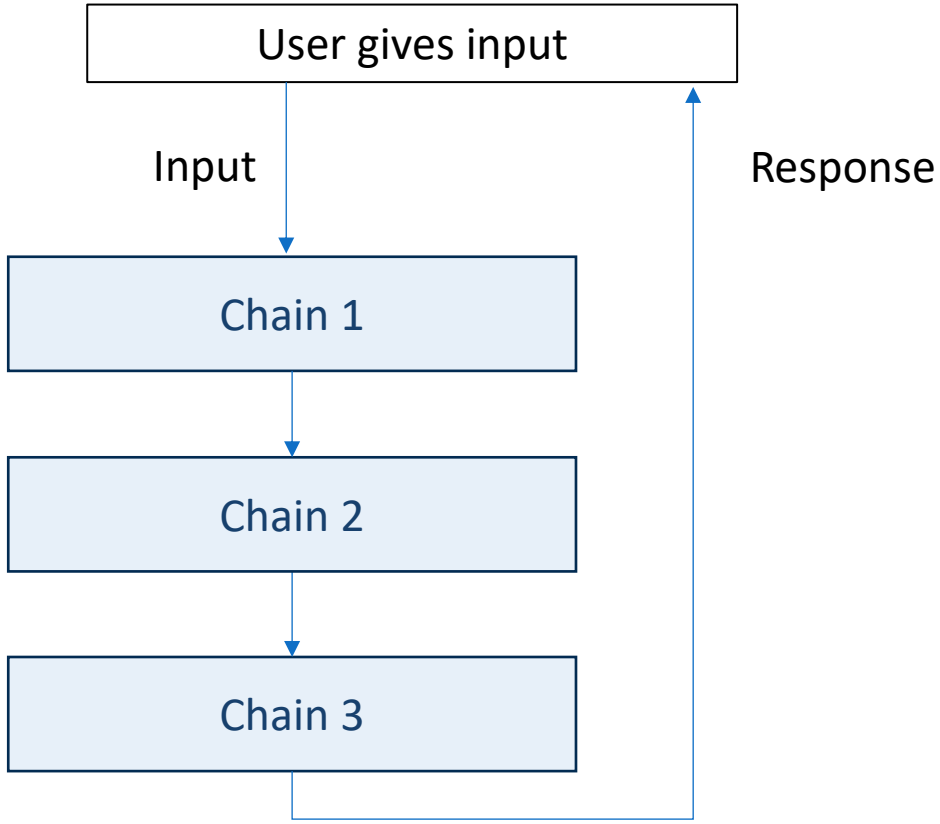
Dynamic Routing



Chains

What?

