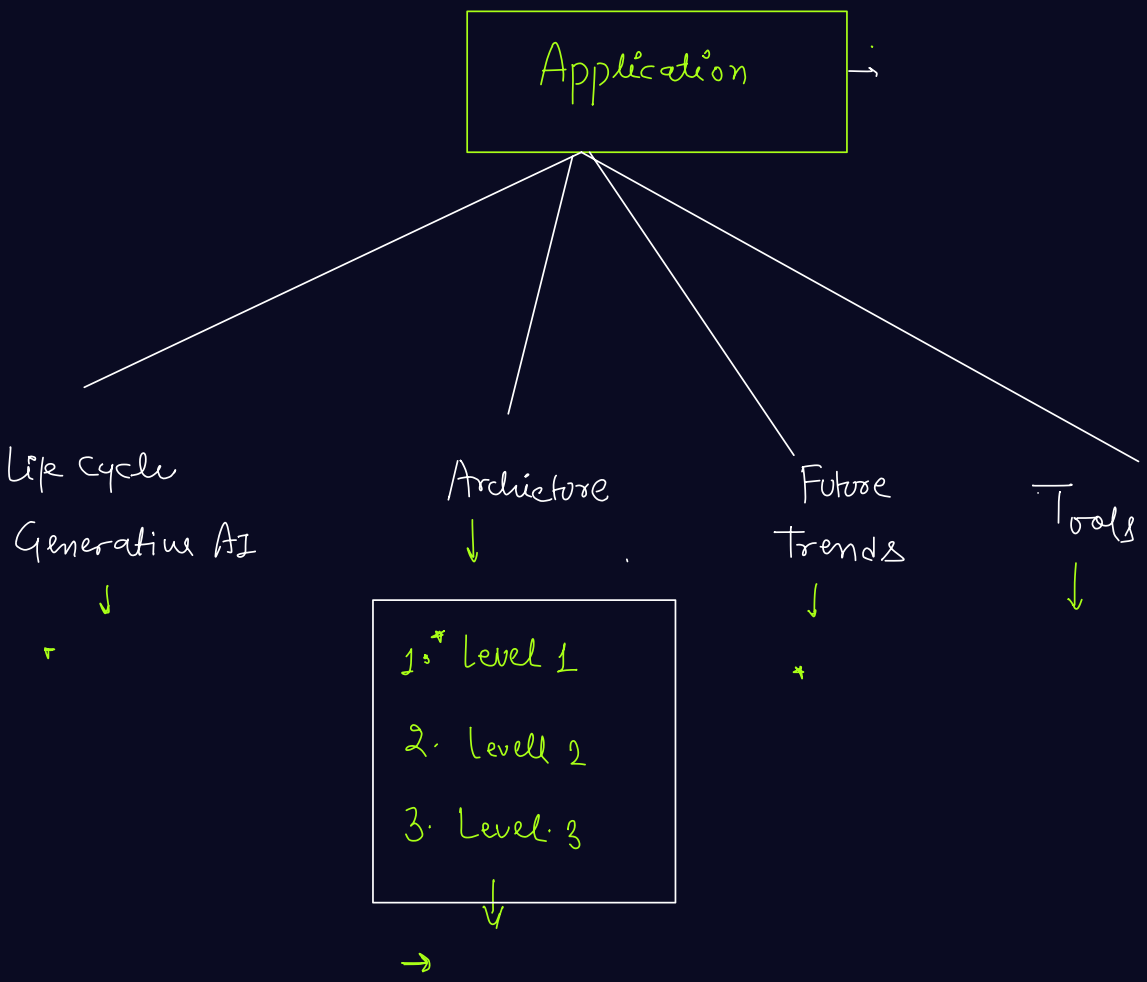
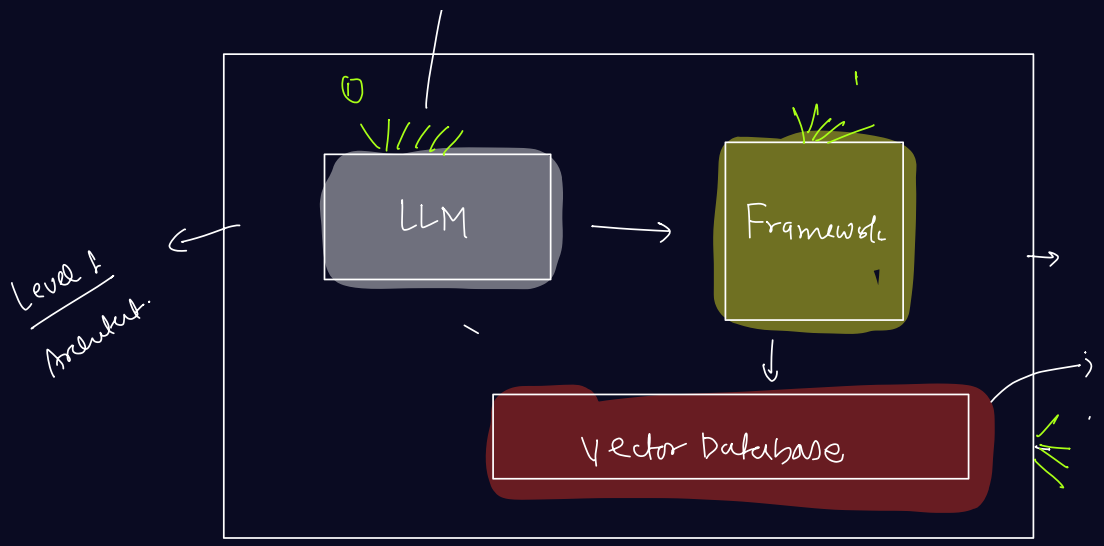


Level 1 architecture → Project → Architecture.

