

Building LLMs application with LangChain

Extra Class: LLMs



Dinh-Thang Duong – TA
Nguyen-Thuan Duong – TA

Objectives



LangChain

```
Curl
curl -X 'POST' \
  'http://0.0.0.0:5000/generative_ai' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "question": "what is BERT?"
  }'
```

Request URL

http://0.0.0.0:5000/generative_ai

Server response

Code	Details
200	<p>Response body</p> <pre>{ "answer": "BERT is a bidirectional transformer model for natural language processing pre-trained on a large corpus of text. It stands for Bidirectional Encoder Representations from Transformers. BERT was designed to be similar to OpenAI GPT, and it outperforms other language representation models in various tasks. The table shows the GLUE test results for BERT and OpenAI GPT, and the table shows the SQuAD 1.1 results for different models. BERT comes in two sizes: BERT BASE and BERT LARGE. The larger model has more parameters and generally performs better. Fine-tuning BERT on different tasks is illustrated in Figure 4. Additional details and ablation studies are presented in Appendices A, B, and C." }</pre> Download

In this lecture, we will discuss about:

1. What is LLMs in Production?
2. What is Langchain?
3. Basic components of LangChain.
4. How to use LangChain to deploy an API?
5. How to use LangChain to deploy a RAG application?

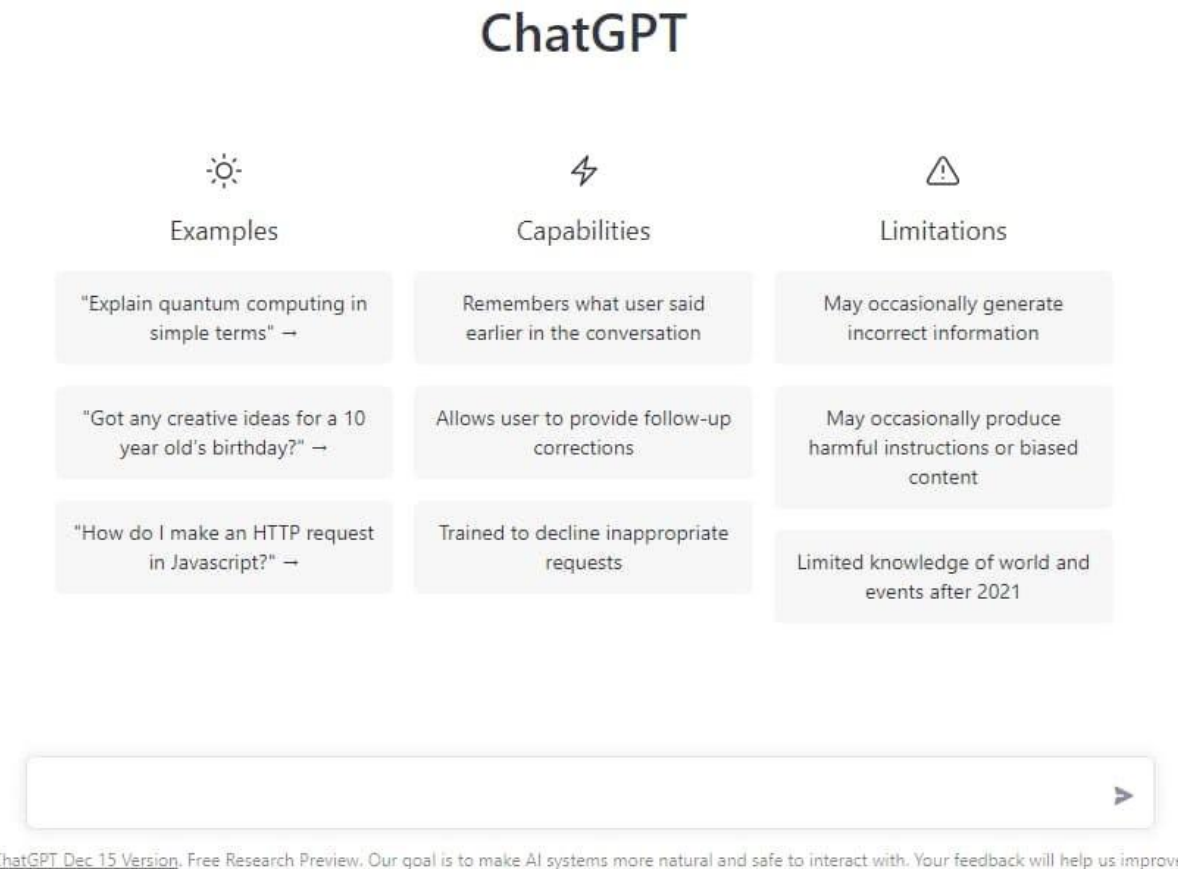
Outline

- Introduction
- LangChain
- API with LangChain
- RAG with LangChain
- Question

Introduction

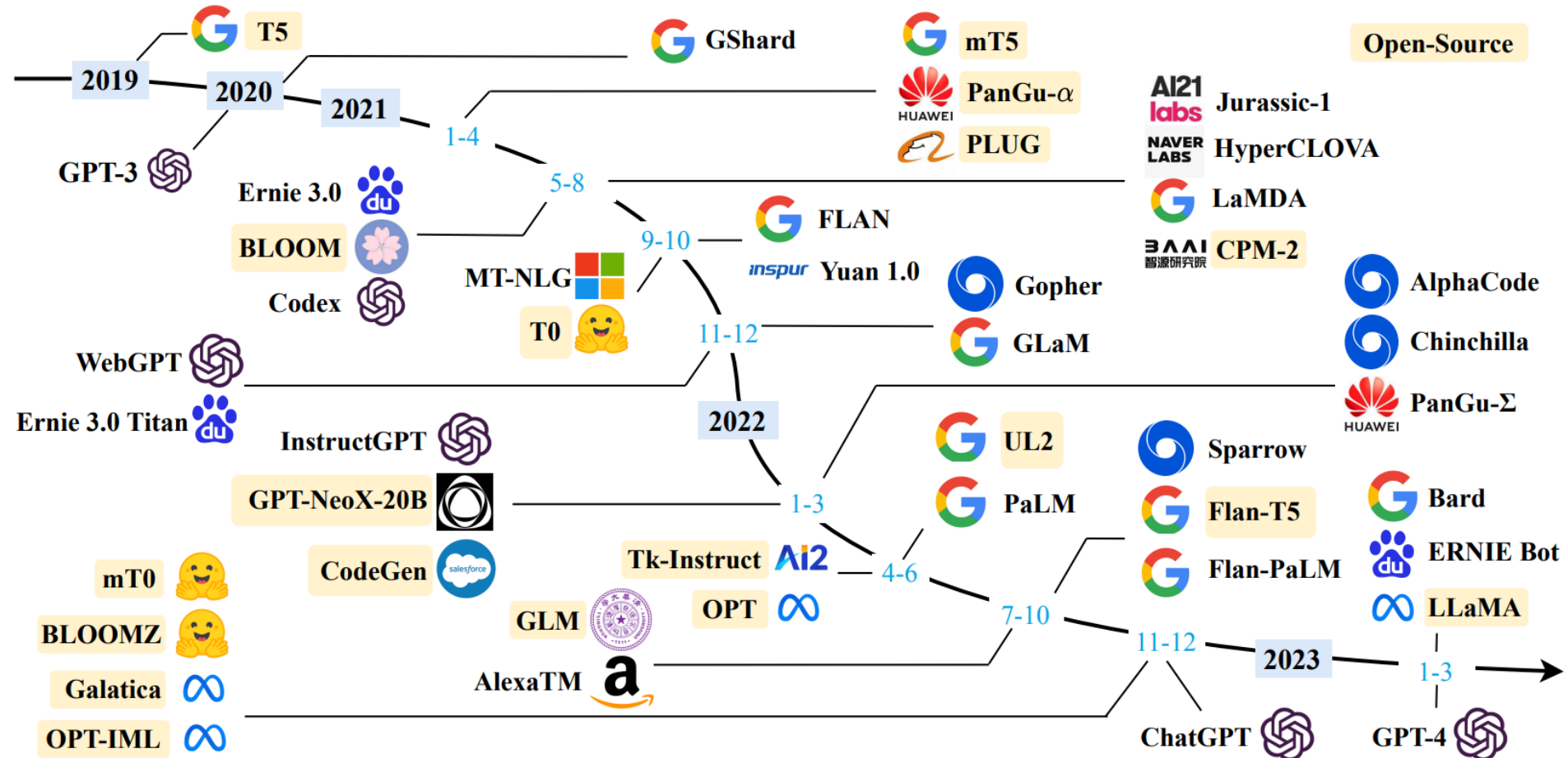
Introduction

❖ Getting Started



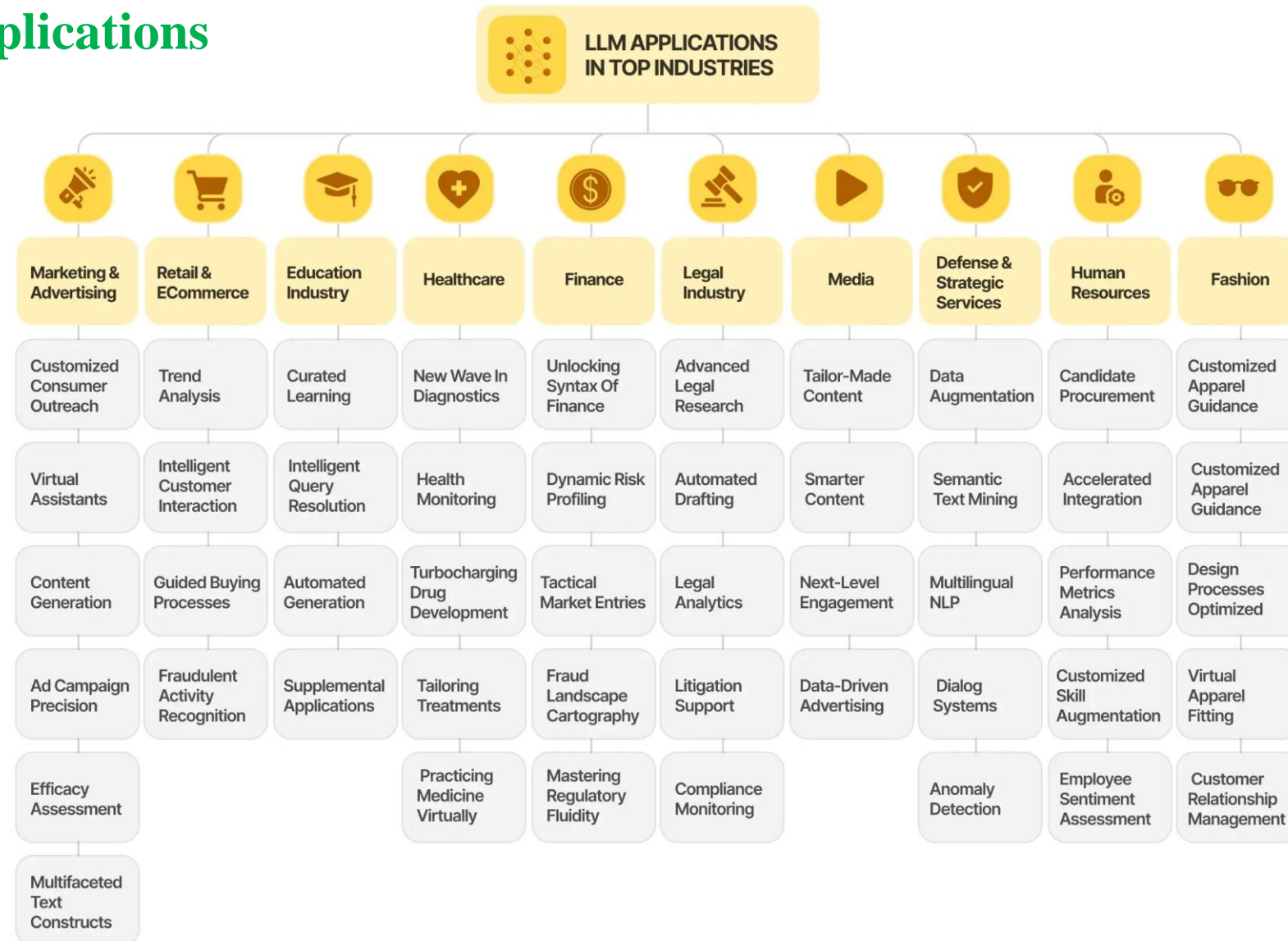
Introduction

❖ LLMs size over time



Introduction

❖ LLMs Applications



Introduction

❖ Getting Started

AI Application + Data Products

Q&A Webapp

Chatbot

Model as an API

LLM Pipeline

Corpus
Creation

Text Pre
Processing

Prompt
Engineering

LLM
Inference

Generated
Text

LLM Model(s)



GPT 3.5



GPT 4.0



LLaMA



Hugging
Face



MPT

and
more..