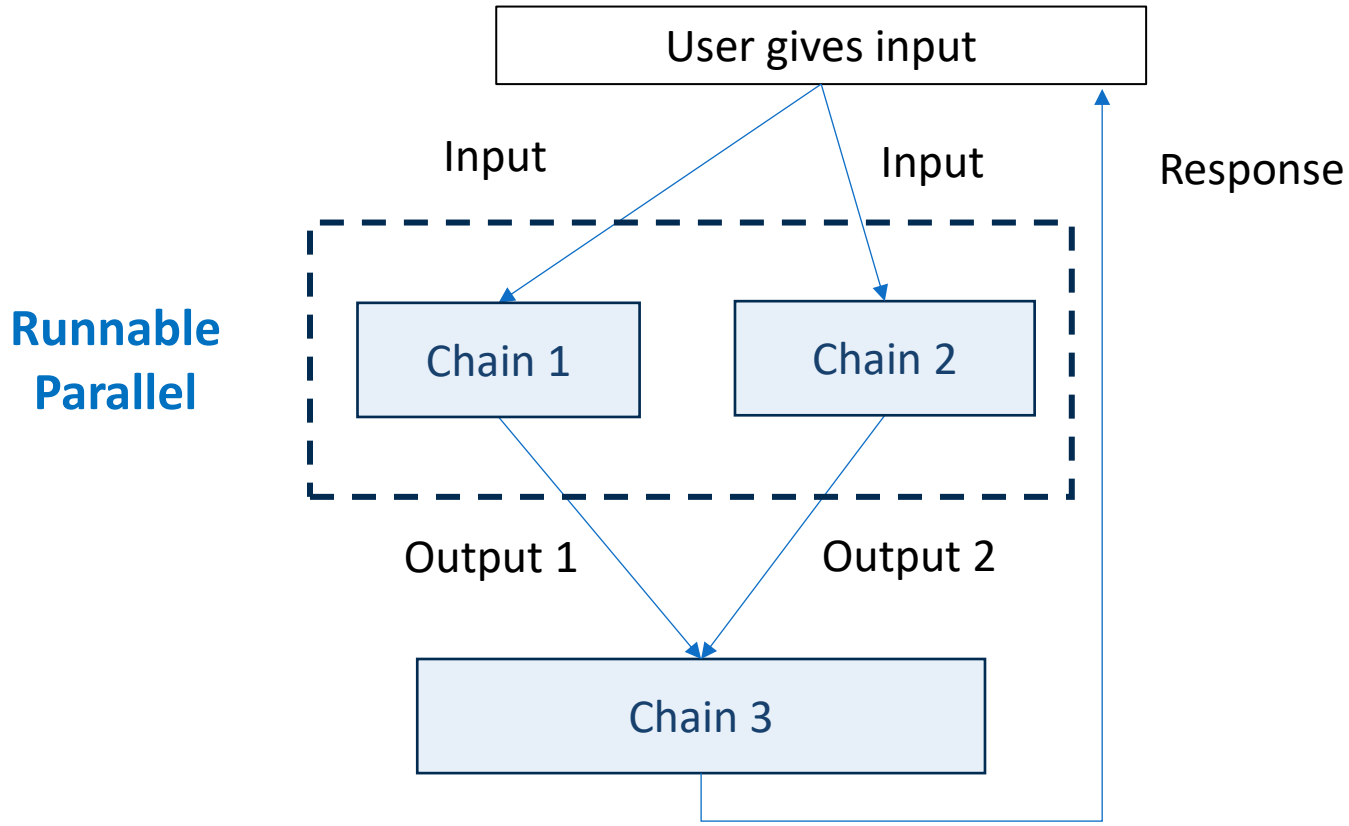
Sequential chains



Chains

Runnable

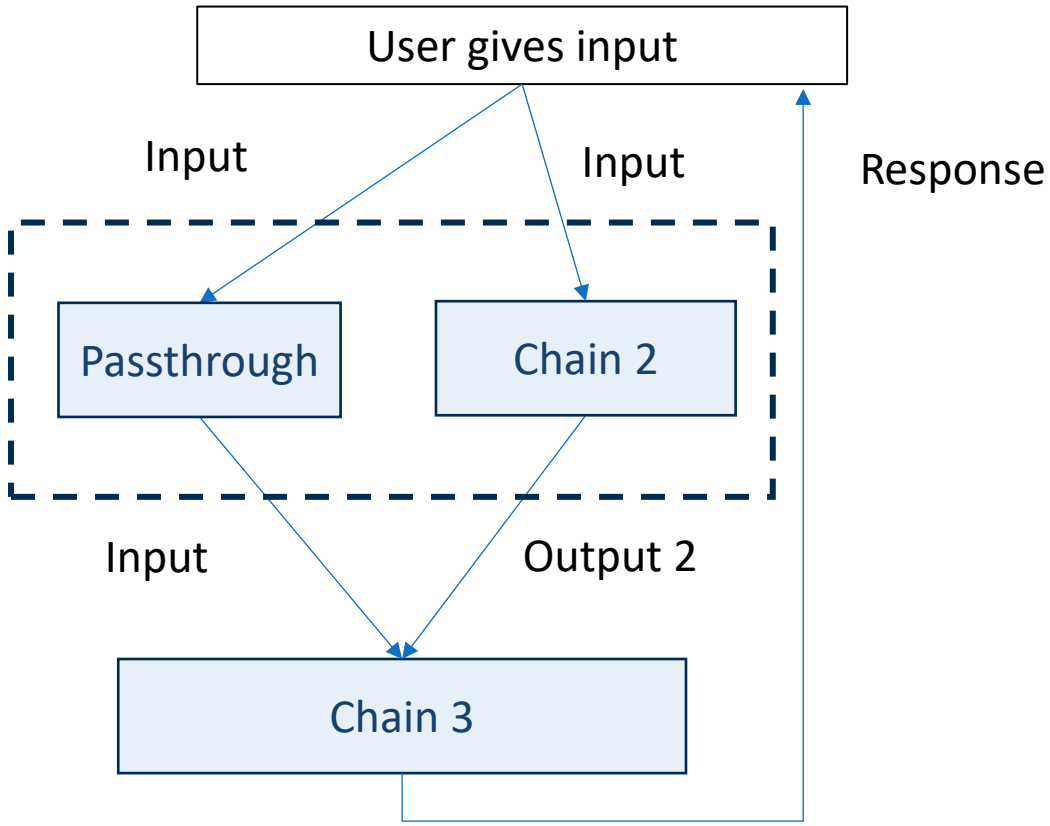
Runnable Parallel



Chains

Runnable

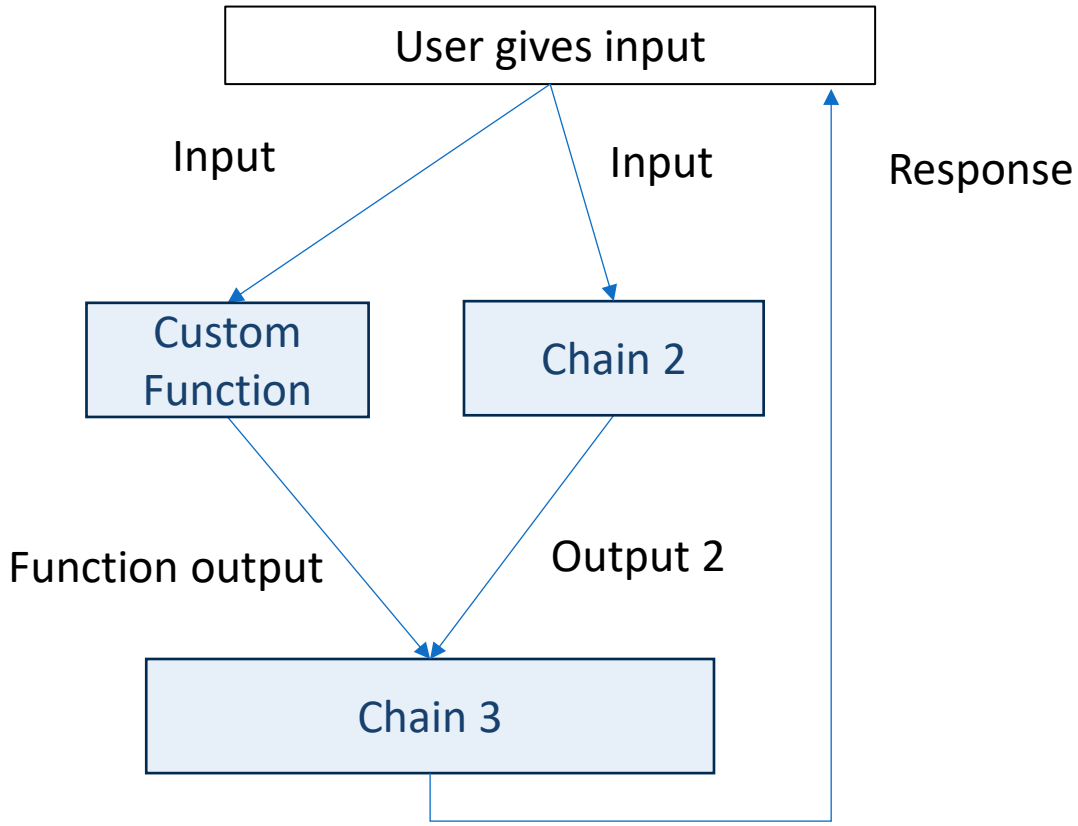
Runnable Passthrough



Chains

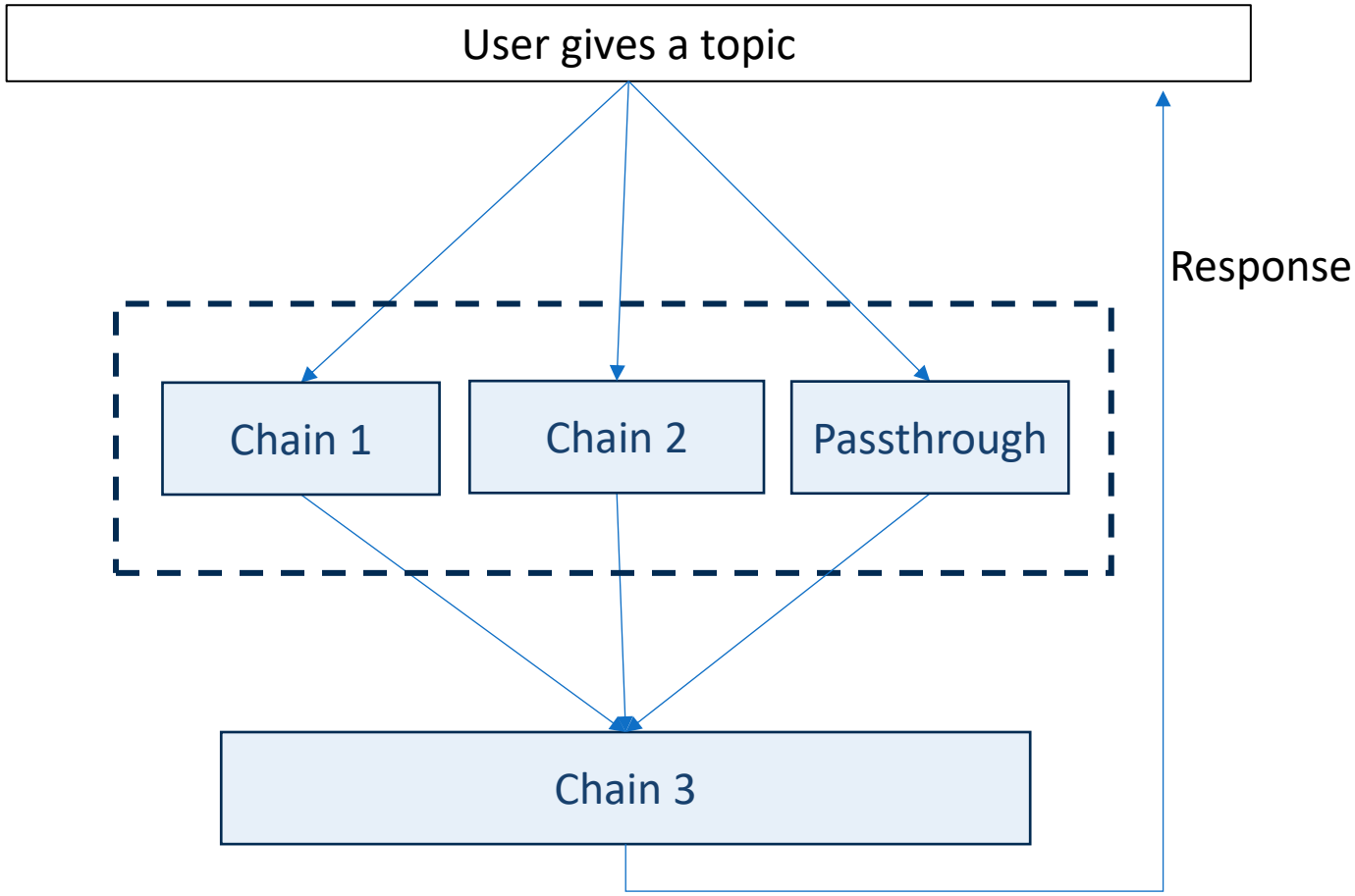
Runnable