① .

Langchain

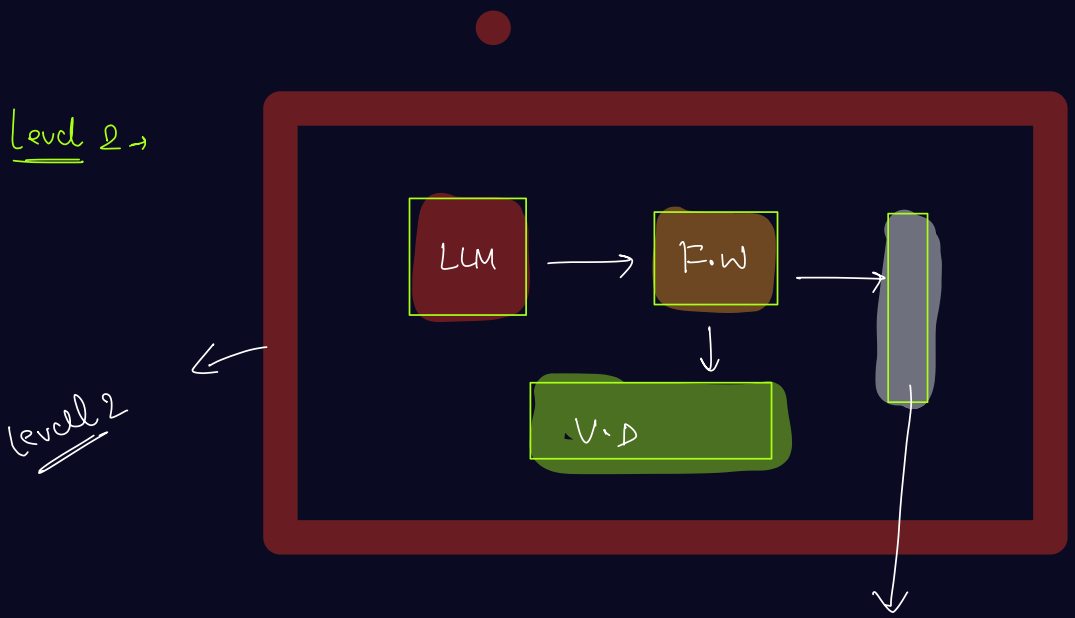


Level 1 Architect.

1) LLM → Large Language Model

2) Orchestration Framework → Langchain / Llamaindex

3) Database → Vector Database



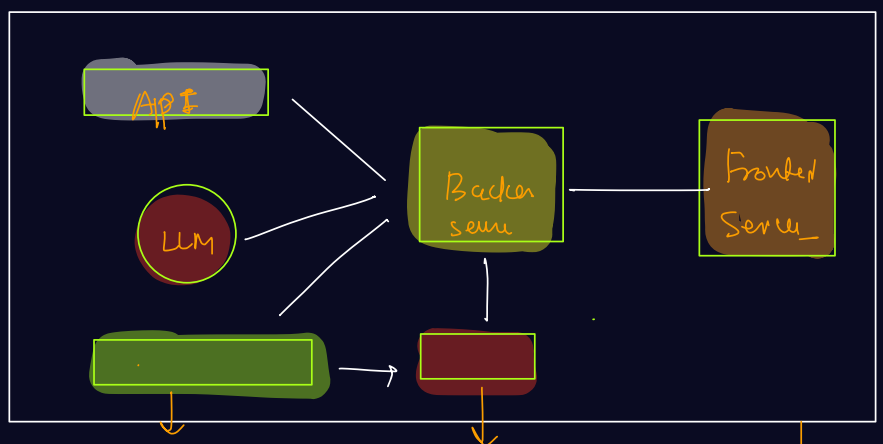
Level 2 →

Level 2

1) Flask ← UI / Frontend
2) Streamlit
3) FastApi

Level 3 =>

Industry 8.

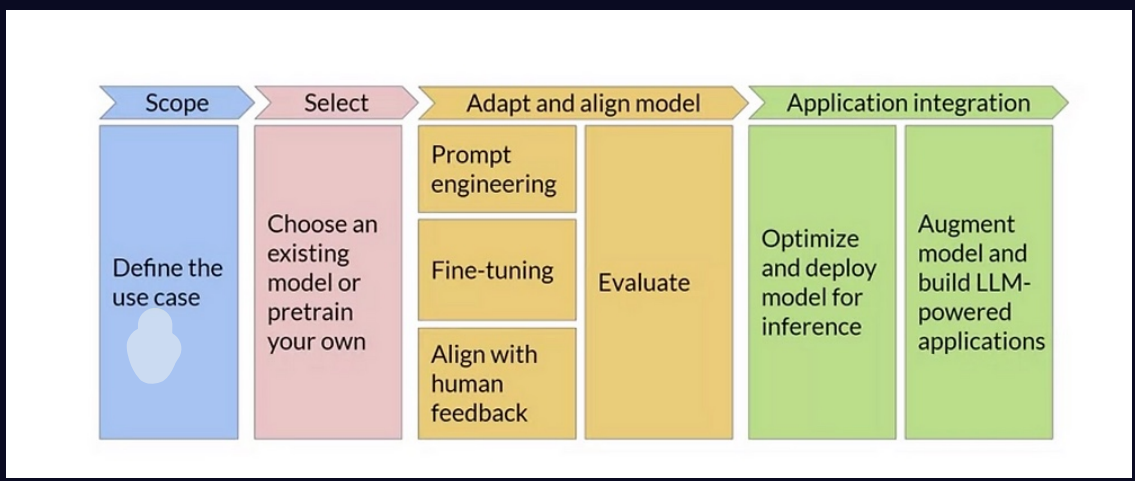


(Vectore) Database

Cloud (AWS/ GCP)

* Memory

CICD



Life Cycle ⇒ 1) Model Selection - ①

⑤ ← 2) Data preparing → Chunks, embeddings, vector

3) Technique Generative →

① ← 1) Prompting
2) RAG
3) Fine tuning

4) Evalution.

