Retrieval Augmented Generation (RAG) with Python

Developers Conference 2024

Presented by: Nirmal Rampersand, Python User Group Mauritius

July 2024

Agenda

Large Language models (LLMS)

Limitations of LLMs

Retrieval Augmented Generation (RAG)

Technical deep dive

Live demo

The potential of RAG

Large Language Models

Large Language Models

Generative AI

Understand and generate “human-like” text

Architecture

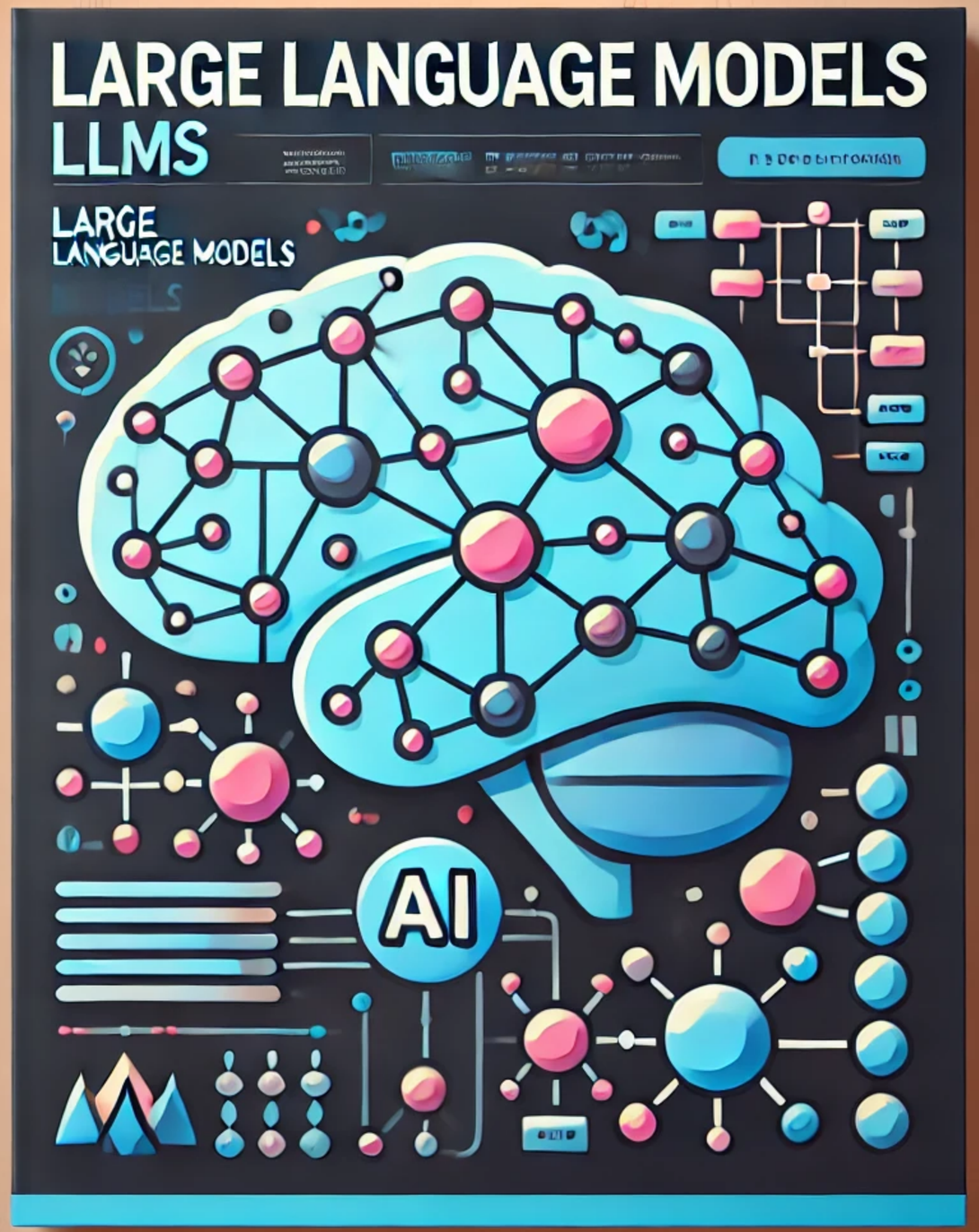
Transformer architectures with self-attention mechanisms

Models

GPT, Llama, BERT, Mistral, Phi

Applications

Translation, summarisation, question answering, document analysis, coding assistance, research, customer service, etc



What’s the key challenges in LLMs today?

