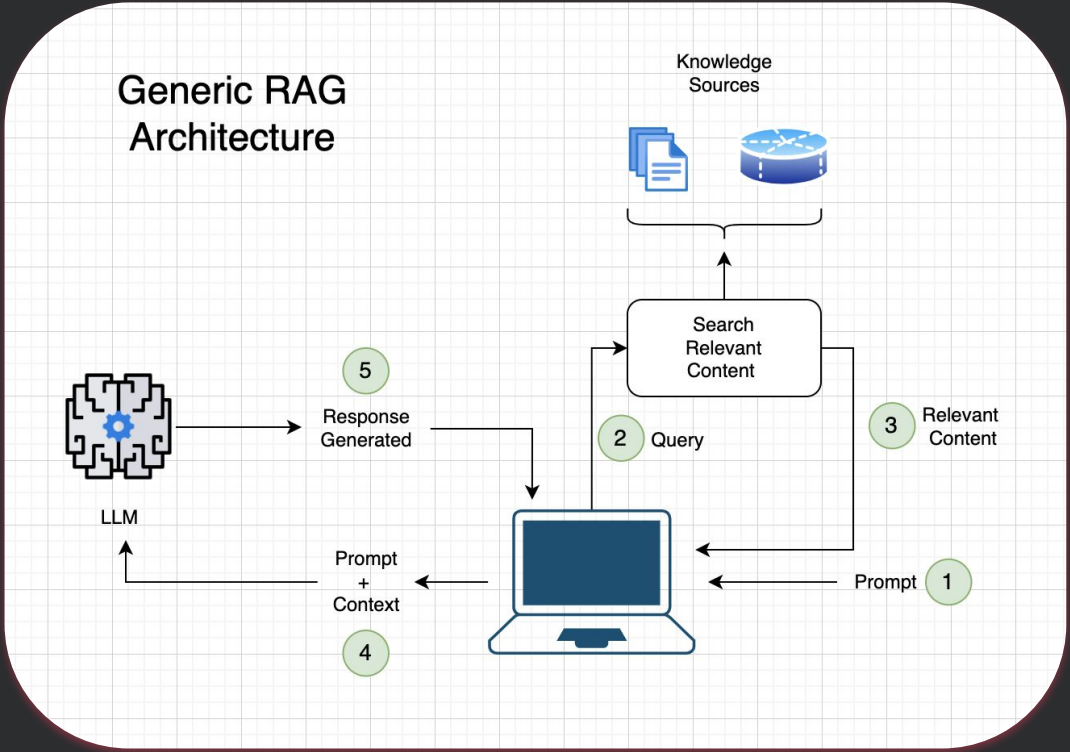
May 2024

Privacy – Aware Enterprise AI with RAG Architecture

Lucian Gruia

What is RAG



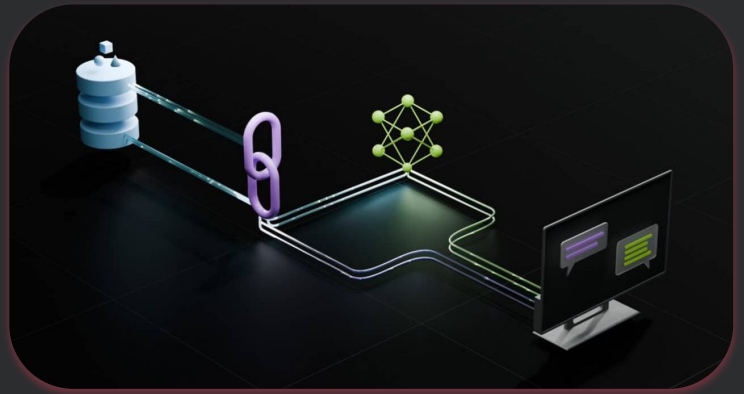
A RAG system essentially **correlates** a user's **prompt** with a relevant data chunk. It does this by identifying the **most semantically similar** chunk from the knowledge base.

This chunk then becomes the **context** for the prompt.

When passed to the **Large Language Model (LLM)**, it enables the system to provide a relevant answer within the given context.

Why do we need RAG

- **Expands Knowledge Base**
  RAG accesses a vast external database, enriching its knowledge beyond initial training data
- **Improves Accuracy**
  Enhances response precision by integrating relevant, real-time information
- **Adaptable**
  Effectively handles novel and niche queries
- **Increases**
  Streamlines information retrieval and generation process
- **Versatile**
  Useful across various fields, from customer support to research



Source: What Is Retrieval-Augmented Generation, aka RAG?

Benefits of RAG

1. **Providing up-to-date and accurate responses:**
   RAG ensures that the response of an LLM is not based solely on static, stale training data. Rather, the model uses up-to-date external data sources to provide responses.

2. **Reducing inaccurate responses, or hallucinations:**
   By grounding the LLM model's output on relevant, external knowledge, RAG attempts to mitigate the risk of responding with incorrect or fabricated information (also known as hallucinations). Outputs can include citations of original sources, allowing human verification.

3. **Providing domain-specific, relevant responses:**
   Using RAG, the LLM will be able to provide contextually relevant responses tailored to an organization's proprietary or domain-specific data.

