1) levels of Archictecure → 1)
2)
3)

(2) - Professional level LLM archictur preview

(3) . Factors for seluting LLM .

1) Requirments -
2) Archictelure
3) Model size
4) Speed Latency
5) Resource & Hardware
6) Cost &
7) Ethical Answer

4 - Contex Window
Multimodality
Accuray
Samdle size.

Llama
Mistral
LLAMA
Bloom
Falcon

→ Opensoure Model

Closeed Some Model

→ LLM → Orchestration → Langcha, LlamaIndex

↳ Backend → Flask, Streamlit, FastApi

Frontend →

Vector Datause →

Cloud storage →

CICD Deployment →

→ Peng langchain

Chro, Faise, weaviate, Pinecone

→ Open ai - pricing → Chat Model

2) Chat Compentio. Models.

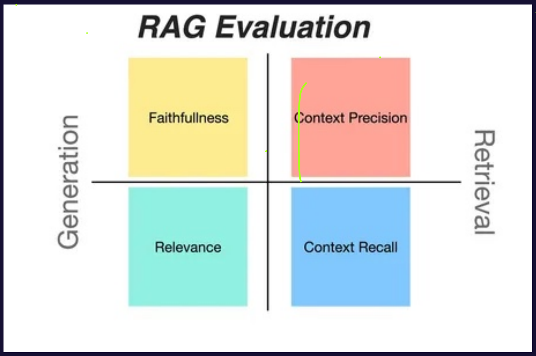
1) Level of Archicteature

2) preview of LLM Archiclture

3) Tech Used in Coorse

4) Factors of Selecting LLMs

5) RAG Models introdution → Code.



RAG Evaluation

Generation

Faithfullness | Context Precision

Relevance | Context Recall

Retrieval

- Hallucination
- Smaller Models
- AI Agents
- Multi Models
- Responsible
- personalised
- Contextual Understand