* Deploy / Integramtion → LLMOps

AWS bedRock, Vertex

(3) → Future Trends ⇒ Generative →

1) Multimodal → Use.

2) AI Agents.

(3) → Responsible AI — (prompting)

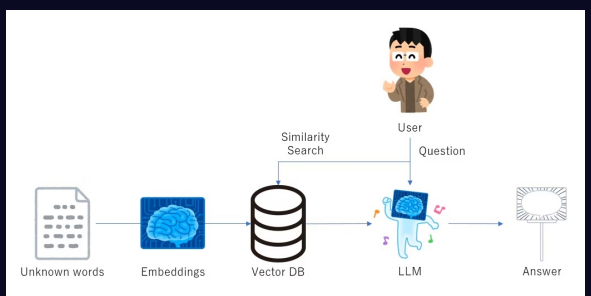
(4) → Personalised Model →
- ↳ Finetune
- ↳ RAG
- ↳ prompting

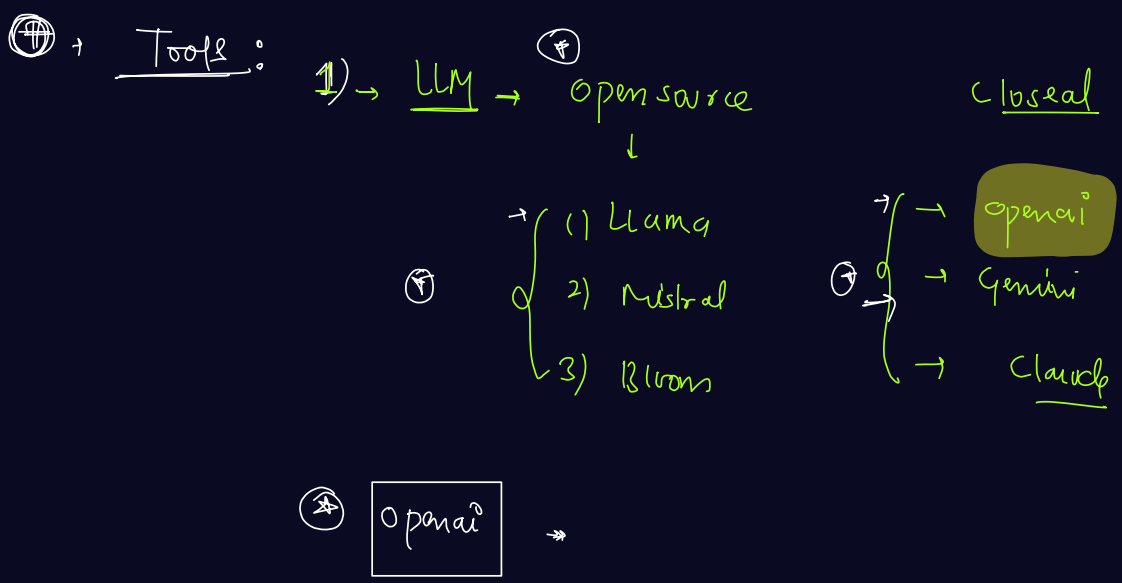
(5) → Contextual window ⇒ Memory (

↓

3|-4 → M m

Buffrm Memor / Conversation.





2) Framework → Orchestration Framework →

1) langchain → Rest
2) openai → openai
3) Llamaindex → AI Agent
4) Huggingface → RAG

embed, vector, partition

7) Vector Database =>
Pinecone
Chromadb
FAISS
4

Weekend ↓

Level 2 → Frontend Server → 1) Fast API

2) Streamlit

3) Flask

Level 3 → 1) Backend → 1) Fastapi

2) Langchain

2) Cloud Storage ⇒ 1) S3 bucket → AWS

2) GCP → Vertex (Bucket)

(3). Deployment → Github Actions.

Circle CI

Jenkins.

Docker / Contain.

(5) Multi Agent → 1) Crew AI — Langgraph

2) Langgraph.

3) Autogen.

7 LLMOps ⇒ (1) Aws-BedRock

(2) Vertex AI

(4) Langsmith

(6) Models — 1) API →

2)

3) Download →

Aws Bed / vertex → Connect → System

Llama

(8)

Interview (3)

Topic → How to Select LLM Models ?

(1) Requirments → 1) Text
2) Audio →
3) Video
4) Image

* (2) → Accuracy → Chatmodel →

(3) → Latency / Speed / → Time taken to Response

↓

Automou → Realtime / Stock

(4) → Model Size → Hardware → GPU →
Electricity pow →

↓

7 Billion 405 Billion

(5) Costing → Model → pricing.

Paid

(6) → Responsible AI → Not Biased ⇒

Bias → Election → Analysis.