Overview of LLMs in Production



LangChain

LangChain

❖ Introduction

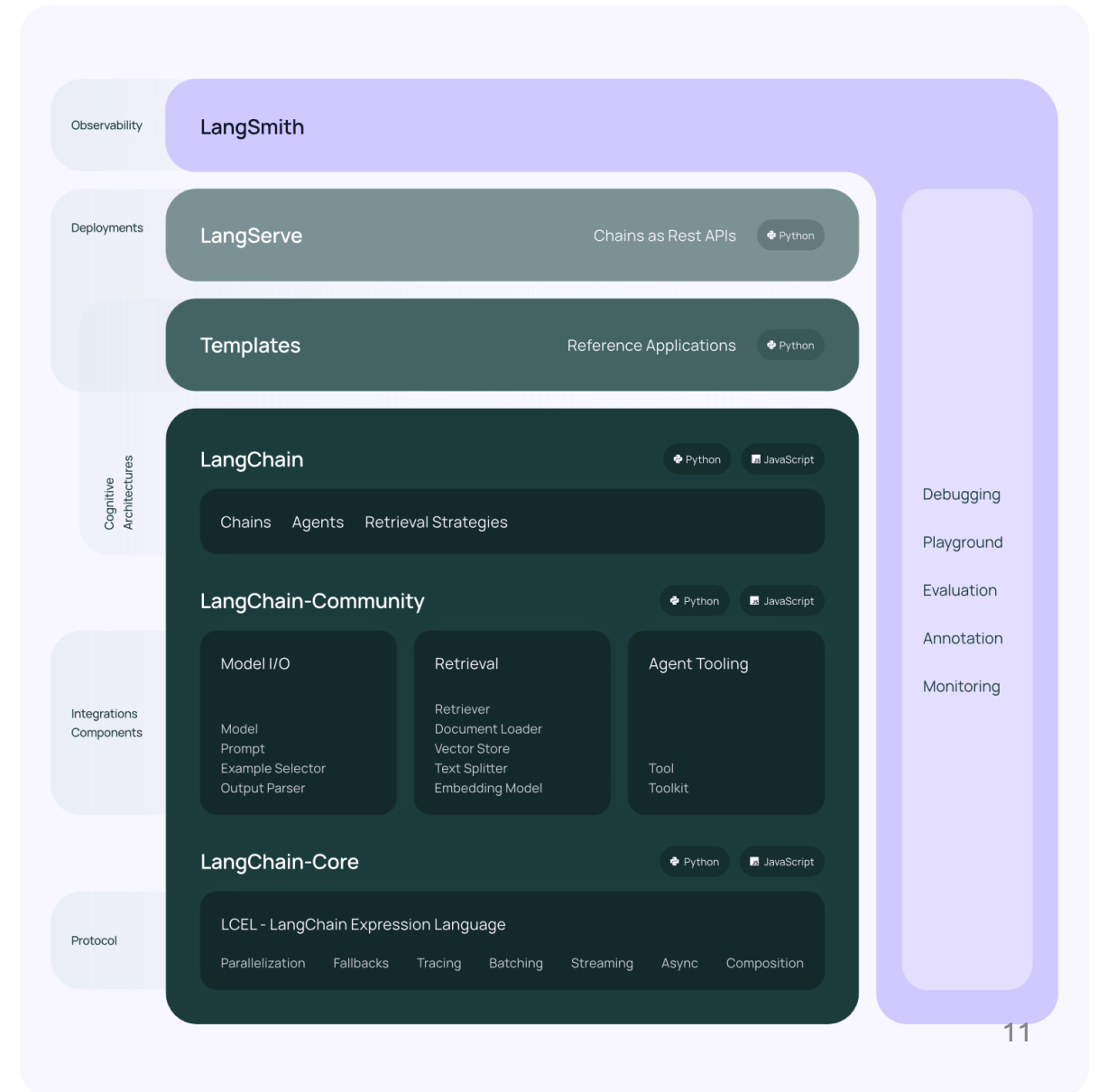


LangChain: A framework for developing applications powered by large language models (LLMs). LangChain simplifies every stage of the LLM application lifecycle: Development, Productionization, Deployment.