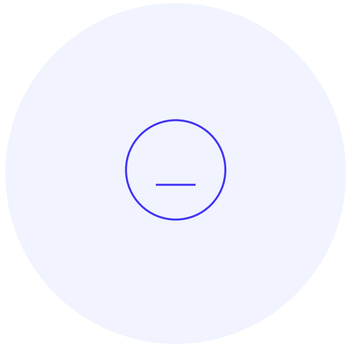
Limitations of Large Language Models

1/3

Hallucinations

2/3

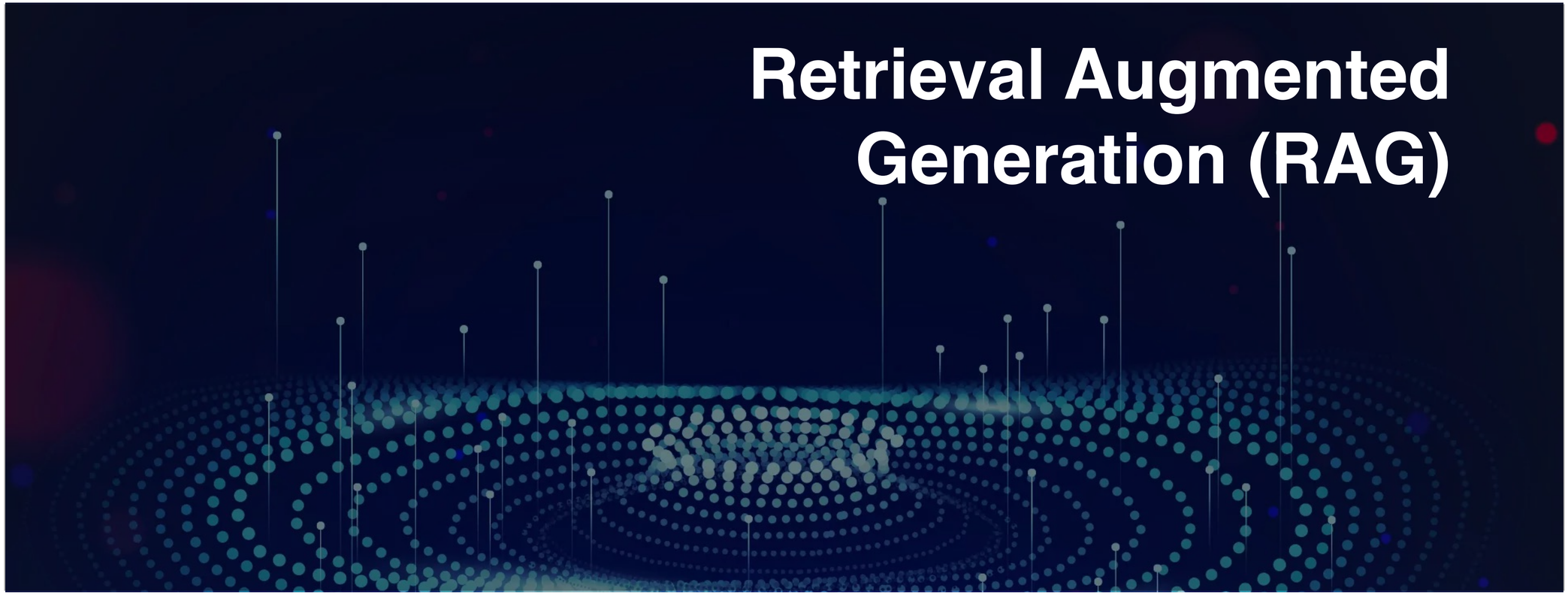
Knowledge cut off



3/3

No interpretability

How do we address these problems?



What is RAG?

- Hybrid AI model that combines capabilities of traditional LLMs with real-time information retrieval

- Enhances the accuracy and reliability of generative models

- Builds trust

USING LOCAL LLMS WITH LOCAL DATA

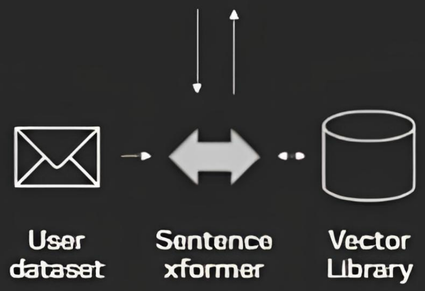
BASE MODEL

What was the name of that movie Chris emailed me about last year?

Sorry, I don't have access to any specific information about emails or ...

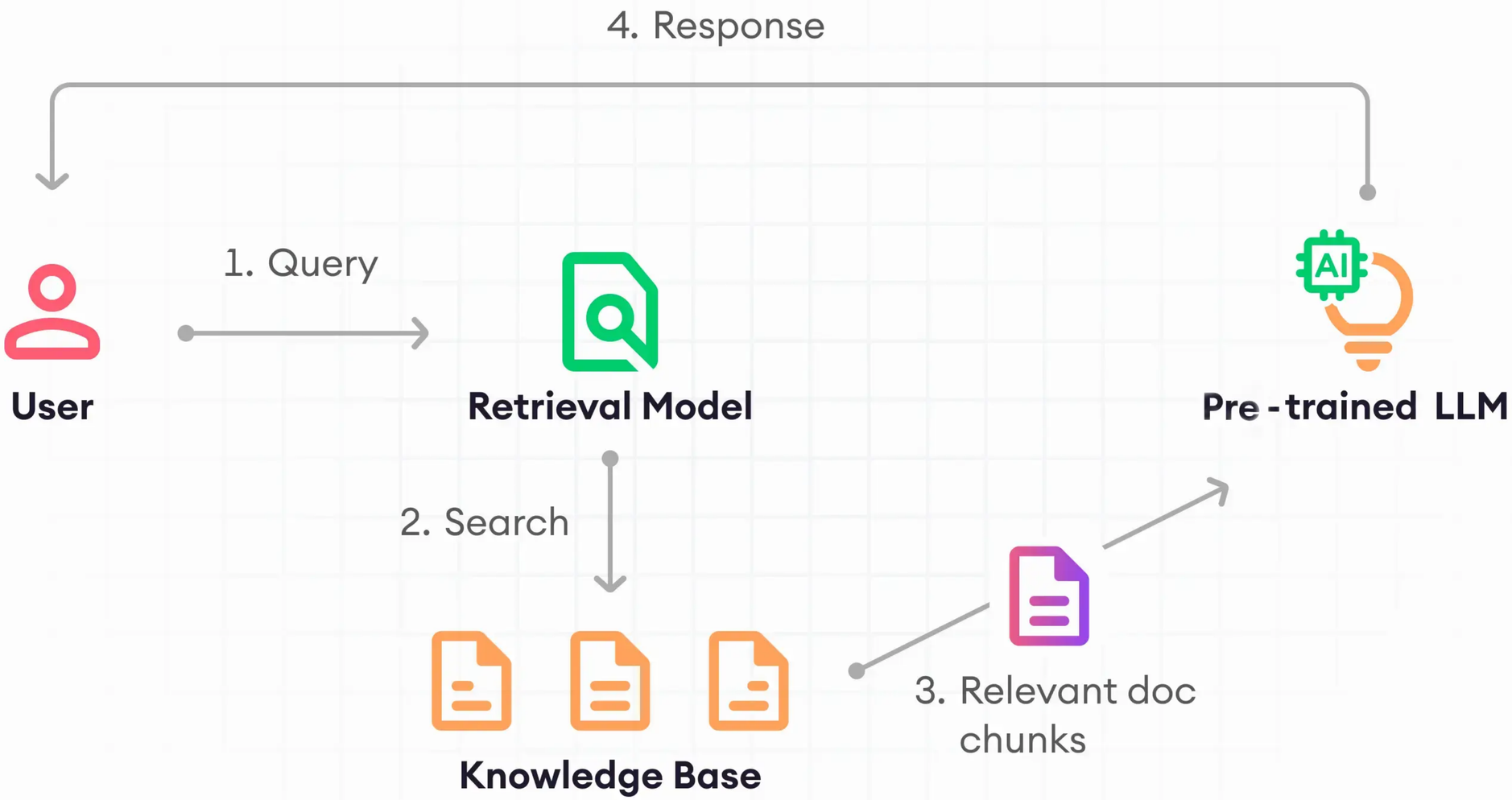
BASE MODEL + USER DATASET

What was the name of that movie Chris emailed me about last year?



Chris Thomson emailed you about "The Fall" on October 23rd last year. The Fall is an adventure fantasy film released in 2006, starring...

An overview of RAG



Why is RAG so important today?