Techique

1) Modelle Selution

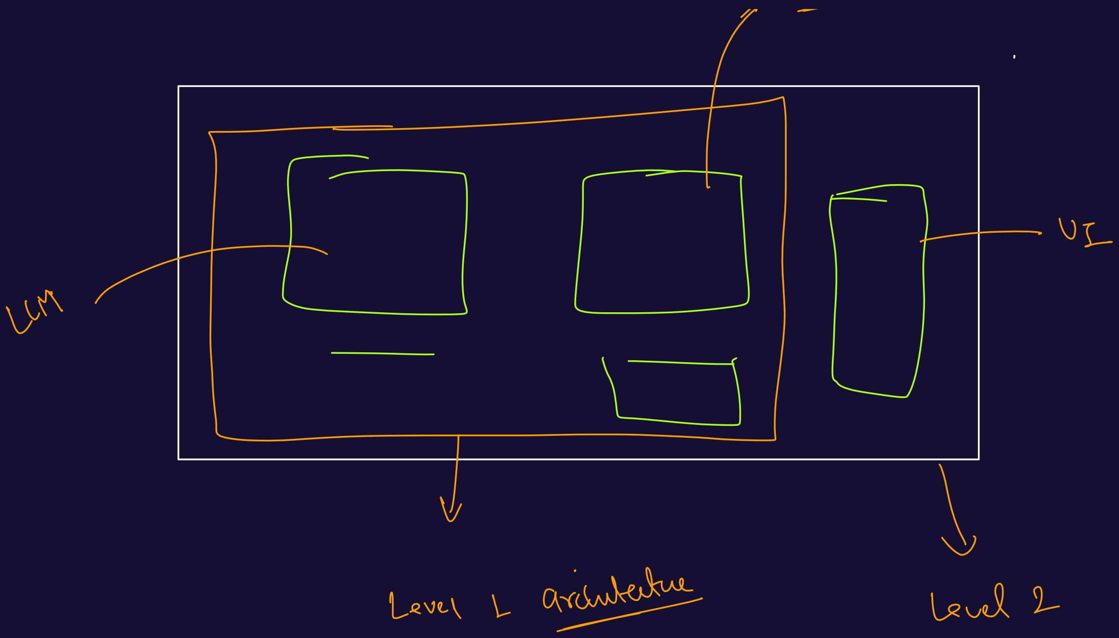
2) Fine toneing pretrained Model

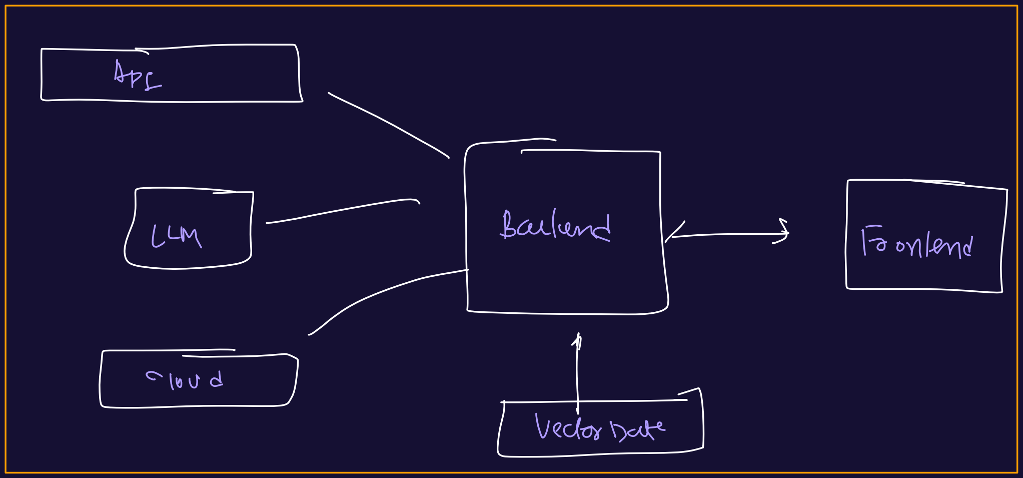
3) RAG

3) Prompt Engineering

4) LLOMps

Framework


LLM
UI
Level 1 architecture
Level 2


API
LLM
cloud
Backend
Frontend
Vector Data

1) LLM → Openai , Hugging , Llamma, GoogApI

2) Orchestration Framework → Langchain , Llamaindex

3) Backend → Langchain , Langgraph .

4) Frontendend → Flask, Fast Api, Streamlit

5) Vector Database → pinecone, Chromadb, Faiss, weaviate

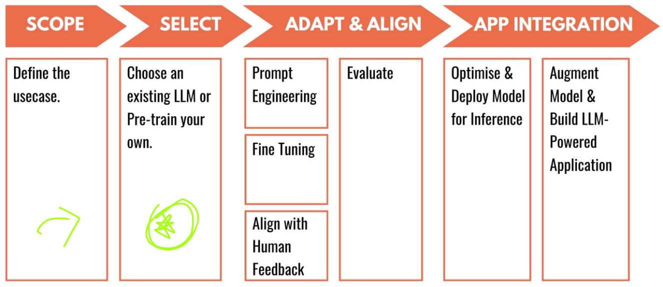
6) Cloud storage → Gcp bucket , S3 bucket

7) Deployme → CICD → Github Actions, Circle CI, Jenkensis

Docker ,

8) Multi Agents → CrewAI, Langgraph, Autogen.

⊕ Aws Bed Rock , Vertex AI → Use LLM Globally .



SCOPE	SELECT	ADAPT & ALIGN		APP INTEGRATION	
Define the usecase.	Choose an existing LLM or Pre-train your own.	Prompt Engineering Fine Tuning Align with Human Feedback	Evaluate	Optimise & Deploy Model for Inference	Augment Model & Build LLM-Powered Application

(1). Factors to Select → LLMs → → Which is best ?

(2). Future trends Achievements → Topic important ⊛

(3). Life cycle of Genai Model → Technique

↳ (1) Tech stack

(4) → Tools → which import – ?