Runnable Lambda



Chains

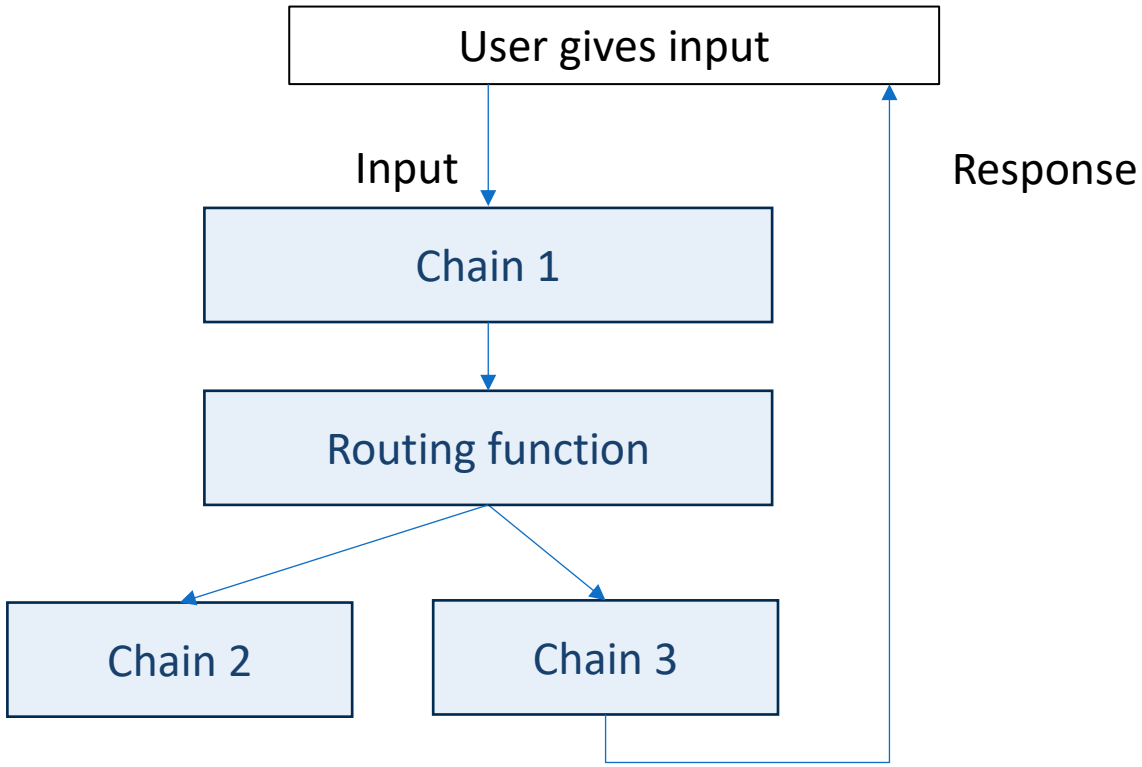
Example



Chains

Runnable

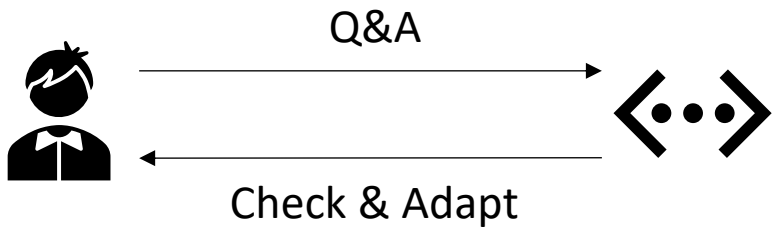
Dynamic Routing



Chains

Problem Statement

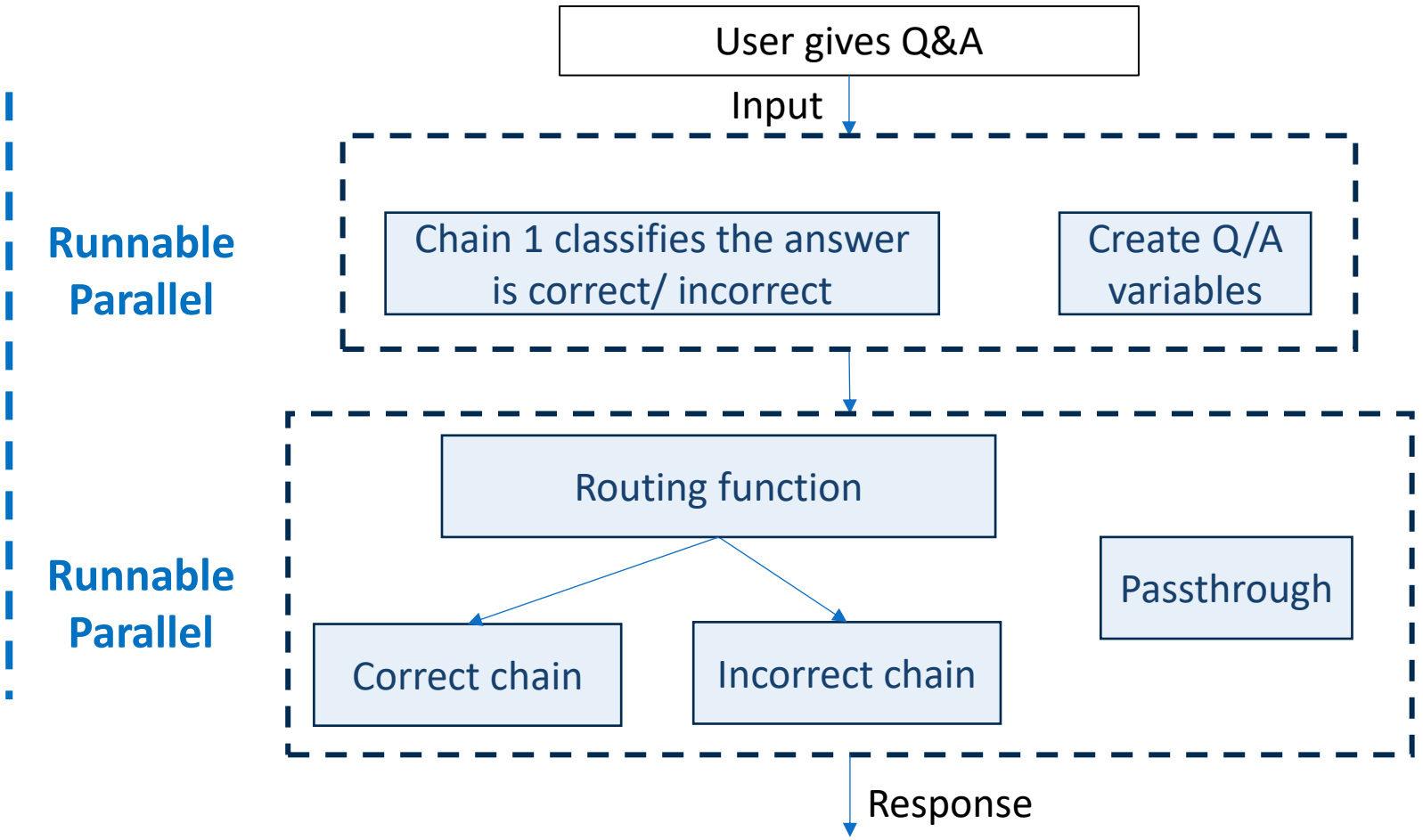
AI Assistant for adaptive practice



User enters a question and an answer. If user answer is correct, the assistant asks a more difficult question on the same topic. If the answer is wrong, the assistant tells the correct answer and explains the correct answer to the student.

Chains

Dynamic Routing



Output Parsers

What?