Dynamic information retrieval

LLM has access to up-to-date contextual data based on query

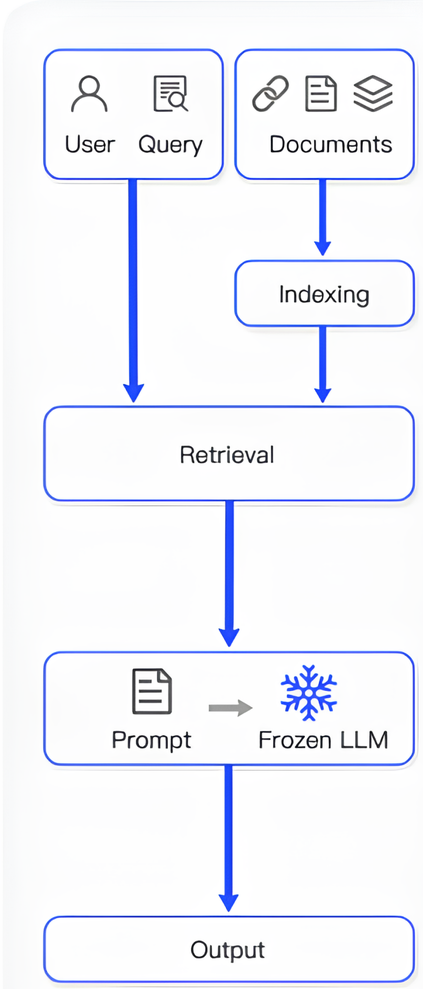
Mitigates hallucinations and knowledge cutoff