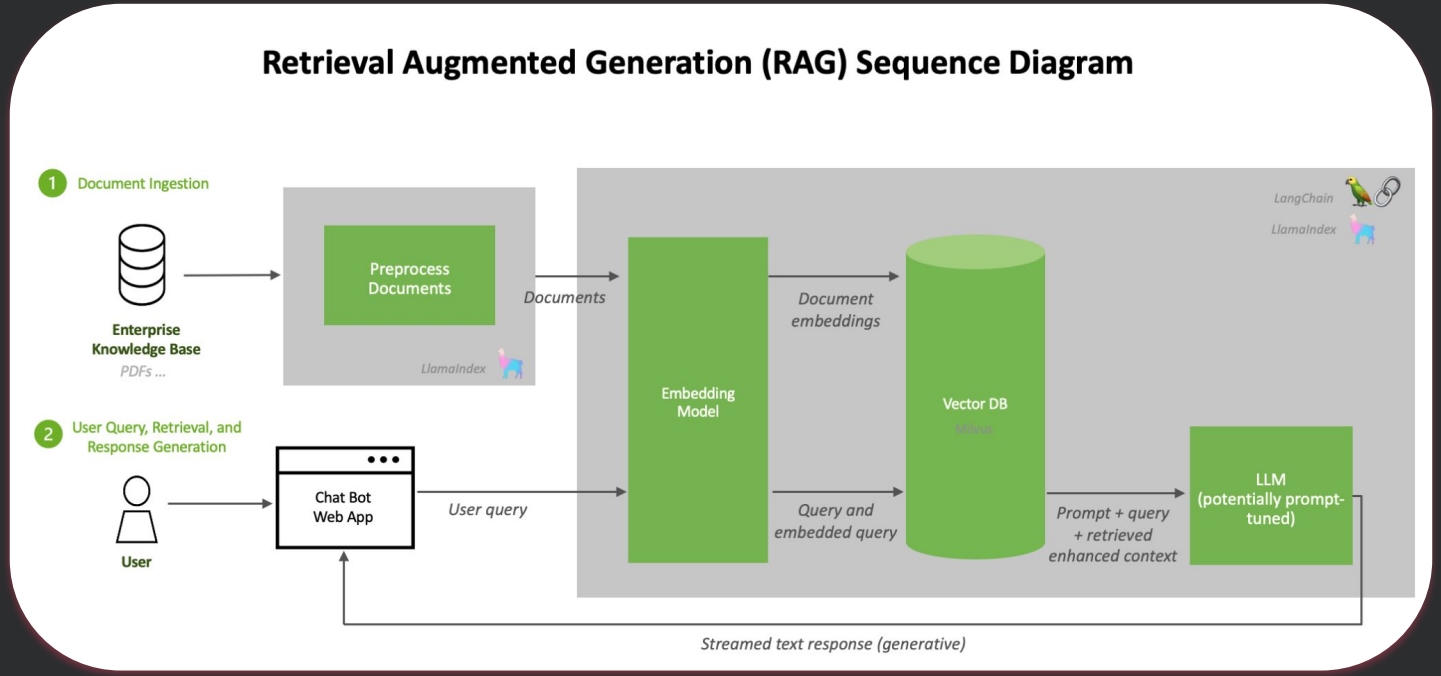
4. **Being efficient and cost-effective:**
   Compared to other approaches to customizing LLMs with domain-specific data, RAG is simple and cost-effective. Organizations can deploy RAG without needing to customize the model. This is especially beneficial when models need to be updated frequently with new data.

RAG vs Model fine-tuning

Generated with DALL·E 3

- **Strengths of RAG**
  Excelling in dynamically pulling in diverse, real-time data for rich, context-aware responses in conversational AI.
- **Key Benefits of Fine-Tuning**
  Particularly effective for customizing pre-trained models to niche applications or unique datasets, ensuring high accuracy in specialized scenarios.
- **RAG can complement fine-tuned models**
  by providing external context, while fine-tuning ensures base model accuracy on domain-specific data.
- RAG and Model fine-tuning **are not mutually exclusive**.

RAG Architecture



Source: RAG 101: Demystifying Retrieval-Augmented Generation Pipelines

Data Chunking and LLMs

LLMs also have a limited capacity for context.
Just as humans **cannot digest unlimited context,** these models have a specific size limit for the content they can process.

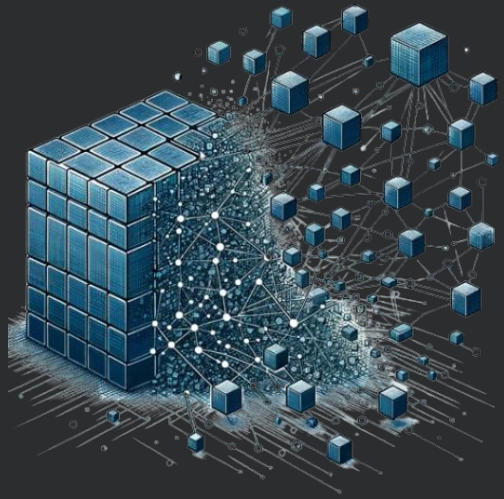
So, what about situations involving very large amounts of data?
Consider a specific use case, such as a book. It's too large to pass the entire book as **the context** for the current prompt, so it **needs to be divided** before being stored in the database.