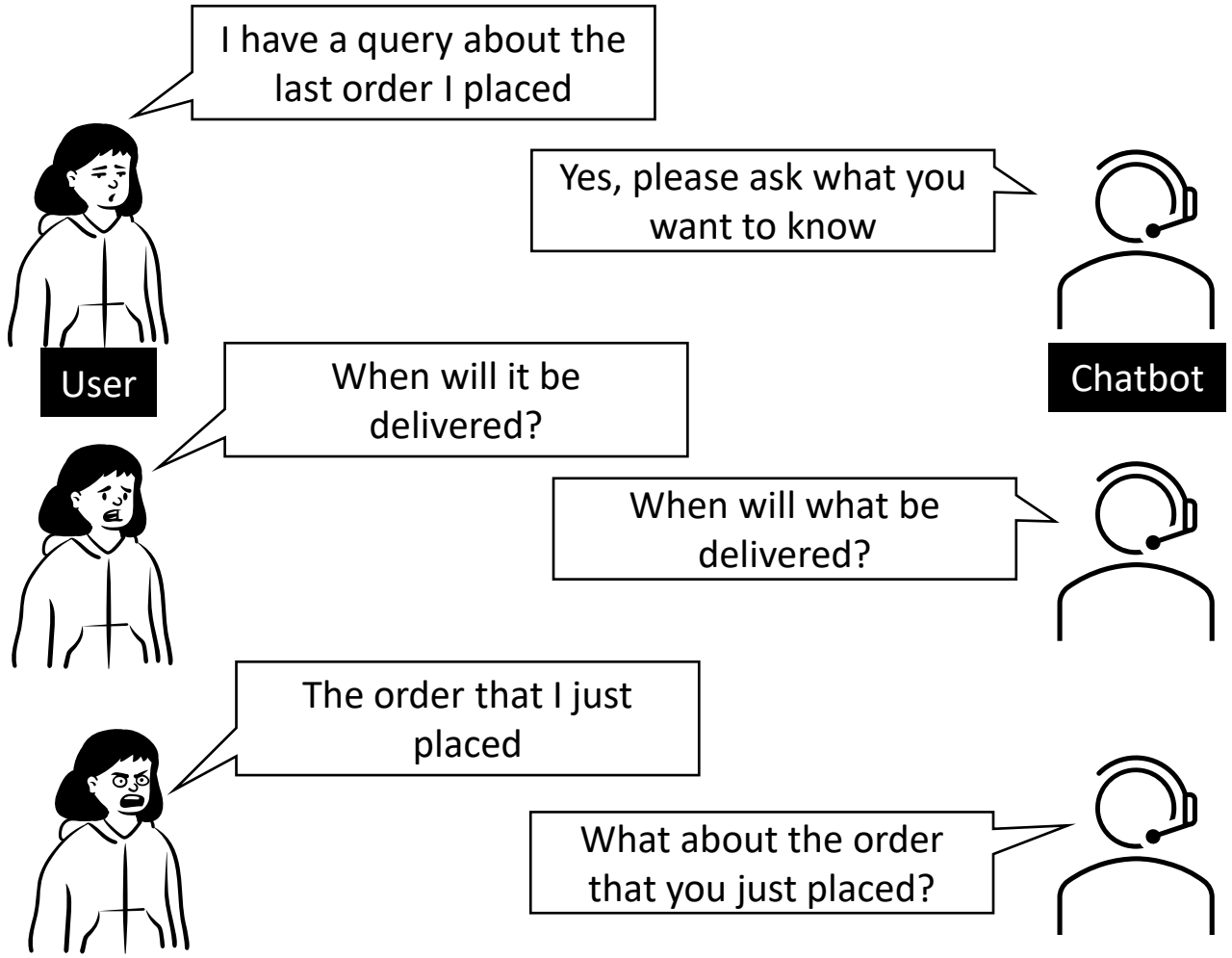
Used to format the output in the desired format

Example applications -

1. Get only string output of the LLM – **StrOutputParser()**
2. Get output in a specified schema – **StructuredOutputParser()**
3. Get output as comma separated values – **CommaSeparatedListOutputParser()**
4. Get formatted date-time values – **DatetimeOutputParser()**

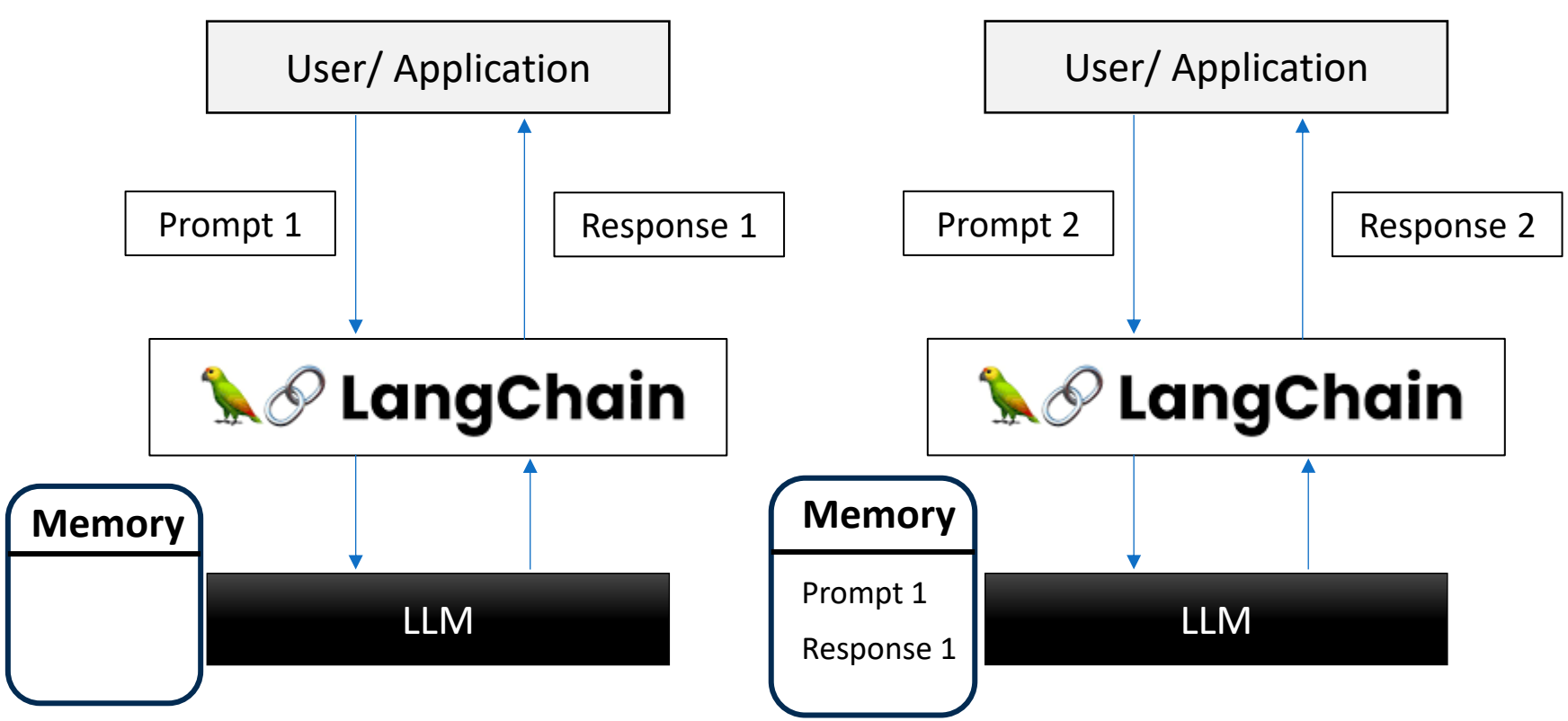
Memory

Why?