Hospital →

(7) → Context window ⇒ Stock Market →

Context

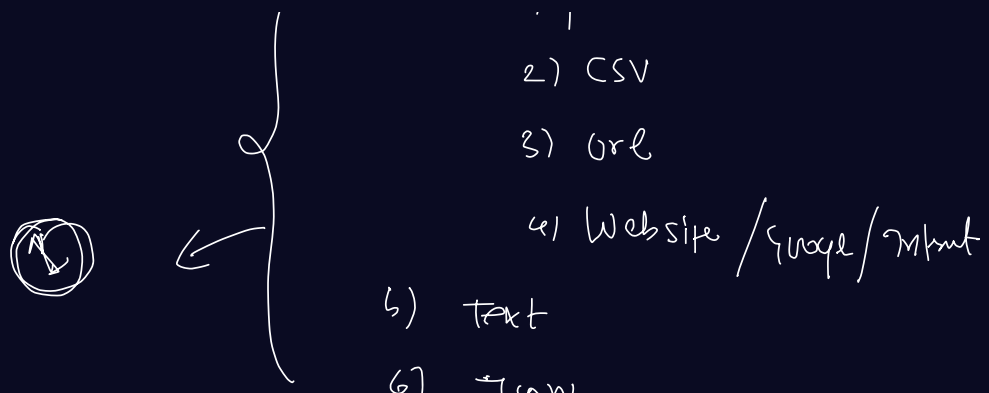
Memory

Memory →

Data preparation ⇒ RAG. → Langchain

(1) Data Loaders →

4) Pdf



(2) ⇒ Data → Text / Image → Tokens

↳ Chunks / Split → Splitter

(3) → Embedding → Vectors → (Meaning Data)

↓

(1) OpenAI / Hugging Face Embedding

(2) Text (Image / Video / Audio)

(4). Vectorstore → Pinecone / FAISS / Weavite / ...

↓

1) Cloud

2) Pinecone

3) Local

Data.

Technique : Prompting → prompt template

RAG

Finetuning

→ Langchain

1) Data → Prod.

RAG → Accuracy

KB

Retrival Augmented Generated

Stage — Combine — Generate.

+

Accuracy

Fusion.

+

LLM

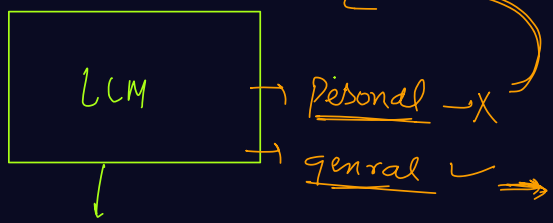
RAG

What:

1) What is RAG
2) Why ?
3) Importance ?
4) Use cases. ?
5) Components ?
6) Phases
7) Benefit
8) Challenges
9) Limitations

* Implemation
  - 5-6

(1) → Why - ?



Q → query → I don't like

↳ (1) Personal Information

(2) Database

(2) → India vs Australia → Who win Second Match →

Real time

Websearch ⇒ Websearch

Share data

↓

Update

(3) • ① Hallucation →

↓

You know.

don't

② → RAG.