This process is known as **data chunking**.

Types of Data chunking (by size)

- Fixed-size
- Variable Chunking
- Semantic Chunking

Read more: Data Chunking Strategies in RAG



Generated with DALL·E 3

Embeddings. Similarity

- **Embeddings**
  Numerical representations of concepts, in a high-dimensional space, capturing semantic meaning.
- **Similarity:**
  - Lexical: entities are alike in appearance
  - **Semantic:** entities are alike in meaning
- In RAG we represent entities **by describing** them.
  This is a form of **knowledge representation**.

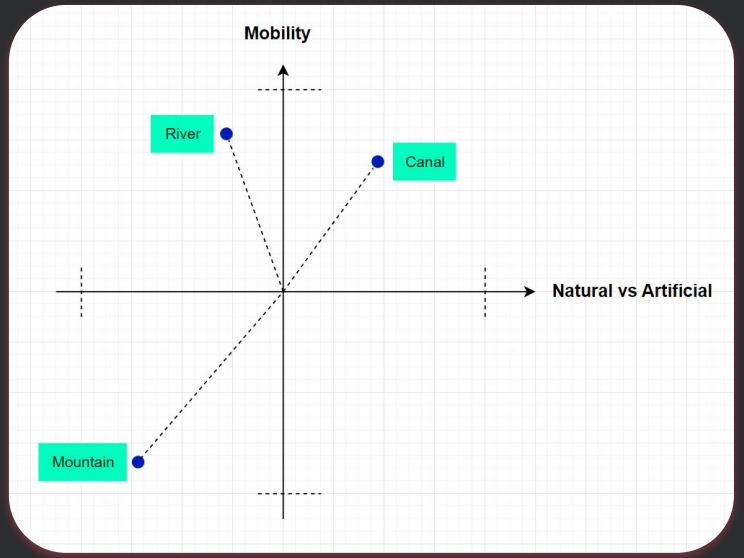
Example: Mountain, River, Canal

One hot encoding

Mountain:	1
River:	2
Canal:	3

2-Dimensional Space

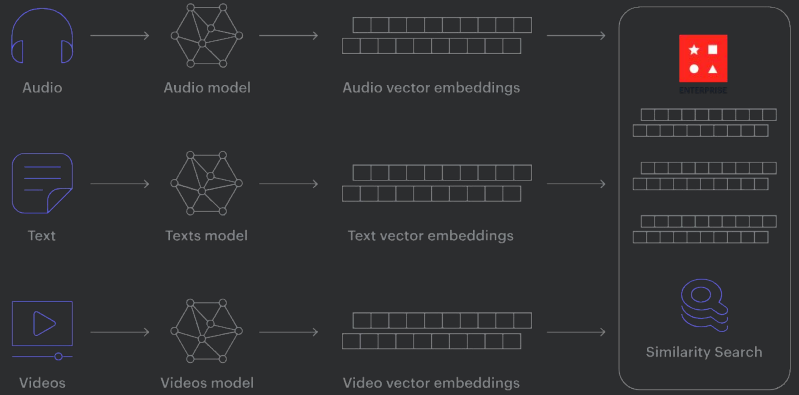
	[Natural vs Artificial, Mobility]
Mountain:	[-0.7, -0.8]
River:	[-0.3, 0.7]
Canal:	[0.4, 0.5]

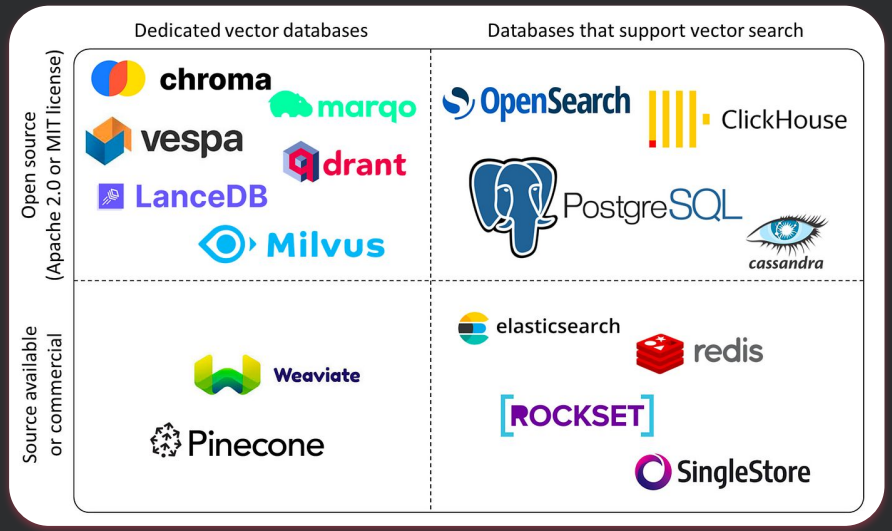


Read more: Wikipedia - Cosine Similarity

Vector Databases

- **Store, index, and manage high-dimensional vector data (aka Embeddings)**
- **Perform similarity search**
  - Cosine similarity
  - Pearson Correlation
  - Levenshtein Distance
  - Jaccard Similarity
  - Euclidean distance
  - Dot Product
  - Minkowski Distance