I have a query about the last order I placed
Yes, please ask what you want to know
User
Chatbot
When will it be delivered?
When will what be delivered?
The order that I just placed
What about the order that you just placed?

Memory

How?


User/ Application
Prompt 1
Response 1
LangChain
Memory
LLM
User/ Application
Prompt 2
Response 2
LangChain
Memory
Prompt 1
Response 1
LLM

Memory

Conversation

Human: I have a query about the last order I placed.
AI: Yes, please ask what you want to know

Human: I want to change the delivery address
AI: Current delivery address is XYZ. Where do you want the package to be delivered.

Human: Deliver it to ABC
AI: OK. I have updated the delivery address. Anything else I can help you with?

Human: When can I expect the package?
AI:

Types of memories

ConversationBufferMemory
Remembers the entire chat

ConversationBufferWindowMemory
Remember previous n prompts and responses

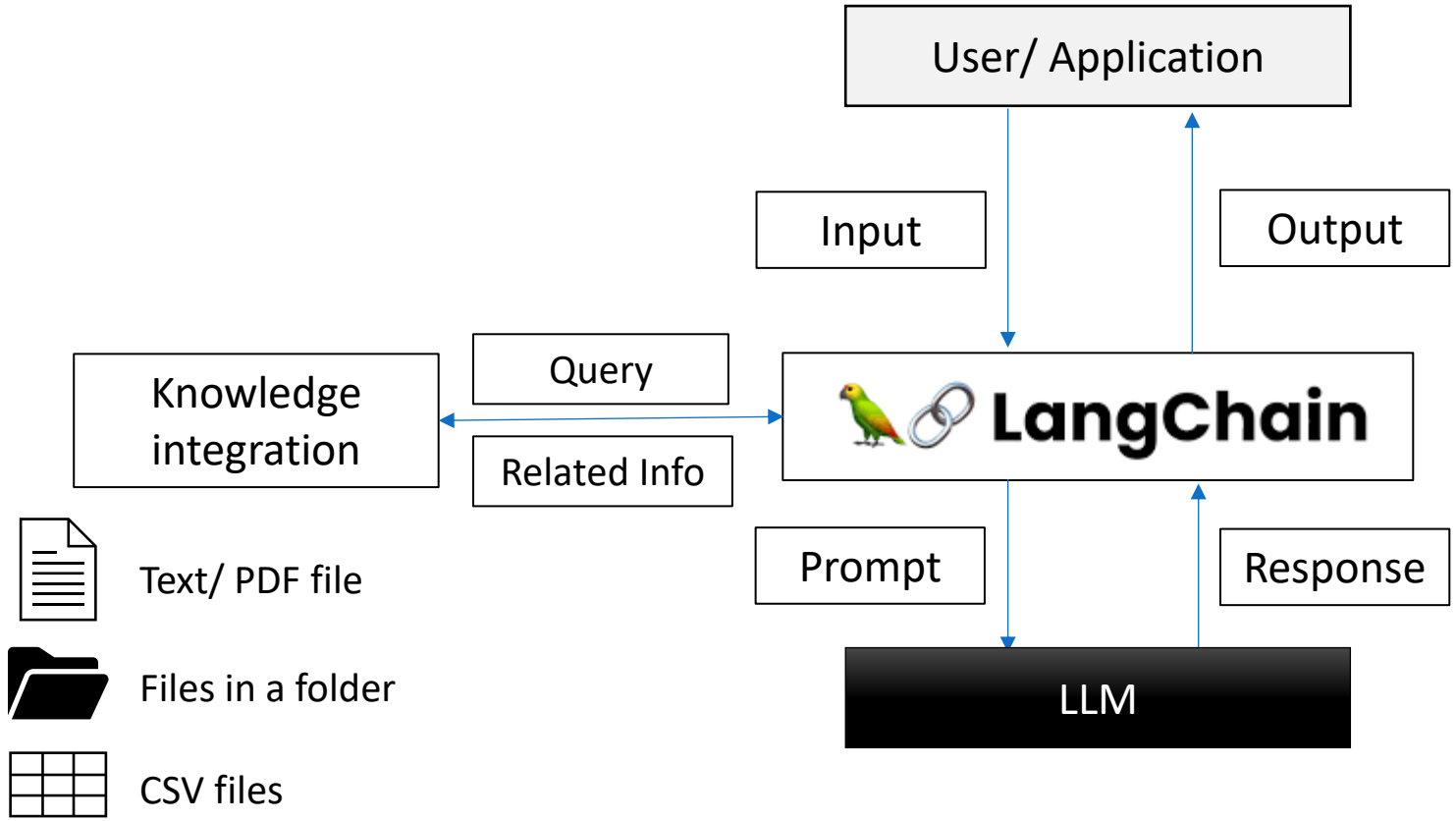
ConversationSummaryMemory
Uses LLM to generate a summary of conversation and remembers that

RunnableWithMessageHistory
LCEL based memory with additional features

Knowledge Integration

What?

Retrieval Augmented Generation (RAG)



Knowledge Integration

Once upon a time in Technopolis, Alex, an analyst at ShopSmart, faced a major challenge: managing the overwhelming volume of customer feedback. Thousands of reviews and complaints were pouring in daily, making it impossible for the customer service team to keep up.

Alex discovered LangChain, a powerful framework for integrating large language models into business applications. Using LangChain's Prompt Templates, Chains, and Memory features, Alex created a system that could summarize, categorize, and detect sentiment in customer feedback.

One day, a surge of negative reviews about a new product defect was quickly flagged by the LangChain-powered system. The issue, identified as a packaging defect, was addressed swiftly. Alex alerted the production team, who halted shipments and fixed the defect. The customer service team reached out to affected customers with solutions, turning a potential crisis into an opportunity for impressive customer service.