↓

① From

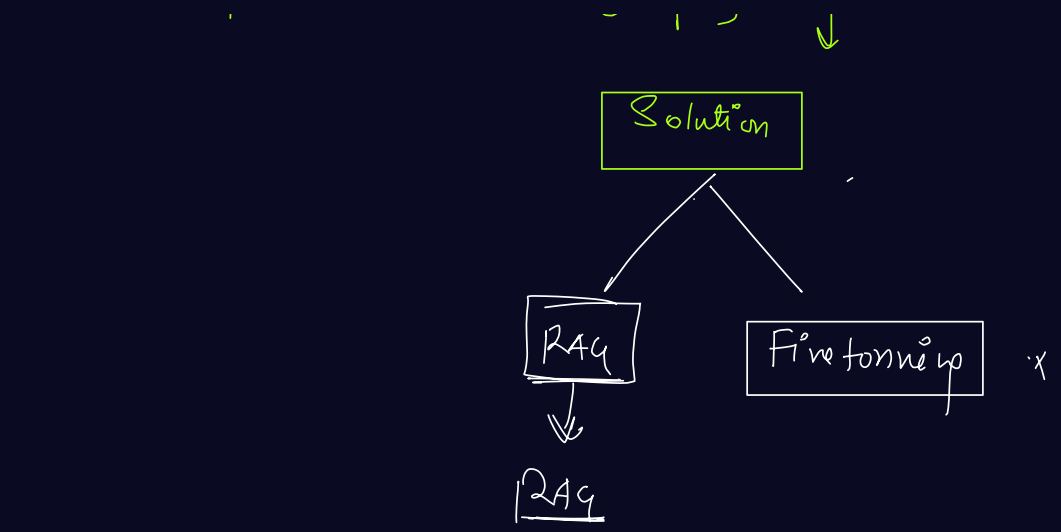
② w

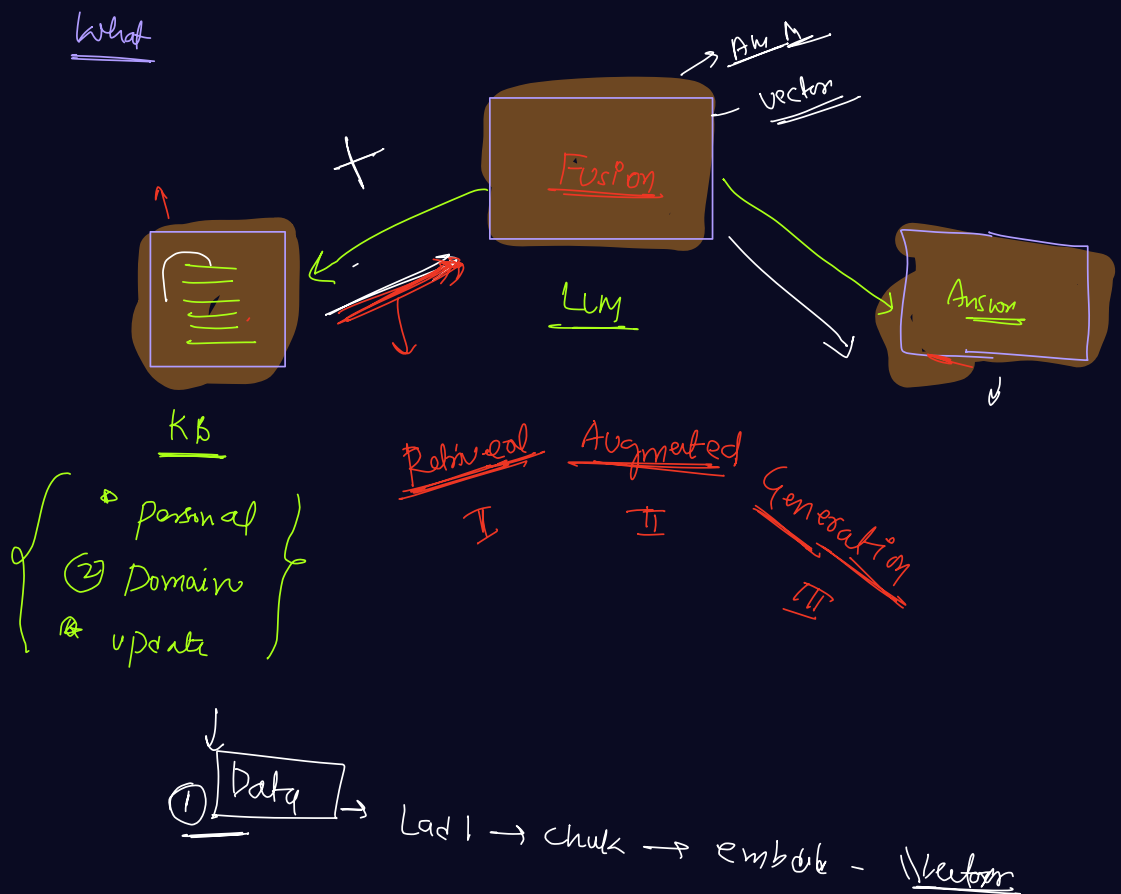
(4) • Domain Specific Models →

Medical ← Hospital →

Data → Attention → Medical

Problem → General (Wallpost) →


Solution
RAG
Fine tuning
RAG


What
Augmented
Vector
Fusion
LLM
Answer
KB
Retrieval
I
Augmented
II
Generation
III
Personal
Domain
update
Data
Load → chunk → embed - Vector

Agenda · 15 Dec →

1) RAG
2) Vs code setup.
3) Chat Model / Completion
4) Data Loaders
5) Prompt Engineering.

Task → langchain / openai → Explore

Task - 2 → 5 data Loaders.

Task - 3 → RAG → limitation ( )

↓

Advance RAG →

Solve → Research — (1)

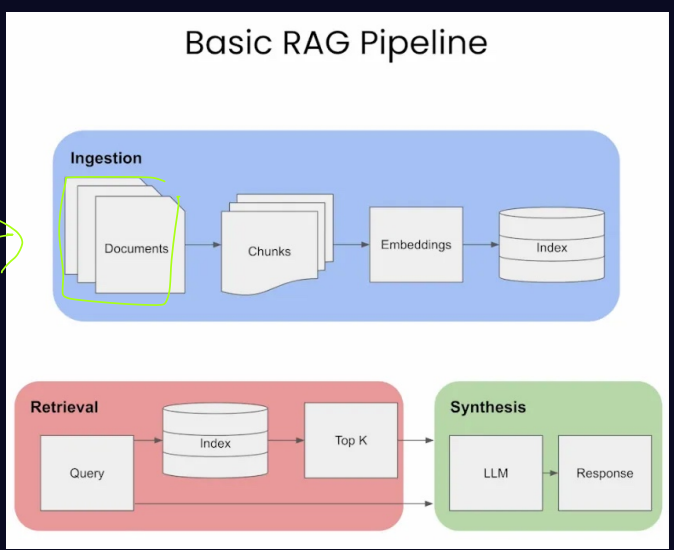
Implement → 2

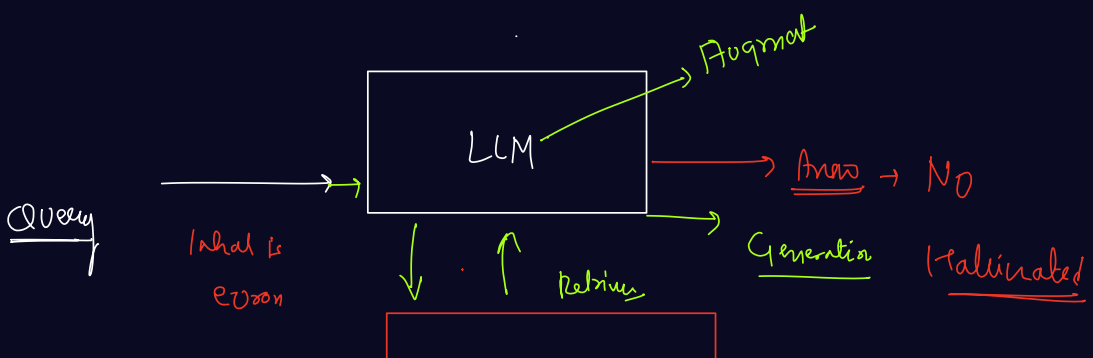
RAG → What ?

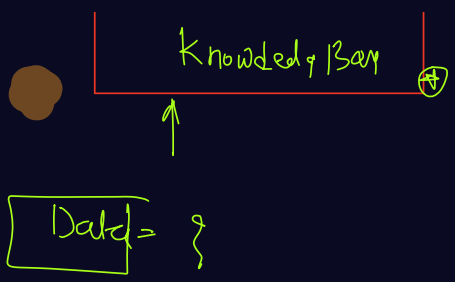
How ?