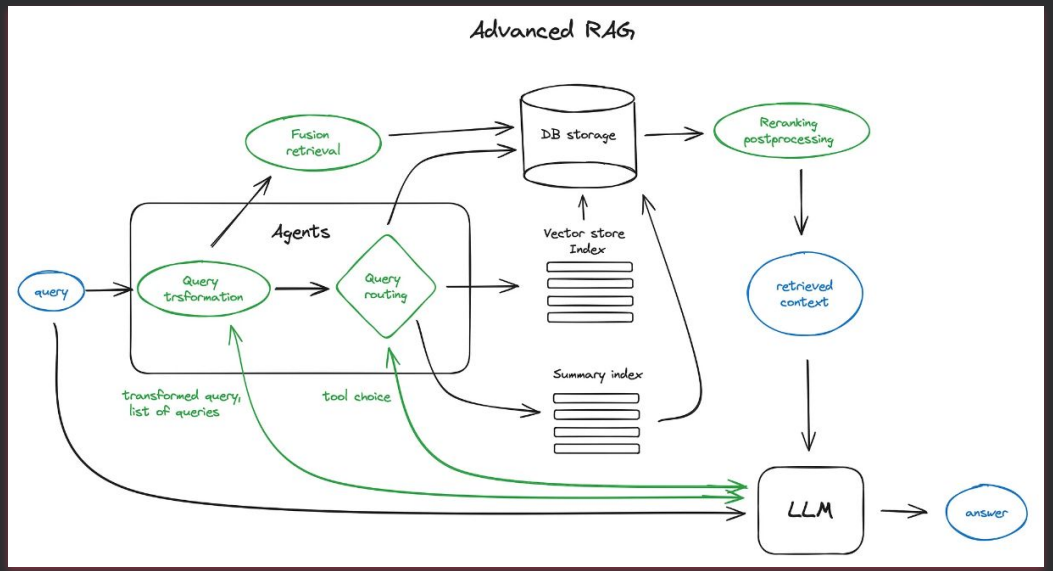




Images source: Datacamp - The Top 5 Vector Databases

Retrieval

The retrieval component is responsible for sourcing relevant information from a database or document collection based on the user's query or input.
This information is attached to prompt and passed to the LLM as an **enriched context.**



Images sources: TowardsAI - Advanced RAG Techniques: an Illustrated Overview

Techniques

PERFORM RETRIEVAL
- Vector Store Index
- Hierarchical Indices (summaries associated)
- Hypothetical Questions and HyDE
- **Context** enrichment
  - Sentence window Retrieval
  - Auto-merging retrieval (aka Parent-Document)
  - Fusion retrieval/hybrid search

POST RETRIEVAL
- Reranking & Filtering
- Query Transformations
  - Step-back prompting
  - Prompt rewriting/Reformulate
- Evaluation

Building context

- Improve Prompting
- Use external resources (Databases & Documents)
- Improve Retrieval
- **Adapt your architecture**