LangChain

❖ Introduction

- **Development:** Build your applications using LangChain's open-source **building blocks** and **components**. Hit the ground running using **third-party integrations** and **Templates**.
- **Productionization:** Use **LangSmith** to inspect, monitor and evaluate your chains, so that you can continuously optimize and deploy with confidence.
- **Deployment:** Turn any chain into an API with **LangServe**.



LangChain

❖ LangChain components

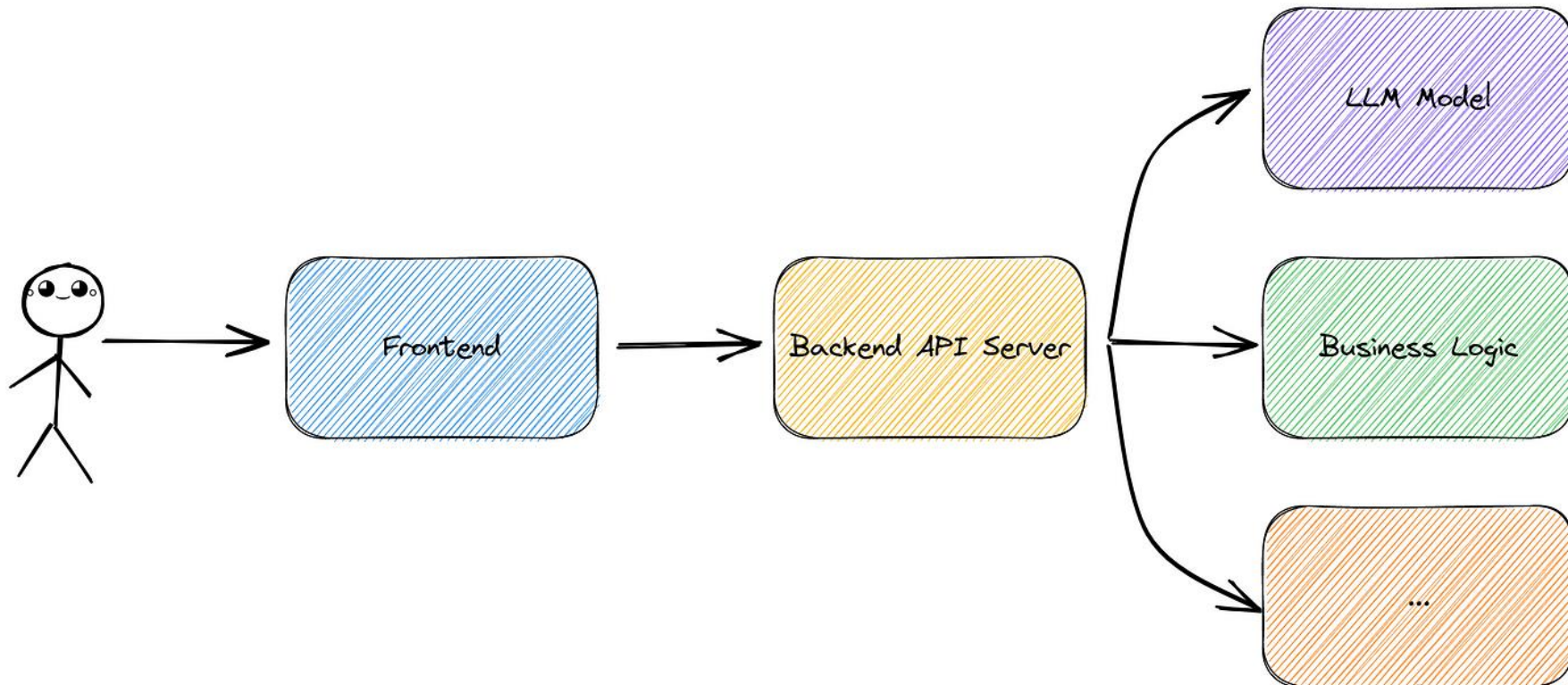


API with LangChain

API with LangChain

❖ Introduction

Description: Serve a LLMs chat application as an API that receive a simple question and return the response of the LLMs (pre-trained model).