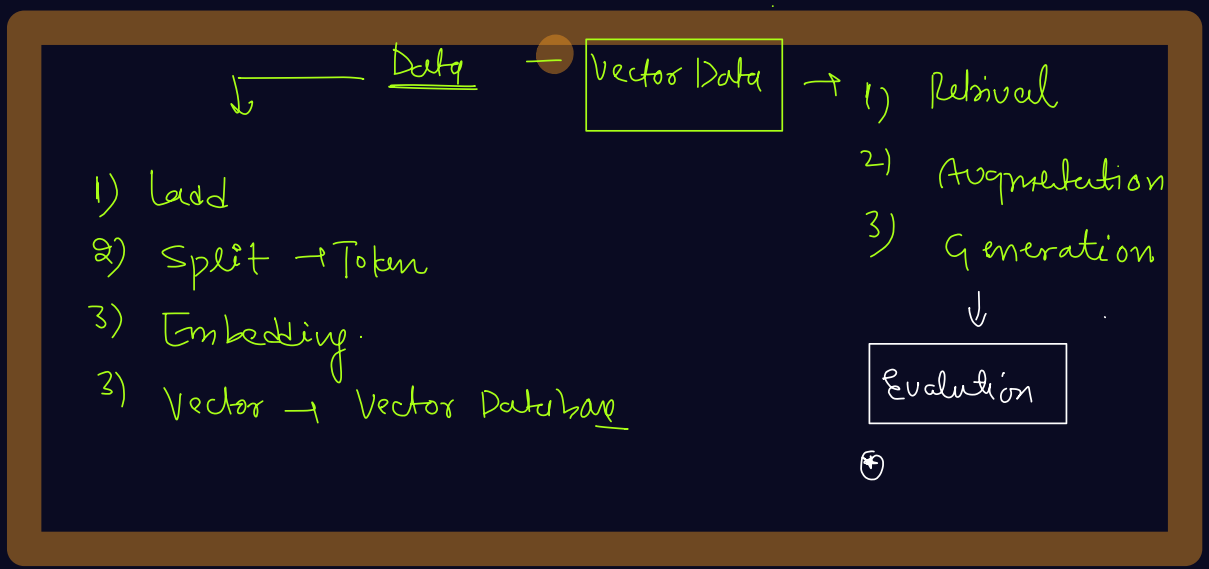
Benefit →
1) Access to personal data
(2) Make it update Model
(3) It Become Domain Specific
(4) Reduce Hallucination

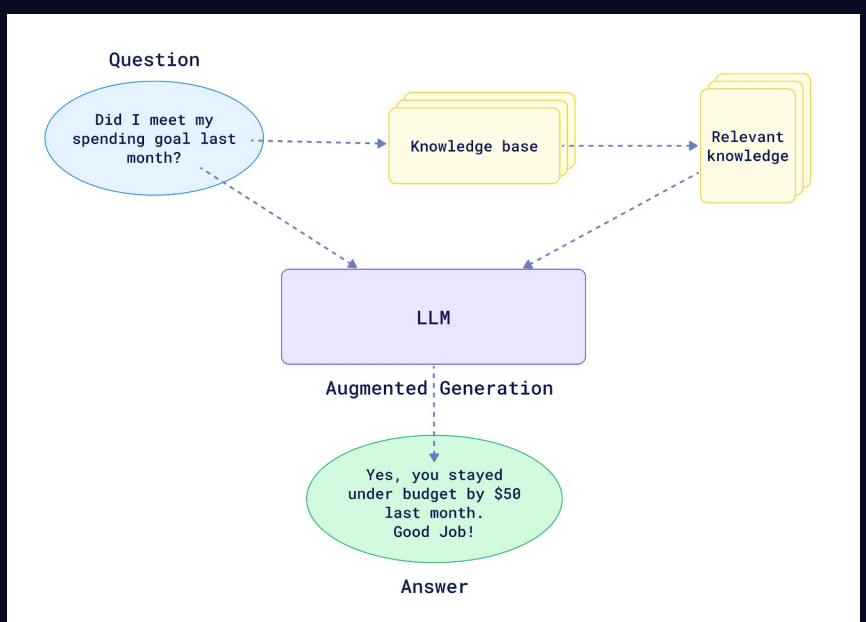
Rag → Query
Retrivall
Generation
→ stage →

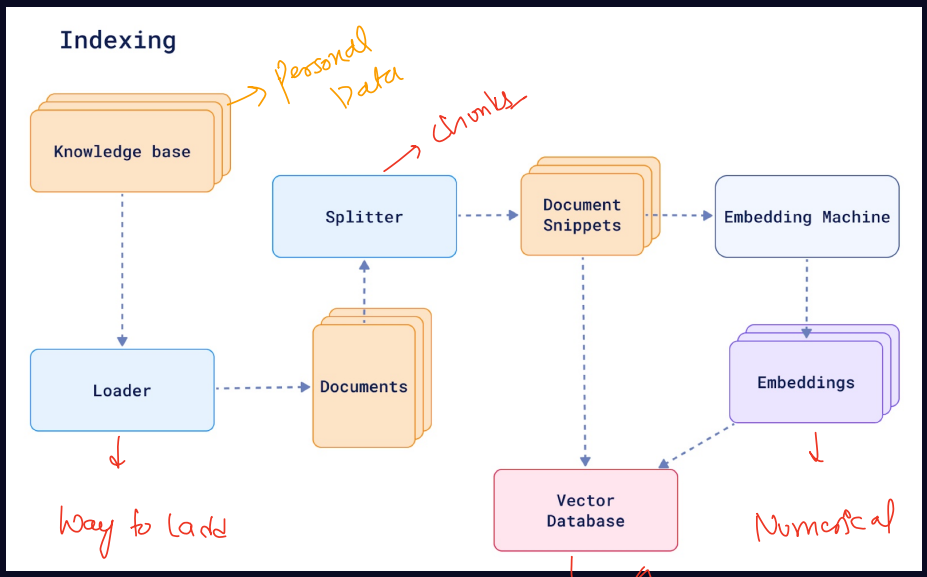





Knowledge Base
Data


Data
Vector Data
1) Retrival
2) Augmentation
3) Generation
Evaluation
1) Load
2) Split → Token
3) Embedding.
3) Vector → Vector Database