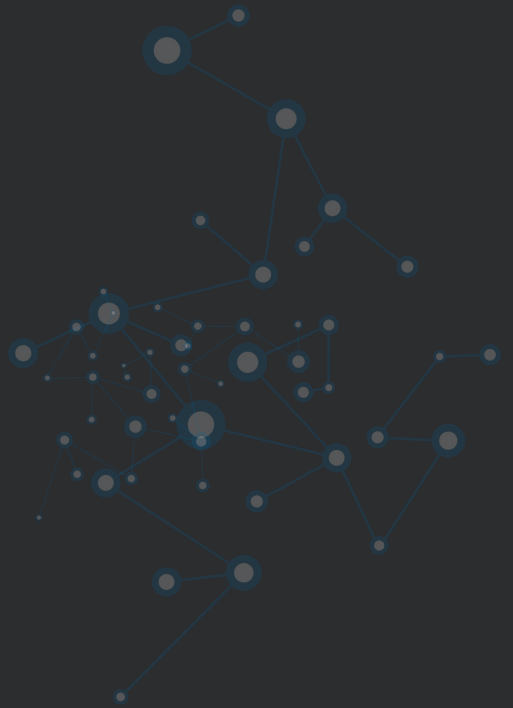
context

Generated with DALL·E 3

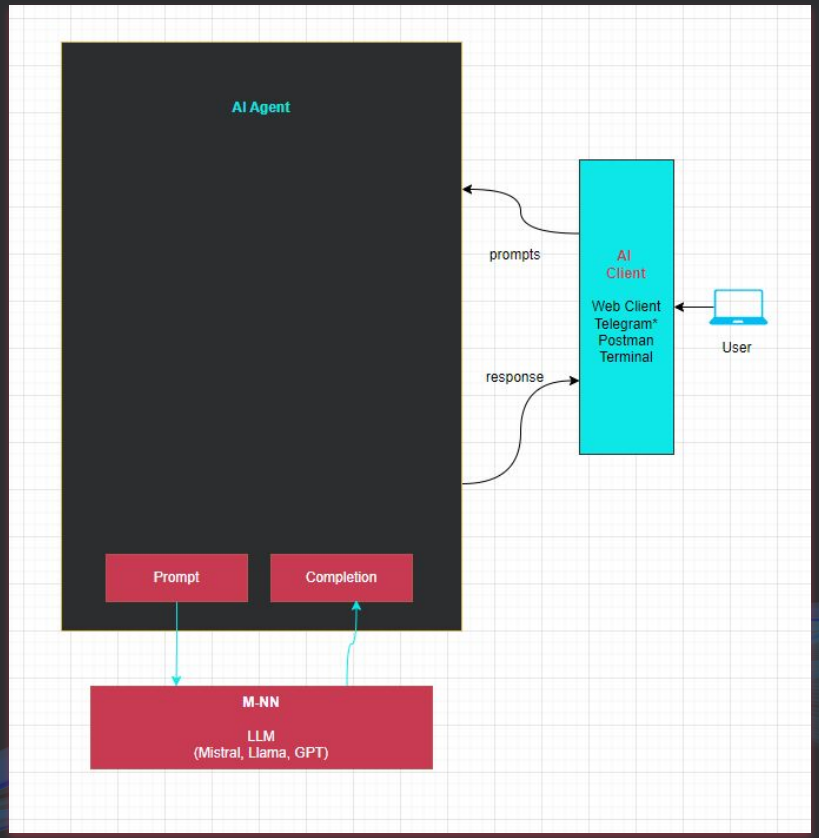
Augmented Talk

oratify.teamcoding.ro/join

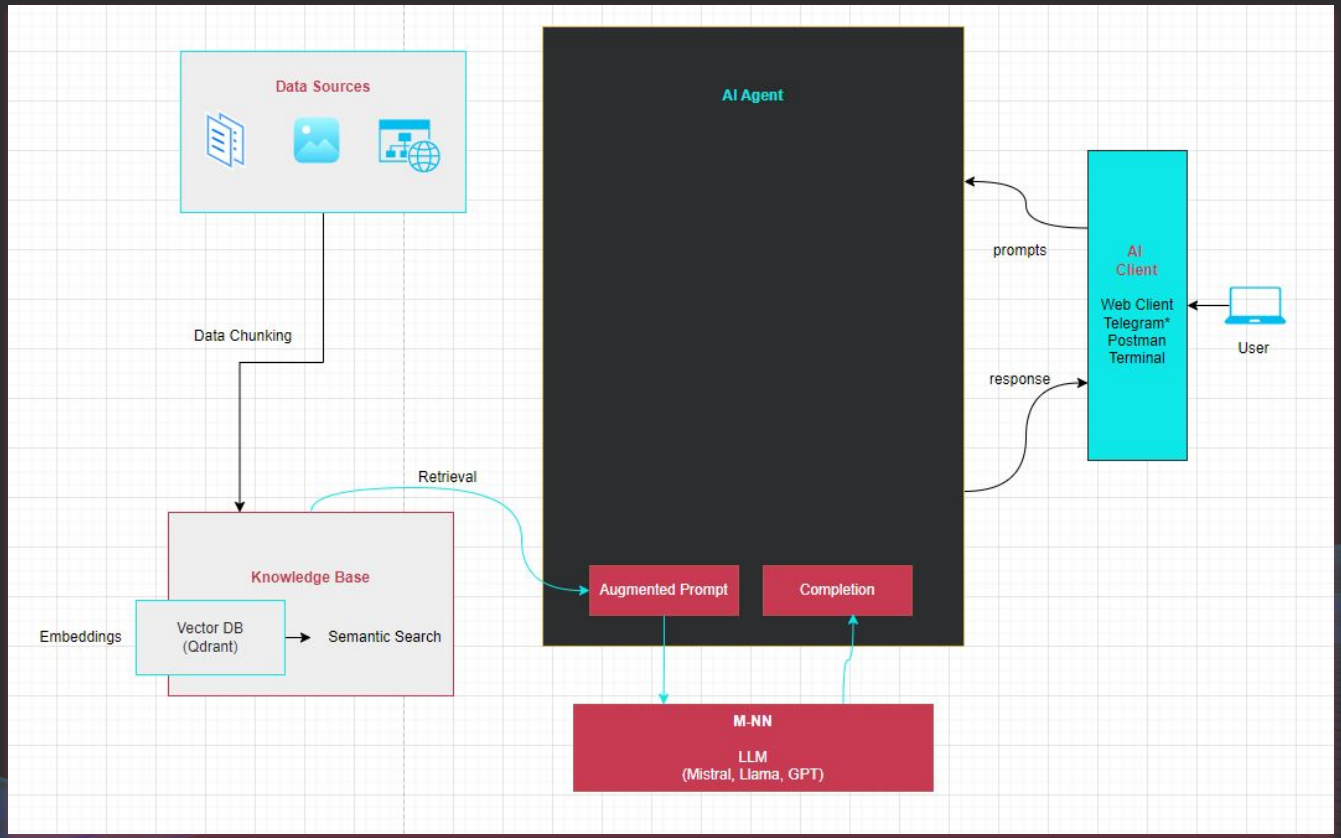