Enhanced transparency

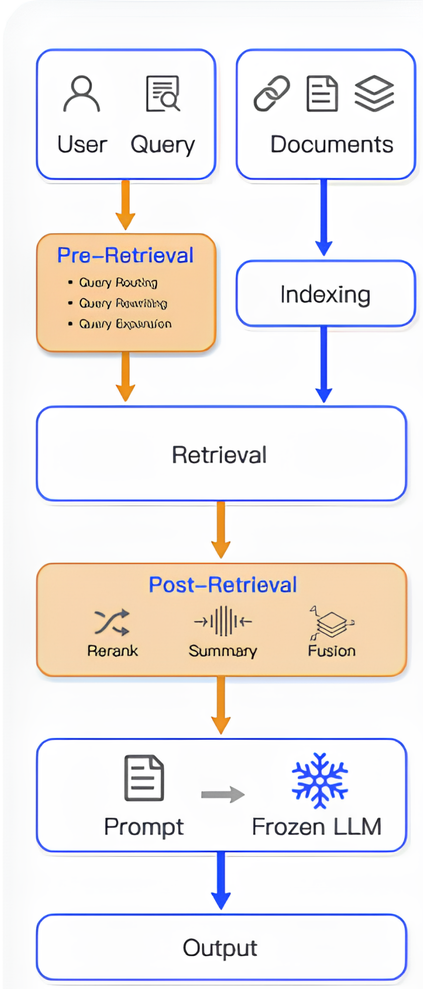
Access to the source or reference

Builds trust

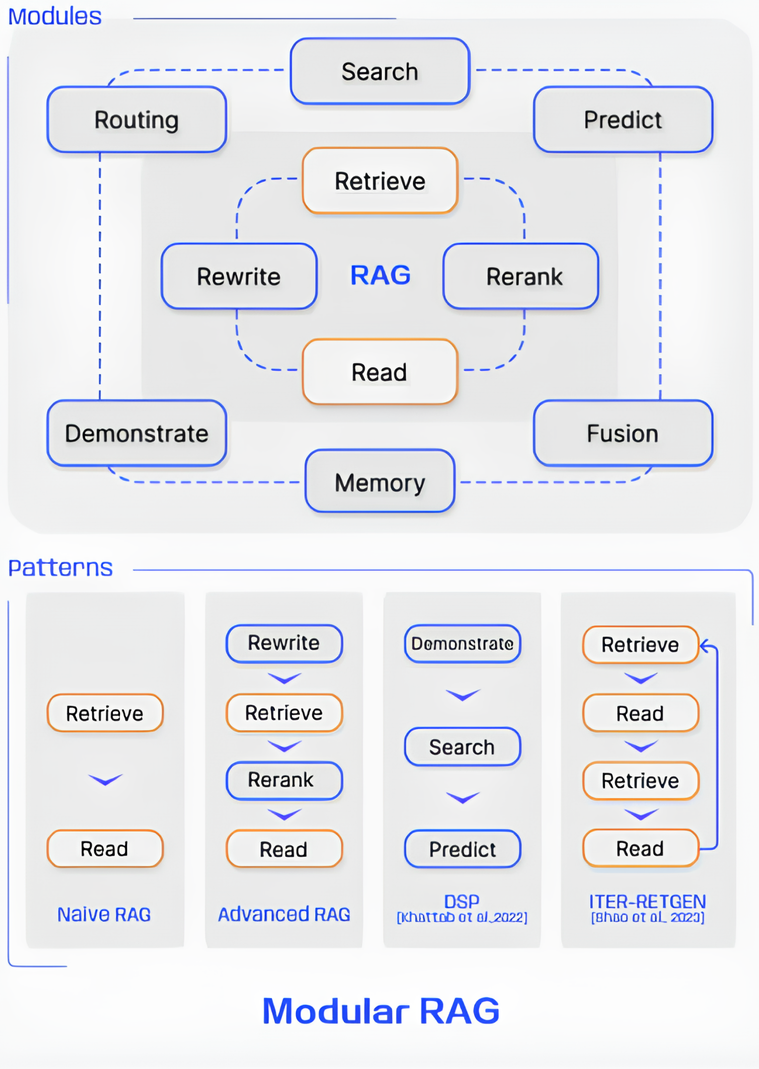
Types of RAG systems



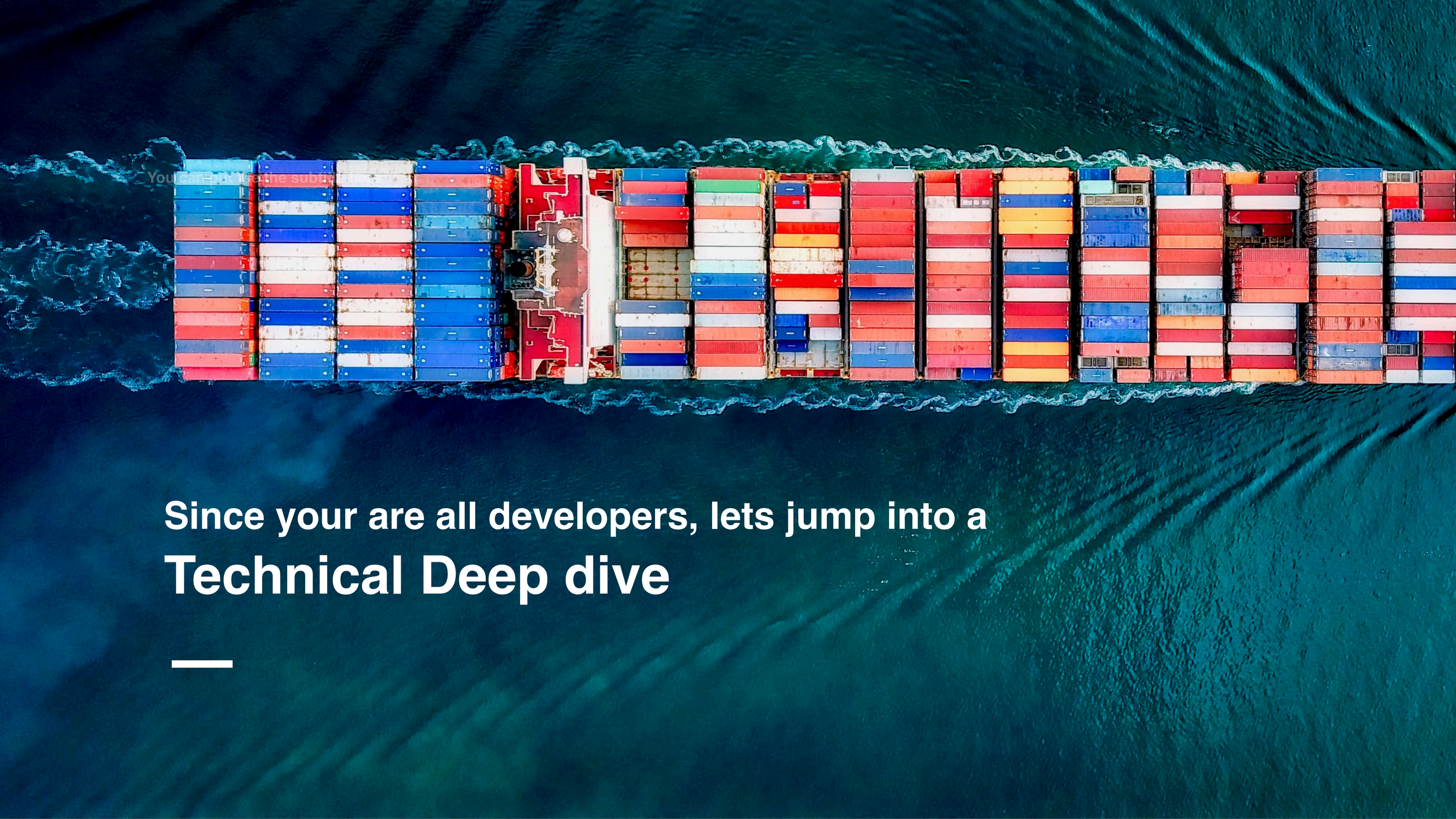
Naive RAG



Advanced RAG



Modular RAG