Thanks to Alex's innovative use of LangChain, ShopSmart improved its feedback management, automated responses to common queries, and personalized product recommendations. Alex became a hero at ShopSmart, demonstrating how cutting-edge technology could solve daunting business challenges, and ShopSmart thrived in the competitive market.

Chunk	Text
1	Once upon a time in Technopolis, Alex, an analyst at ShopSmart, faced a major challenge: managing the overwhelming volume of customer feedback.
2	Thousands of reviews and complaints were pouring in daily, making it impossible for the customer service team to keep up.
3	Alex discovered LangChain, a powerful framework for integrating large language models into business applications.
4	Using LangChain's Prompt Templates, Chains, and Memory features, Alex created a system that could summarize, categorize, and detect sentiment in customer feedback.
5	One day, a surge of negative reviews about a new product defect was quickly flagged by the LangChain-powered system.
6	The issue, identified as a packaging defect, was addressed swiftly.
7	Alex alerted the production team, who halted shipments and fixed the defect.
8	The customer service team reached out to affected customers with solutions, turning a potential crisis into an opportunity for impressive customer service.
9	Thanks to Alex's innovative use of LangChain, ShopSmart improved its feedback management, automated responses to common queries, and personalized product recommendations.
10	Alex became a hero at ShopSmart, demonstrating how cutting-edge technology could solve daunting business challenges, and ShopSmart thrived in the competitive market.

Knowledge Integration

Chunk	Text
1	Once upon a time in Technopolis, Alex, an analyst at ShopSmart, faced a major challenge: managing the overwhelming volume of customer feedback.
2	Thousands of reviews and complaints were pouring in daily, making it impossible for the customer service team to keep up.
3	Alex discovered LangChain, a powerful framework for integrating large language models into business applications.
4	Using LangChain's Prompt Templates, Chains, and Memory features, Alex created a system that could summarize, categorize, and detect sentiment in customer feedback.
5	One day, a surge of negative reviews about a new product defect was quickly flagged by the LangChain-powered system.
6	The issue, identified as a packaging defect, was addressed swiftly.
7	Alex alerted the production team, who halted shipments and fixed the defect.
8	The customer service team reached out to affected customers with solutions, turning a potential crisis into an opportunity for impressive customer service.
9	Thanks to Alex's innovative use of LangChain, ShopSmart improved its feedback management, automated responses to common queries, and personalized product recommendations.
10	Alex became a hero at ShopSmart, demonstrating how cutting-edge technology could solve daunting business challenges, and ShopSmart thrived in the competitive market.

Question

What was the issue that led to a surge in negative customer reviews?

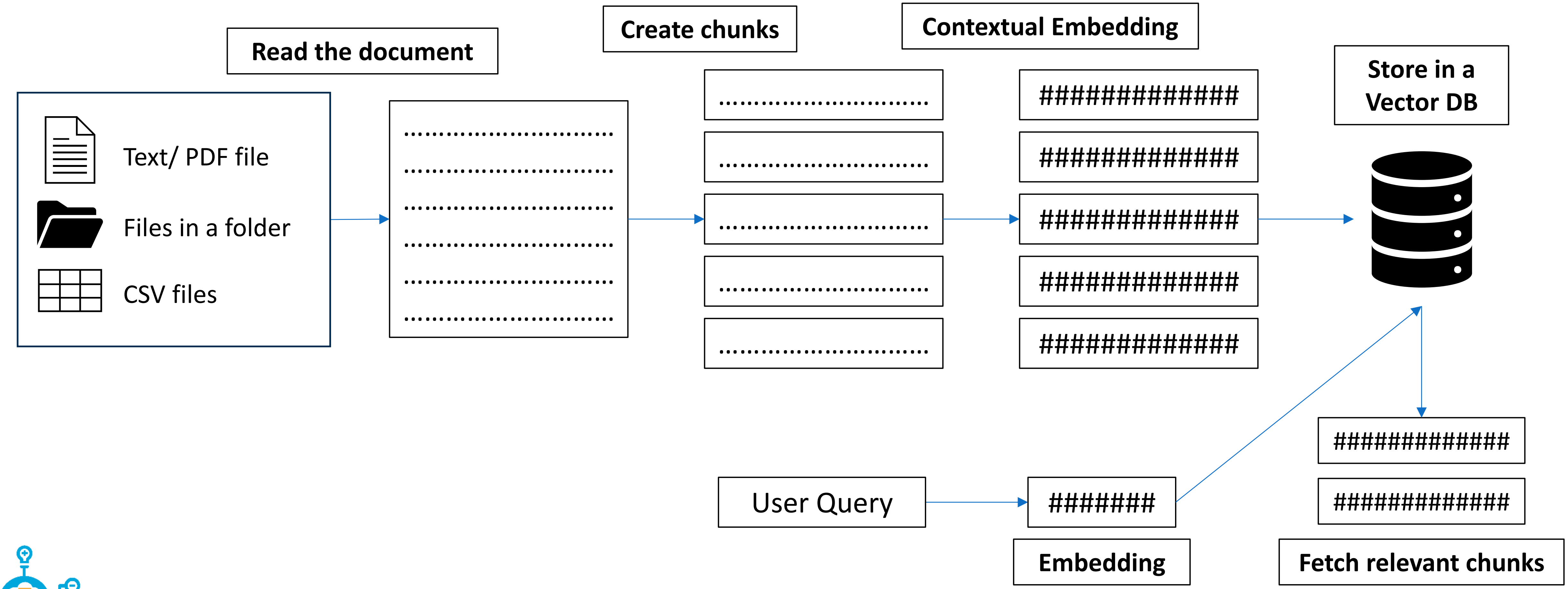
Relevant chunks

Chunk	Text
5	One day, a surge of negative reviews about a new product defect was quickly flagged by the LangChain-powered system.
6	The issue, identified as a packaging defect, was addressed swiftly.

LLM response

The issue that led to a surge in negative customer reviews was a packaging defect in a new product.