CONTINUAL
BOT



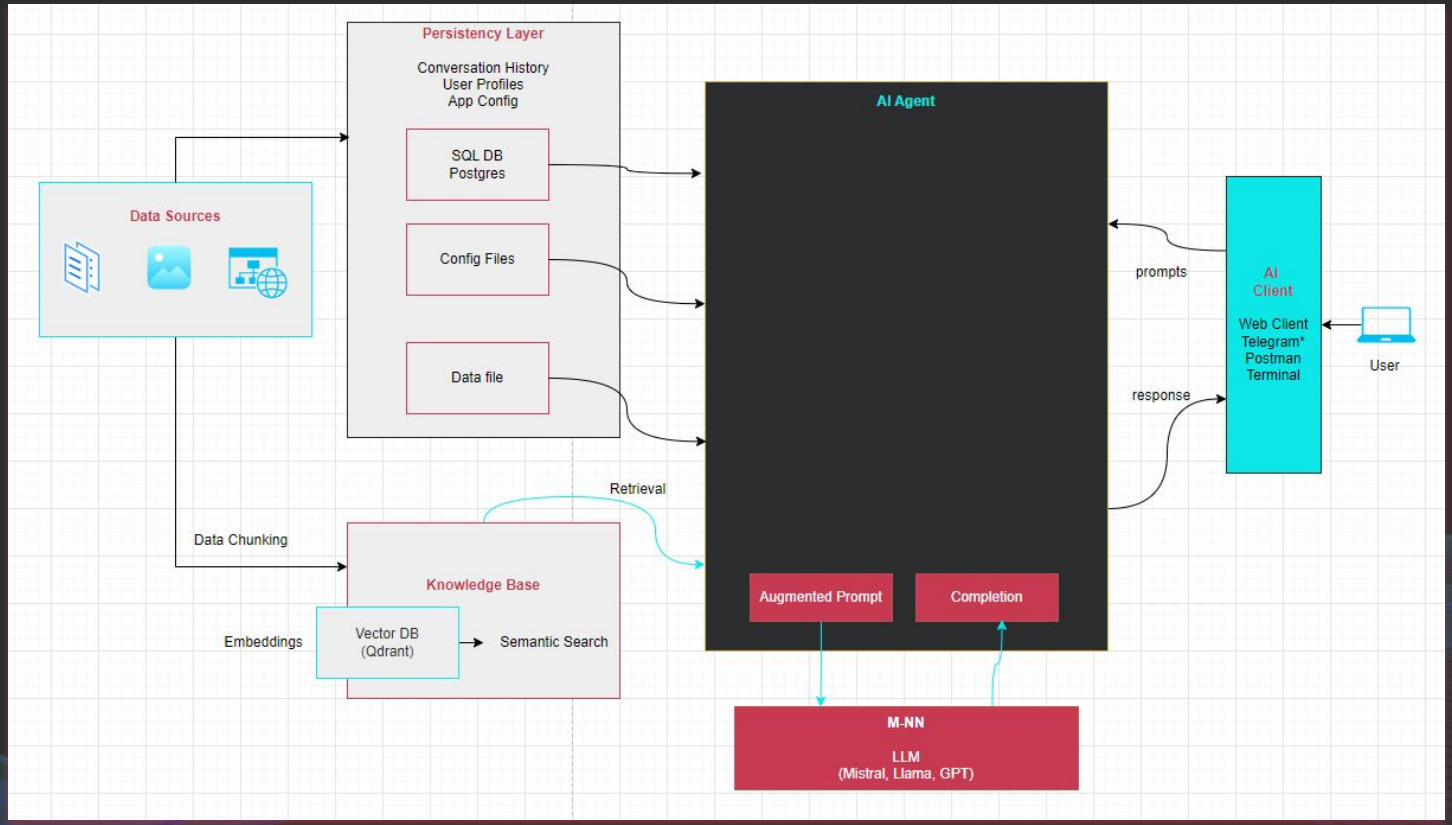
First Steps


AI Agent
prompts
response
AI
Client
Web Client
Telegram*
Postman
Terminal
User
Prompt
Completion
M-NN
LLM
(Mistral, Llama, GPT)