Since your are all developers, lets jump into a

Technical Deep dive

Main RAG Components

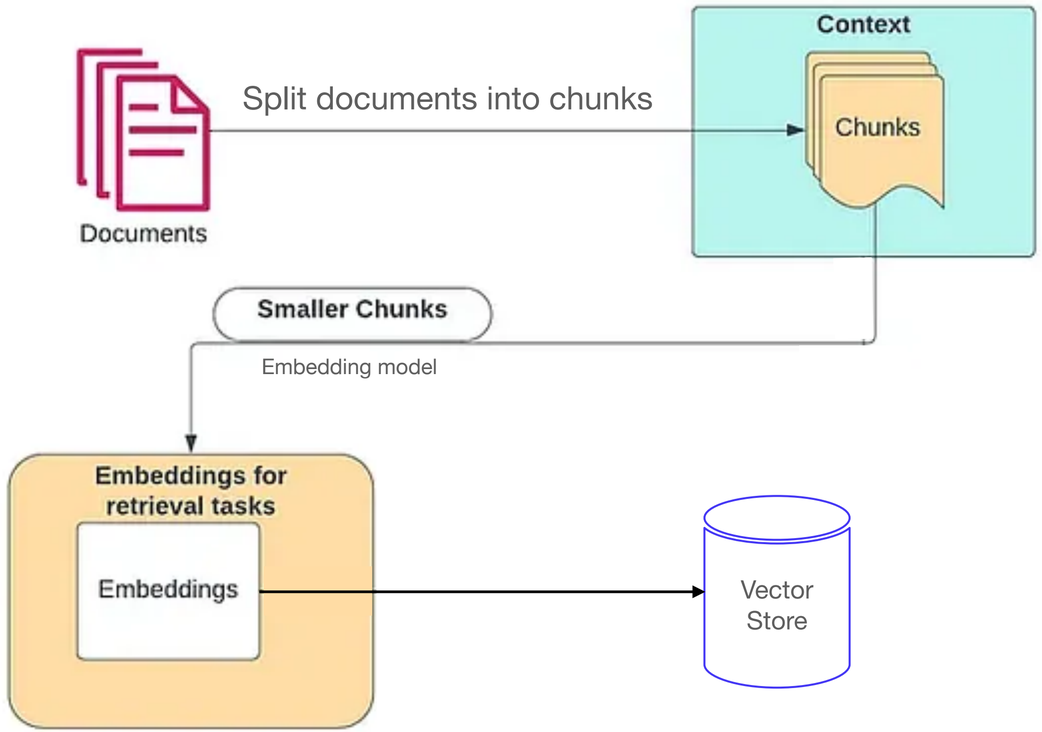
Processing documents

Query processing and information retrieval

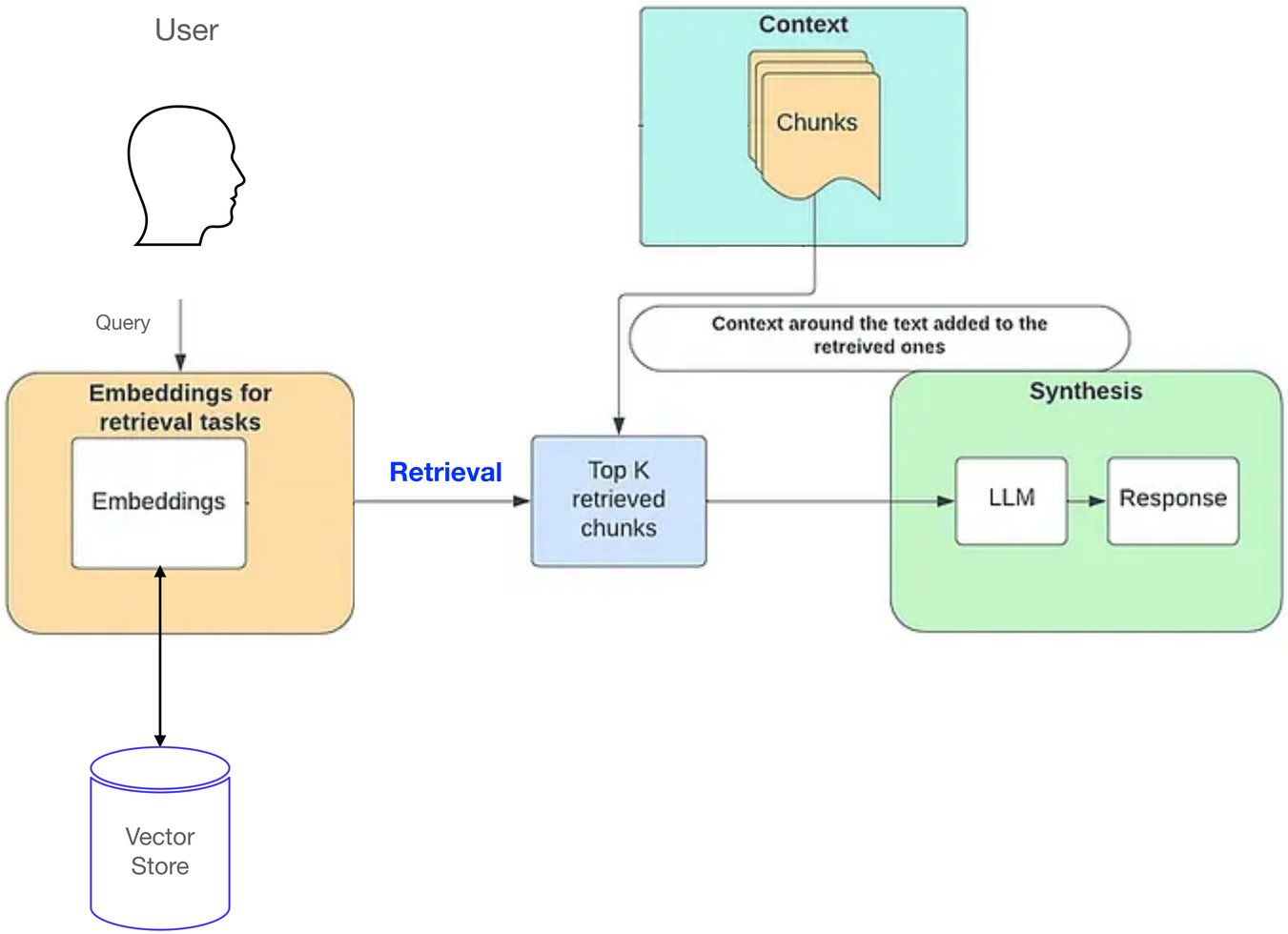
Response generation

Processing documents in a RAG

“training”