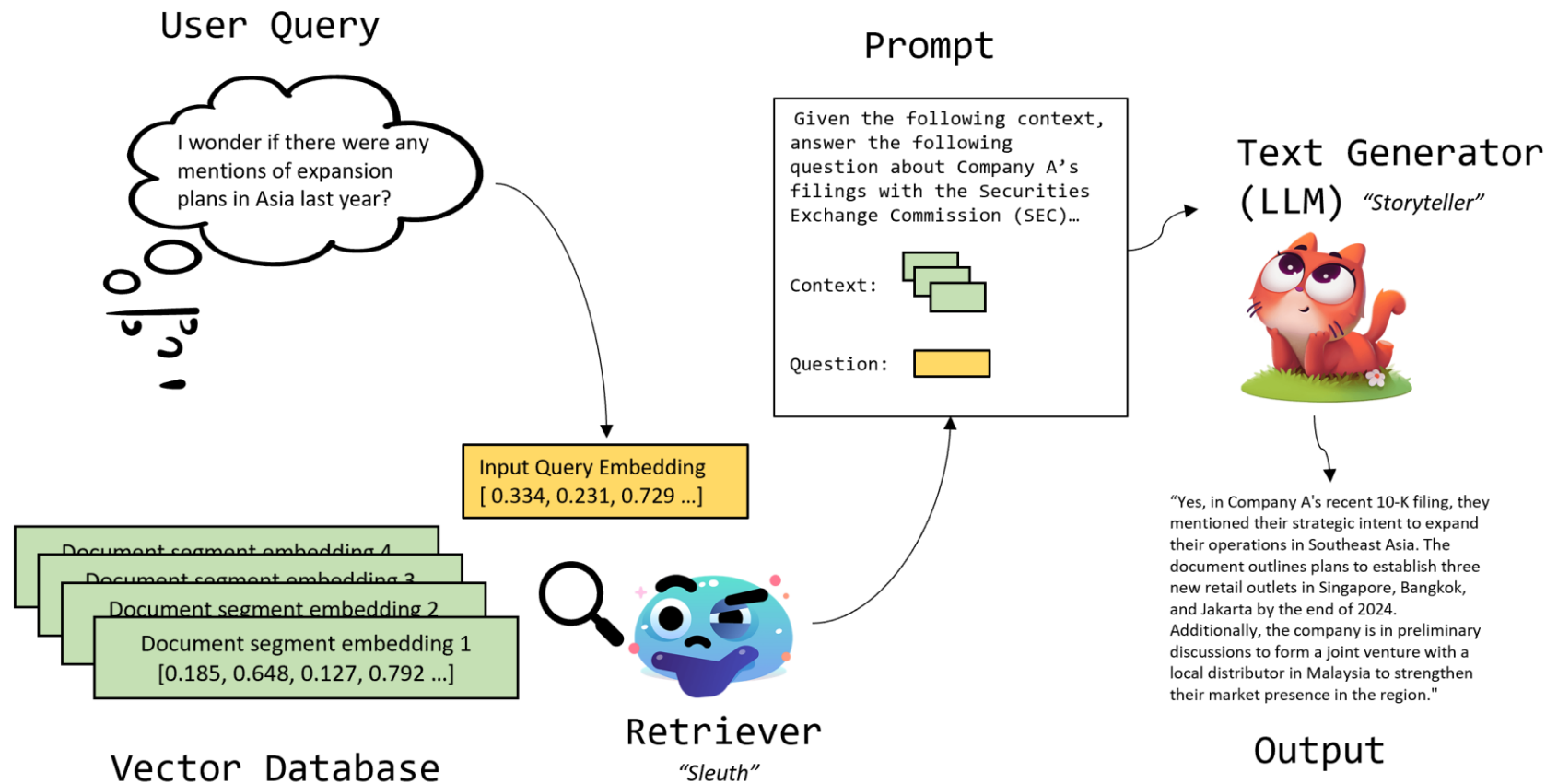
Quiz

RAG with LangChain

RAG with LangChain

❖ Introduction

Description: Serve a LLMs RAG application as an API that receive a simple question and return the response of the LLMs (utilizing context retrieved from a vector database).



Summary

In this lecture, we have discussed:

1. What is LLMs in production?

1. How is it differ from LLMs in research?
2. What are some challenges when deploying LLMs in production?

2. Basics of LangChain

1. The key concept of LangChain.
2. Basic components of LangChain: Prompt Template, LLM Chain, Chat History, Document Loader...

3. How to build an API using LangChain.

4. How to build a RAG application using LangChain.

1. Retrieve and answer questions related to Academic Paper.

Question