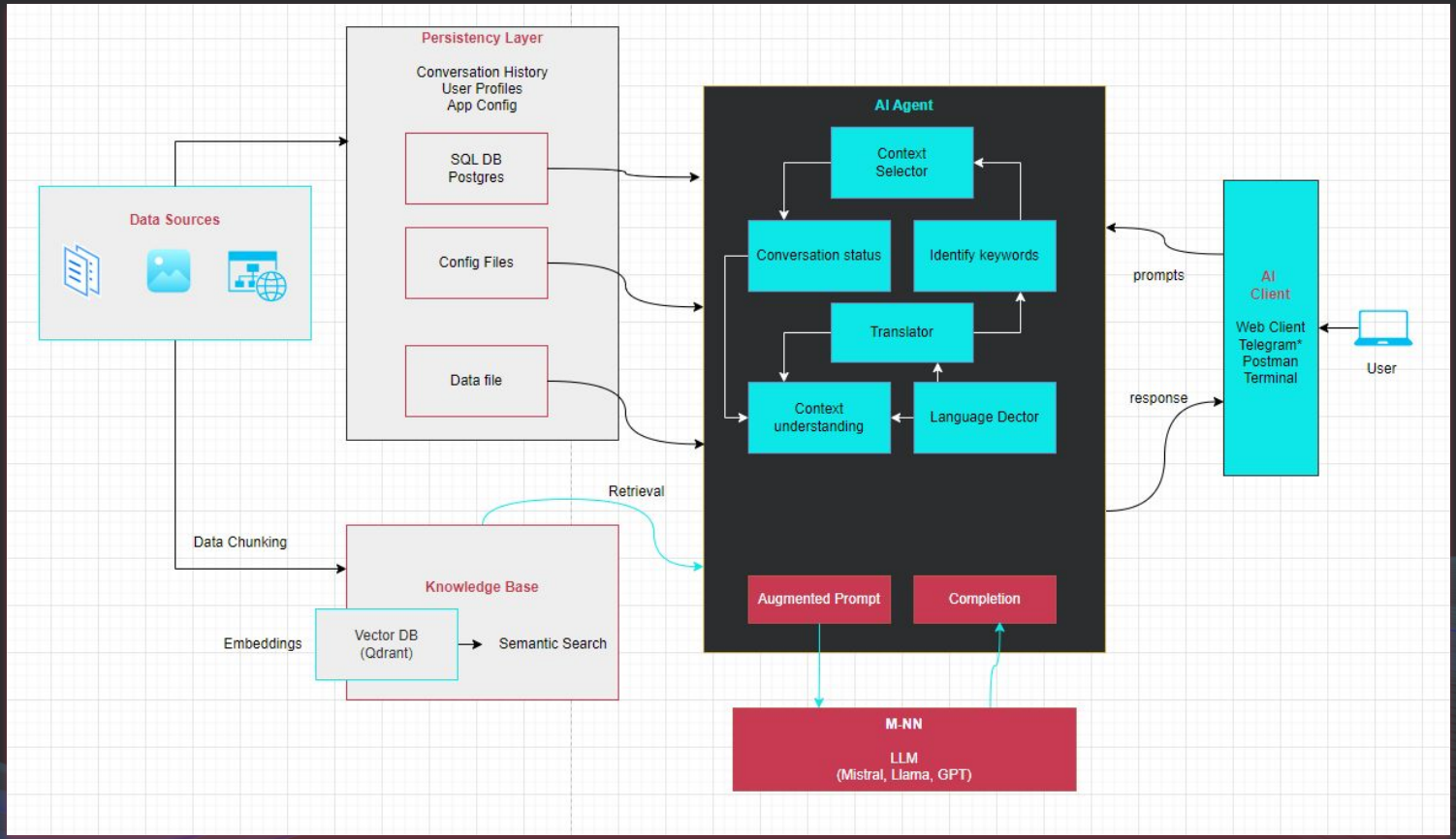
LLM Wrapper


Data Sources
AI Agent
prompts
AI Client
Web Client
Telegram*
Postman
Terminal
User
response
Data Chunking
Retrieval
Knowledge Base
Embeddings
Vector DB (Qdrant)
Semantic Search
Augmented Prompt
Completion
M-NN
LLM
(Mistral, Llama, GPT)