Query processing, retrieval and response generation



Retrieval techniques

Sparse retrieval

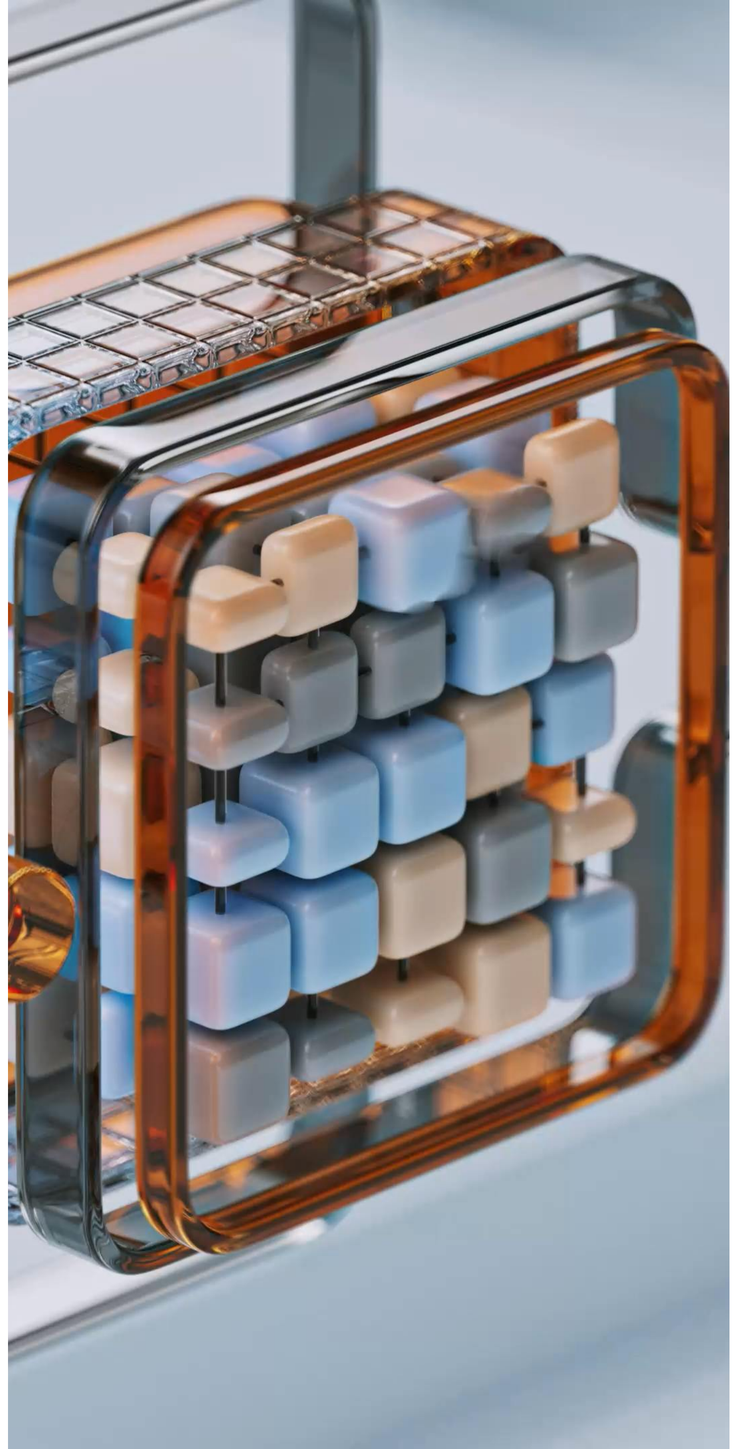
Uses keywords and tf-idf scores

Efficient but not context-aware

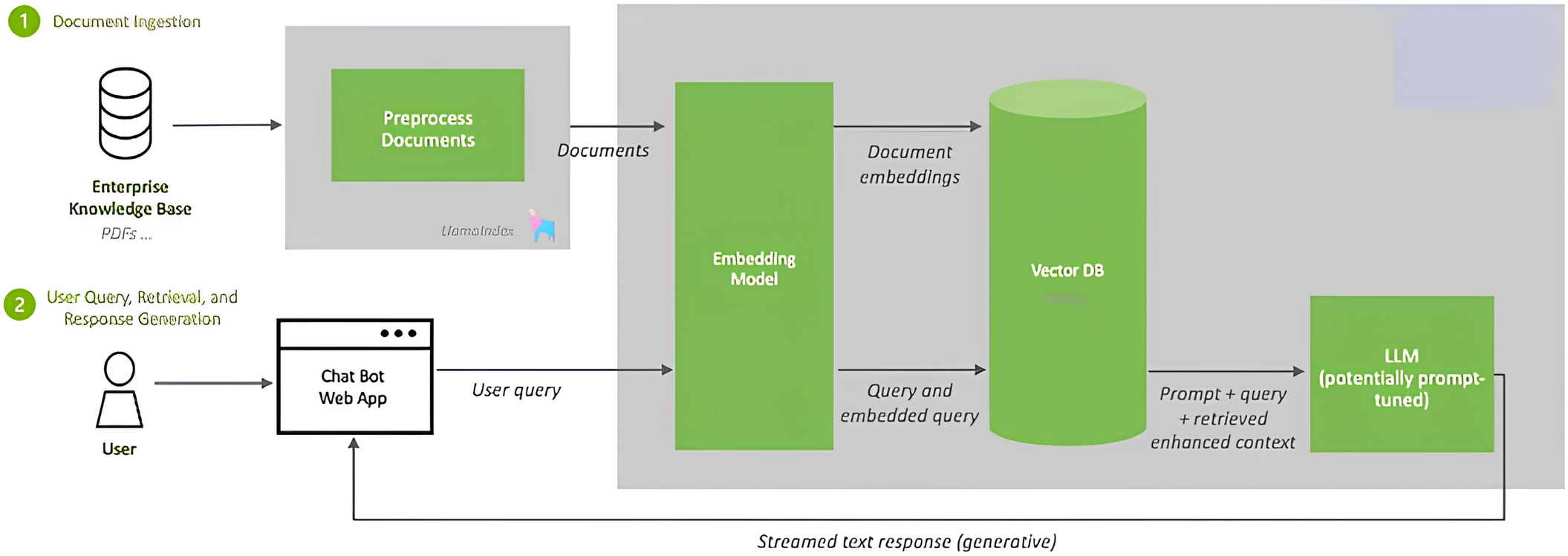
Dense retrieval

Uses ML models to get the semantic context of queries and documents

Context-aware



Simple RAG Architecture

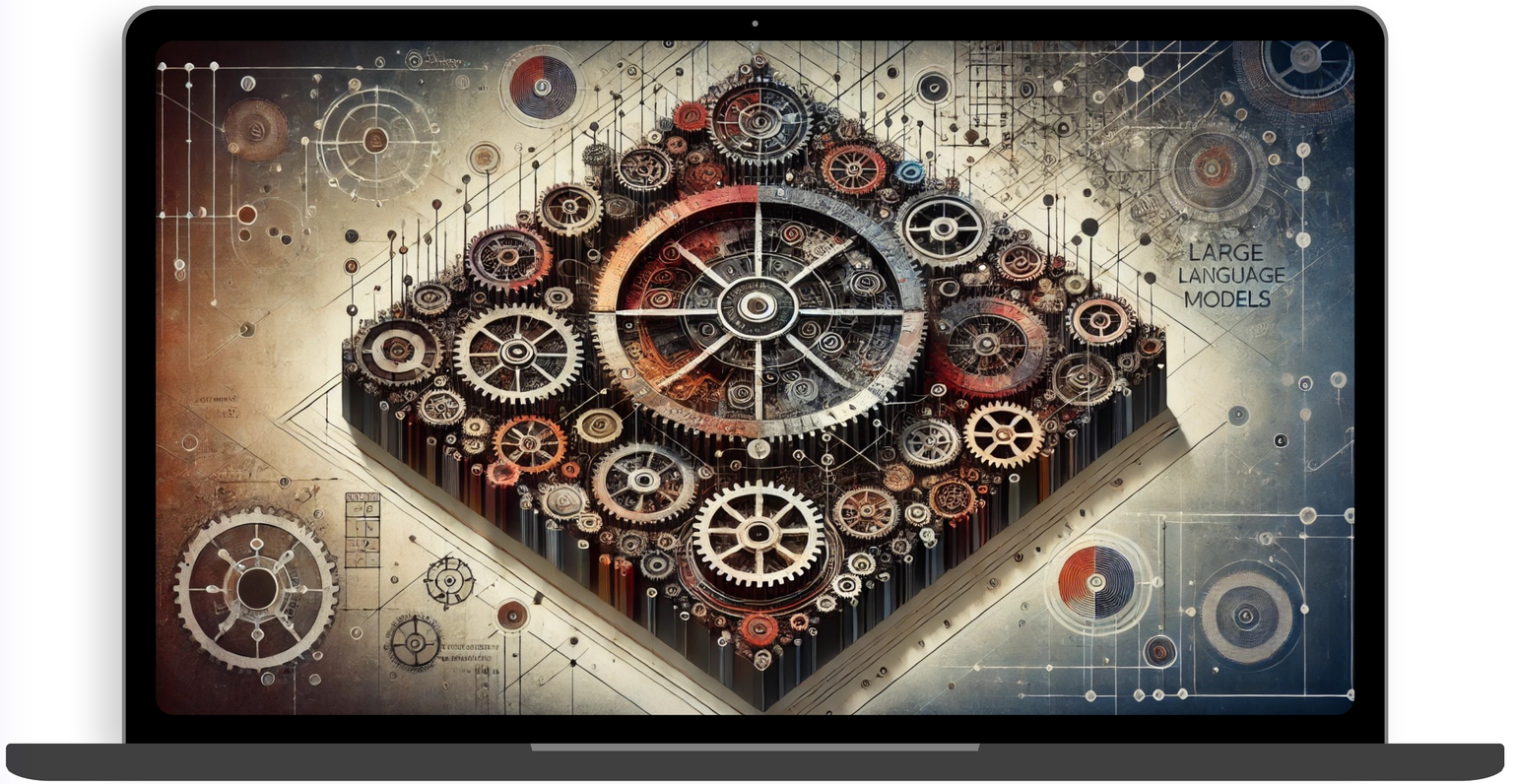


The full RAG Process

```
#the implementation process

1. Load documents from folder
2. Split documents
3. Create vector embeddings
4. Store vector embeddings