Question
Did I meet my spending goal last month?
Knowledge base
Relevant knowledge
LLM
Augmented Generation
Yes, you stayed under budget by $50 last month. Good Job!
Answer



Personal Data

Chunks

Way to Load

Numerical

MYSQL

Relational

format + Meaning

Vector

* Vector store =

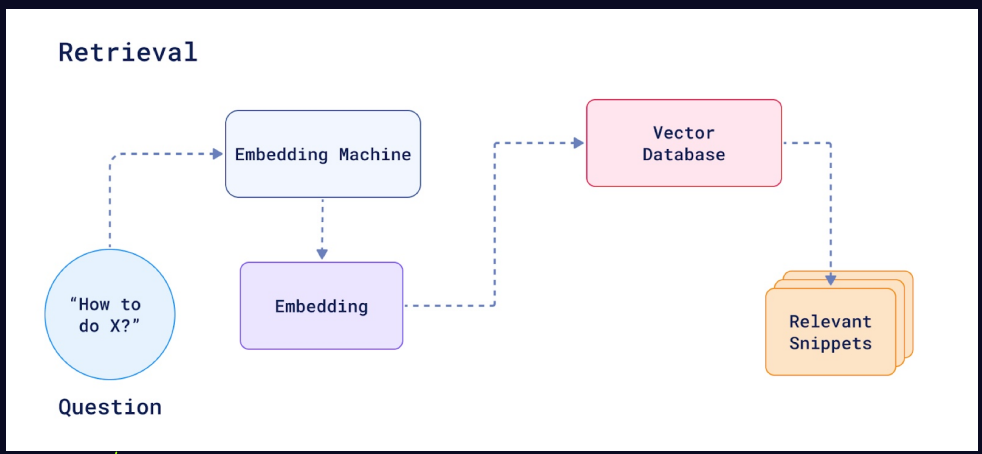
Vectorstore

Model → Trained

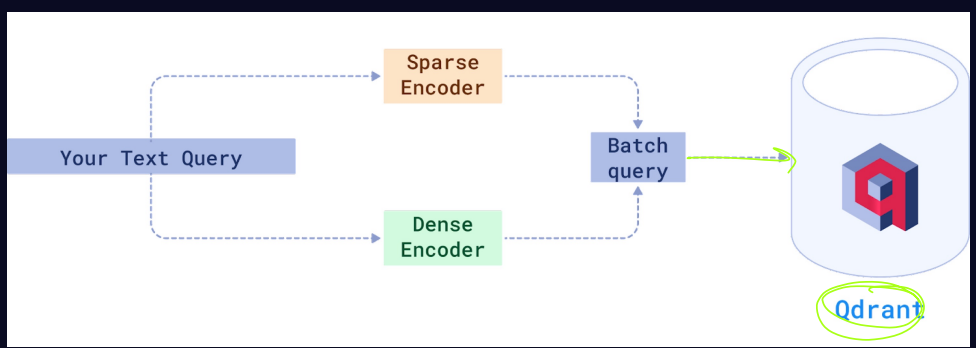
Vector

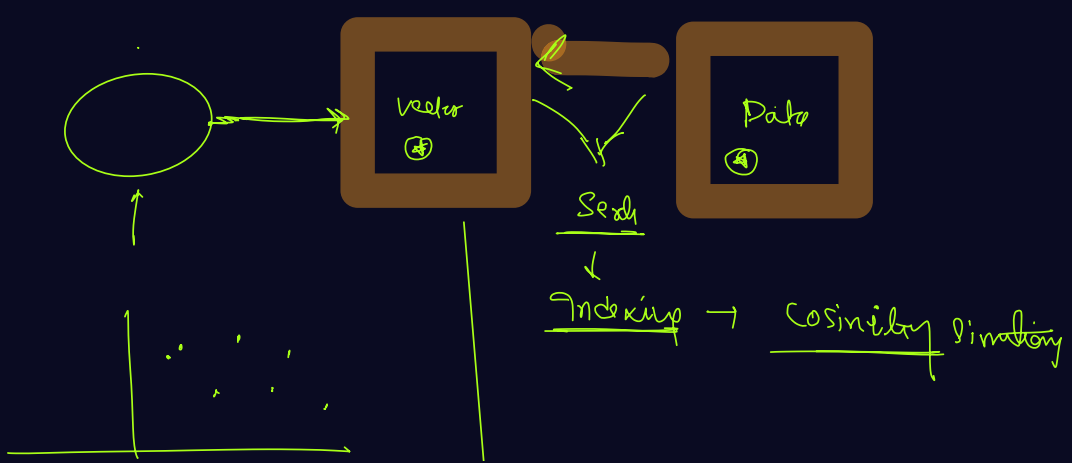
Parameter Trained

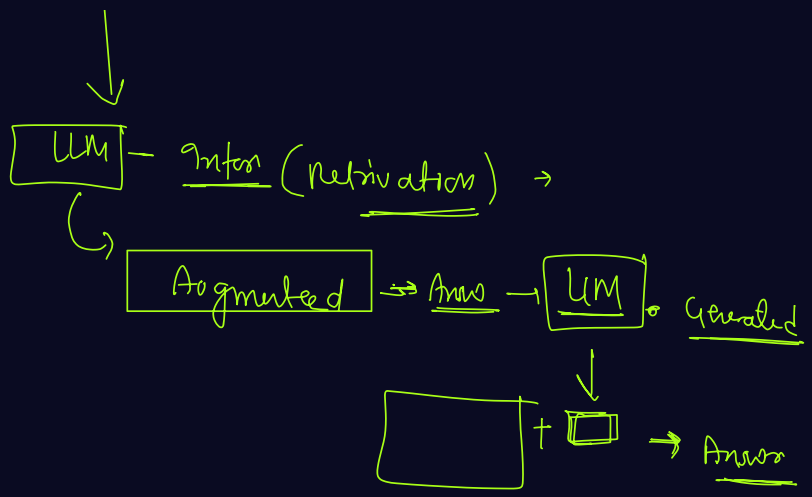
1) Pinecone
2) Weaviate
3) Faiss
4) Chromadb