⊛ Future → 1) Responsible AI → prompt

2) Multi Agent / Agentic AI →

3) Multimodel →

4) Personalized → Finet / RAG / prompt

6) Contextual → Memory concept

Life Cycle → Select Model (1) → How to select

(2) Data → processing → * Load / embedding / vectordatabase

⊛ (3) → Technique → Prompt / Finet / memory / RAG.

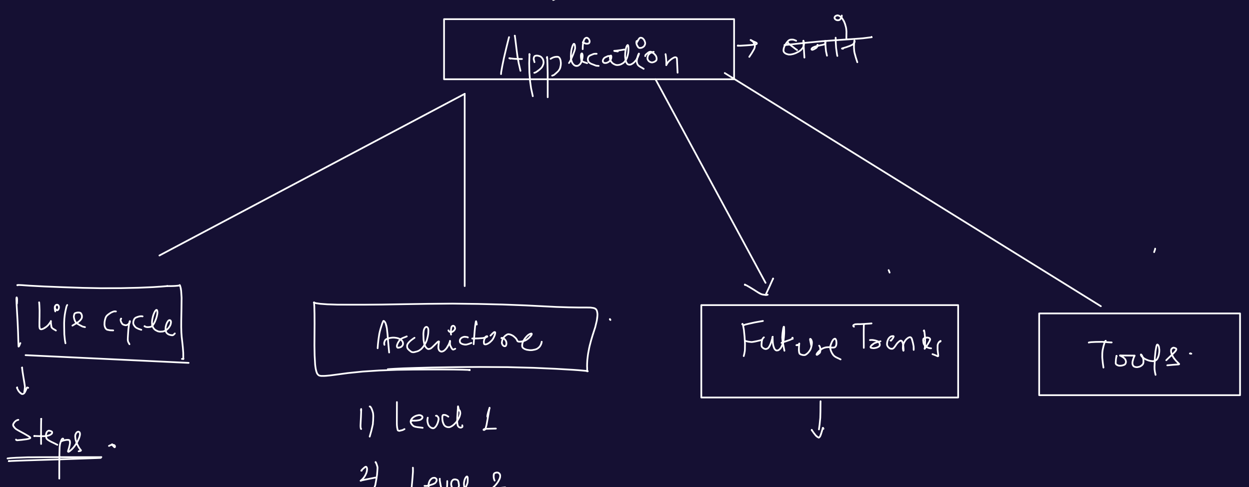
(4) → Evaluation →

5 Deploy → LLMOps ⇒ Aws Bedrock

Vertex AI

Langsmith

Tech Tools ⇒



1) Level 1

2) Level 2

3) Level 3

1

2

*

Steps →

2 Concept | Topics / Technique →

3 → Tools. | Platform.

* Sample tools. / Platform.

→ How to sellect LLM Models

→ Data process | Ready →

→ RAG introduction.

How to Decide:

1) Requirements. (Text, image, Multimodal) → Video Graphers, Audio
2) Model Size
3) Accuracy → Hospital
   Architecture
4) Latency | speed → Auto mobile
5) Hardware & Resources.
6) cost & Lincensing (openai)
7) Responsible AI - (Not Biased) → Hospitality
8) Context window → Trading
(9)

↓

Why open ai

① Ecosystem

+

② Connection with Others

Models → open Vs Closed Source

↓ Free	↓ Closed
1) Llama	1) GPT 4
2) Mistral	2) Claude
	3) Gemini

Response ← 1)

↓

Error

3) Falcon

4) Bloom.

4) Cohere

⊛ * Accuracy, Latency, Ecosystem ⇒

⊛ - How to use → ① API

② Download

③ Cloud Based → Aws Bed Rock / GCP

Vertex AI

url Pdf website Text Json csv

Data Related Issues -: 1) Data loaders. →

2) Chunks | Split ⇒

3) Embeddings ⇄

4) Vector store |⇒ Diffent.

①

① * RAG ⊛ Finetuning ⊛ * Prompting

(*) AI Agent

(*) → - RAG.

1)