5. Get user query
6. Create vector embeddings of question
7. Conduct dense retrieval using a semantic search of vector database]
8. Filter the top 3 chunks of data according to similarity score
9. Generate promt and combine user query, context and prompt config
10. Send prompt to LLM
11. LLM will generate an accurate and reliable answer
```

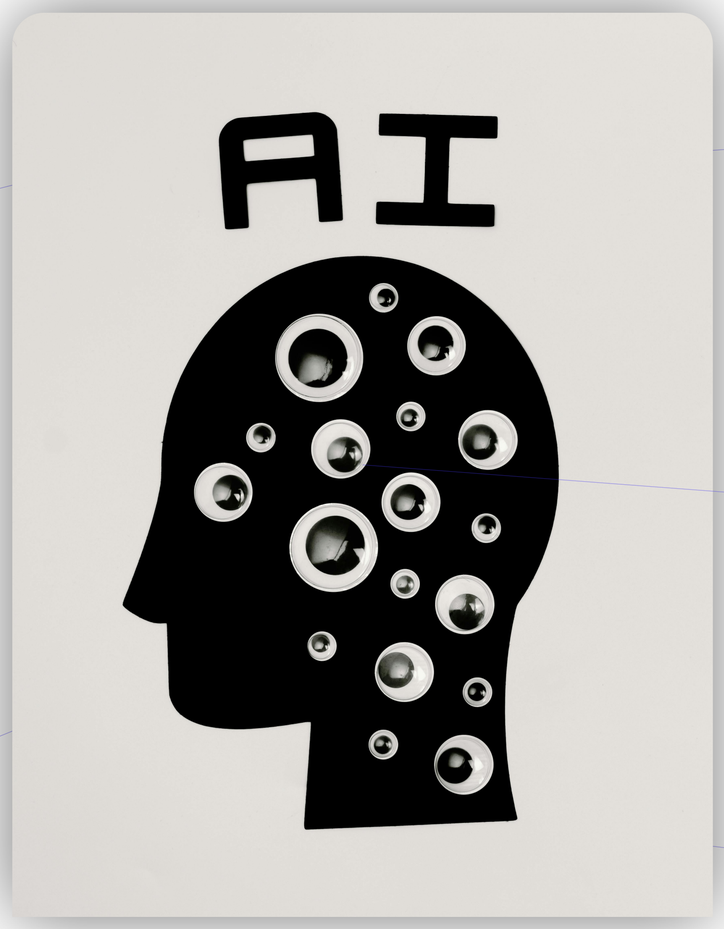
Let’s implement RAG



What do we need?

01
LLM

02
RAG framework



03
Data

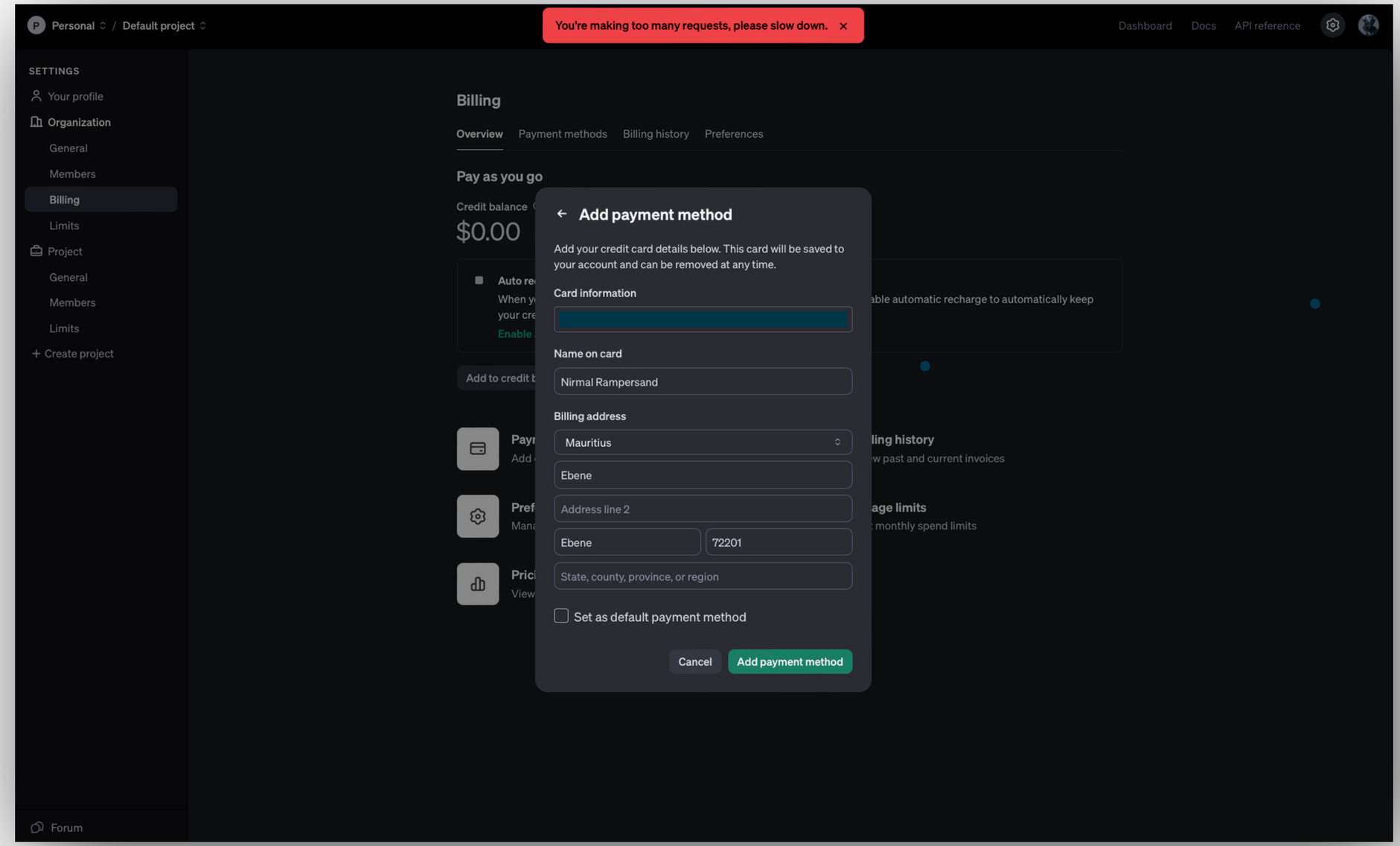
04
RAG pipeline

05
Some coffee ☕

Lets use CHATGPT as LLM

Its the best right?