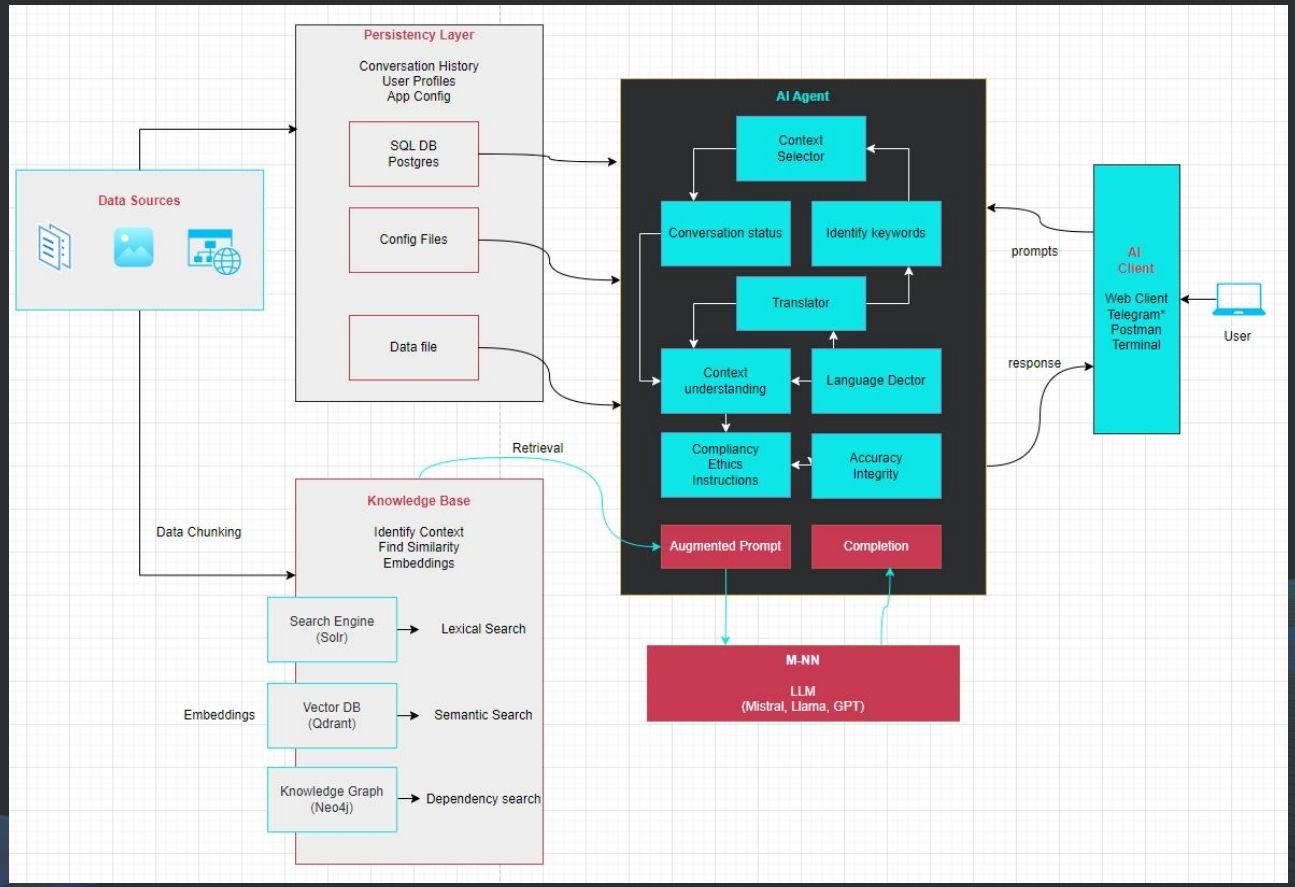
System Handlers



Context Handlers



Go Enterprise

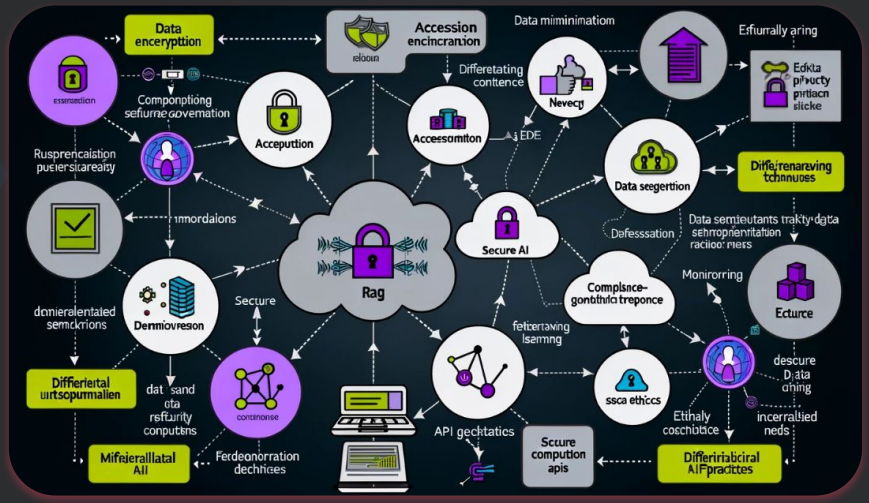


Enterprise Data Protection and Privacy

- **Safeguarding sensitive information** and maintaining customer trust.
- **Regulatory Compliance**: Adherence to laws like GDPR, which mandate data security and privacy.
- **Encryption & Anonymization:** Secure data and protect individual identities.
- **Access Control:** Implementing strict permissions and access policies to sensitive data.
- **Regular Audits:** Monitoring and auditing to identify and rectify any vulnerabilities.
- Importance of using **External Data** & **RAG** models.
- **Augment context** in order to get accurate completions.

Key Factors in Privacy

- Knowledge base separation
- Closed LLM
- Dependencies management
- Leverage LLM in privacy analysis
- Ensure high accuracy on retrieval layer
- Choose the right data chunking strategy



Generated with DALLE-3

RAG Use Cases

- **Similarity Search (Image, Text etc)**
- **Recommendations**
- **Chatbots**
- Matching Engines
- **Anomalies detection**
- HR & Job search
- Advertising (Product match & discover)
- Biometric identification
- Customer support & Sales optimization
- **Fashion Search**
- Fintech
- Law Case search
- **Media & Games**
- **Medical Diagnosis**
- ***Future***

Generated with DALL·E 3

luciangruia.ro

Get in touch

AI MultimediaLab

CIKLUM

CONTINUAL BOT

teamcoding
software based world

luciangruia.ro