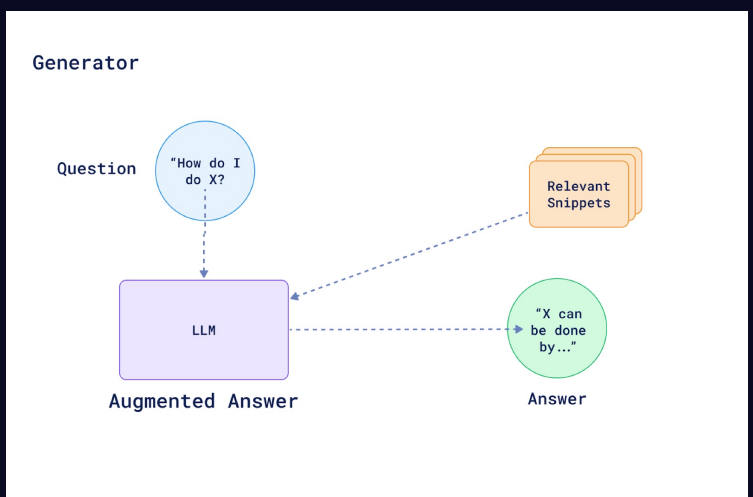


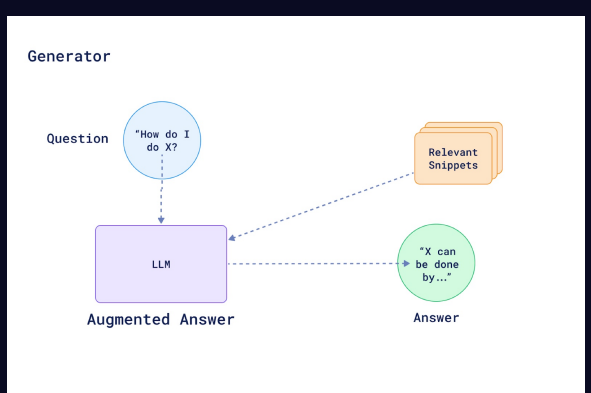
What is Qdrant ? ① Convert into vector

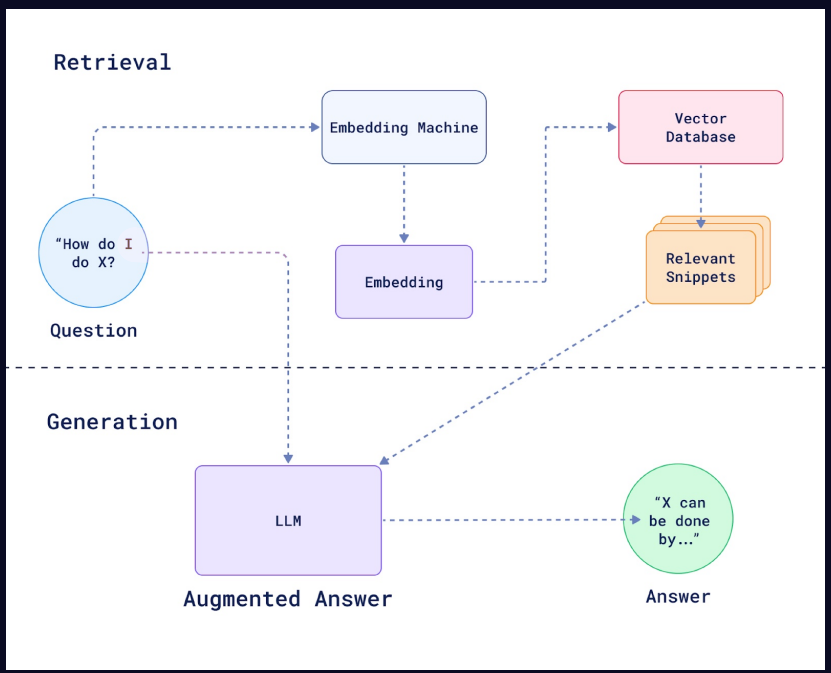





LLM
Augmented
LLM
Answer


Generator
Question
"How do I do X?
Relevant Snippets
LLM
"X can be done by..."
Augmented Answer
Answer


Generator
Question
"How do I do X?
Relevant Snippets
LLM
"X can be done by..."
Augmented Answer
Answer



1) Data loader
2) Retriever
3) Embeds

1) Vector store
2)
3)

→ coding

⭐ Challenges Or Limitations ⇒

1)

1) Accuracy → Depend Database
2) Vector Datas → Storage
3) Realtime → update Difficult

(4).

Advance RAG →

1) Dense Retrieval & Hybrid Search
2) Reranking. Reranking.
3) Query Expansion
4) Context Distillation

Problem ⇒ ① Start [ ] End

X [lost in middle]

② Insufficient information Compression

③ → Context window issue → (ST)

- Task -1 (1) Research Advance RAG Technique
- Task -2 (2) Implement RAG Architecture
- (3) Task-3 - [5] → Model Explore

(4) → Document loader → Download in lang

(5) → langchain → github →

(6) .env → Credentials → Api → Paise

* openai = " "

Requirement = [ ] →