Epic Fail 😌


Personal / Default project
You're making too many requests, please slow down.
Dashboard
Docs
API reference
SETTINGS
Your profile
Organization
General
Members
Billing
Limits
Project
General
Members
Limits
Create project
Forum
Billing
Overview
Payment methods
Billing history
Preferences
Pay as you go
Credit balance
$0.00
Add payment method
Add your credit card details below. This card will be saved to your account and can be removed at any time.
Card information
Name on card
Nirmal Rampersand
Billing address
Mauritius
Ebene
Address line 2
Ebene
72201
State, county, province, or region
Set as default payment method
Cancel
Add payment method

A Local LLM 😆

“Ollama”

Get up and running with large language models.

Run Llama 3, Phi 3, Mistral, Gemma 2, and other models. Customize and create your own.

Download ↓

Available for macOS, Linux, and Windows (preview)

Tools and technologies

So, what did I use?

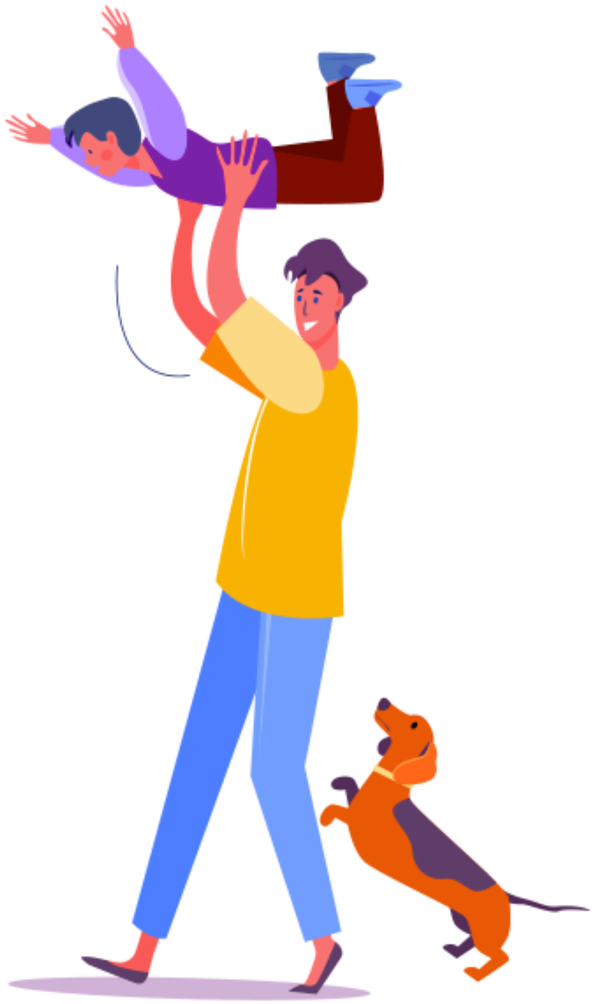
- **Language: Python** 💥
- **LLM: Ollama**
- **Model: Llama 3 - 7b**
- **Rag framework: Langchain**
- **Data: A biography about Paul Graham (text)**

FYI 🤦 Another very capable RAG library is llama-index

Lets look at some code code 👾

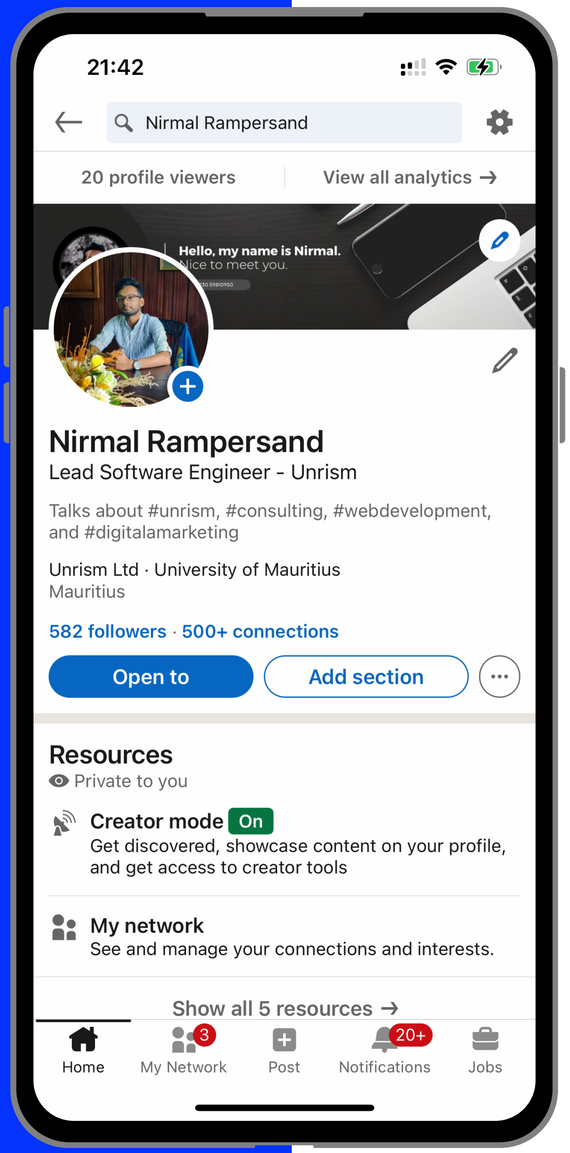
SUCCESS :)

Not only can we now run local LLM’s, but we can also generate accurate and traceable responses!



Some real world applications of RAGs

- Customer support
- Academic research
- Healthcare and diagnostics
- Documentation and development
- Legal document analysis
- Financial analysis



Thank you for your attention!

Nirmal Rampersand

+230 59810950

nirmal@coderfaculty.com