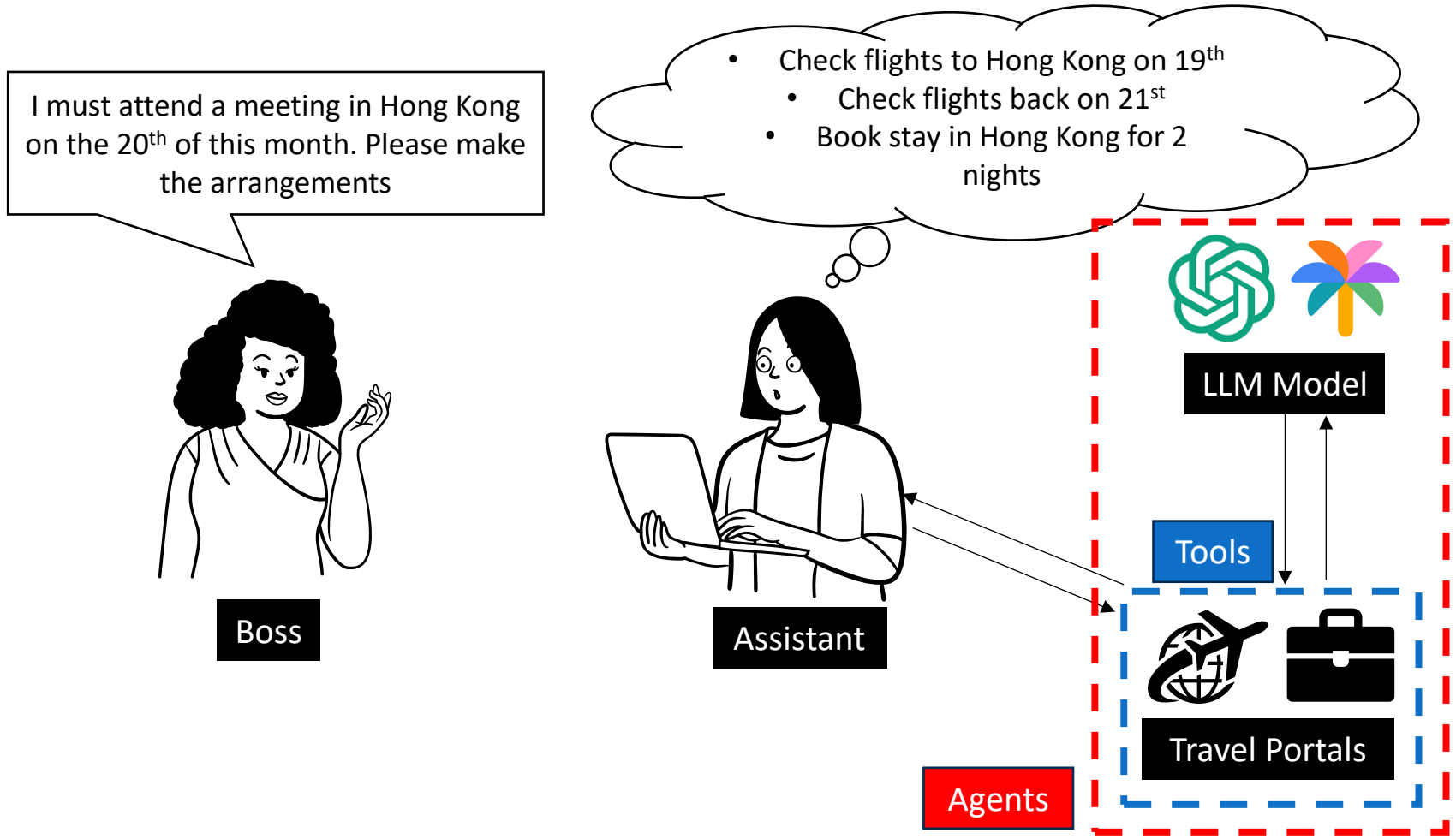
RAG Process



Read the document
Create chunks
Contextual Embedding
Store in a Vector DB
Text/ PDF file
Files in a folder
CSV files
....
#############
User Query
#######
Embedding
Fetch relevant chunks

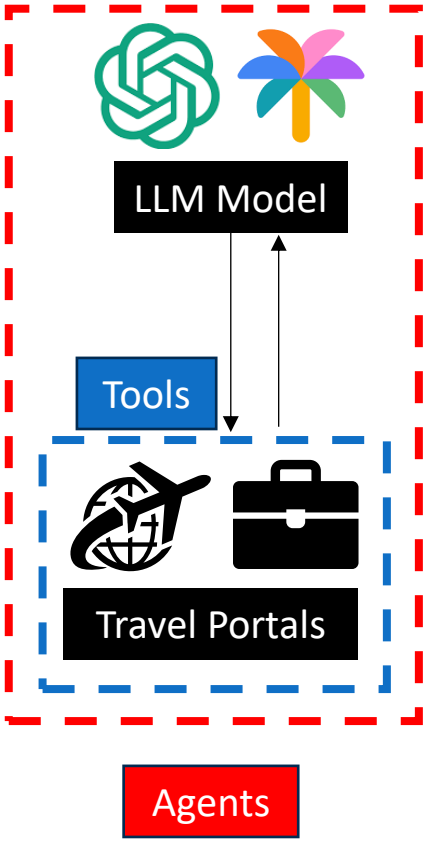
Tools and Agents

What



Tools and Agents

What's covered



- Creating own custom tools
- Use LangChain in-built tools
- Create Agents
- Create Agents with chat memory

Tools and Agents

Steps

- Creating own custom tools
- Use LangChain in-built tools
- Create Agents

1. Define the tools
2. Attach tools to LLM
3. LLM tells which tool to call
4. Invoke the tools
5. Send tool output to LLM to get the final result

LangSmith

What & How

What

Platform for monitoring and debugging LLM based applications

Use

- When we have multiple LLM based applications, and they are going into production.
- LangSmith can help in debugging code, evaluating performance and keeping an eye on usage metrics

How

- Create a login on LangSmith and generate an API key
- Link - https://smith.langchain.com/

Streamlit

What & How

What

Open-source Python library that makes it easy to create and share custom web apps

Use

Quickly create Graphical User Interface (GUI) for your application. Can serve as a POC for a bigger project

Samples: https://streamlit.io/gallery

How

- Install the Streamlit python library and run on local/ cloud server
- Use the Streamlit server to host your application

Getting Started

- pip install streamlit
- streamlit hello