Introduction to RAG:

Introduction to RAG

Discover how Retrieval-Augmented Generation (RAG) combines the power of language models with external knowledge to deliver accurate, context-aware answers

Challenges Before RAG

Before RAG, most generative AI systems (like GPT-3 or GPT-4) were "**closed-book models**"—meaning they relied only on what they learned during training.

Here are the key limitations:

1. **Static Knowledge (No Real-Time Updates)**

❑ Once trained, LLMs can't access new or updated information.
❑ Example: GPT-3 (trained till 2021) doesn't know current events or recent research.
❑ Real-time use cases (e.g., customer support, finance, news) became **infeasible**.

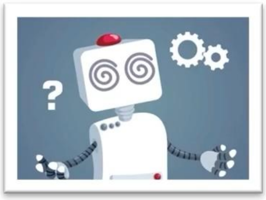
Challenges Before RAG

2. Limited Context Memory

❑ LLMs have a token limit (e.g., 4K, 8K, 32K tokens).
❑ Feeding large documents or full corpora isn't practical or possible.
❑ **Important context often gets left out**, leading to hallucinations or incomplete answers.

3. Hallucinations

❑ LLMs often generate **plausible-sounding but incorrect** information.
❑ Example: Citing fake URLs, inventing authors, or misquoting facts.
❑ There was **no grounding** in external truth or documents.



Challenges Before RAG

4. No Source Attribution

❑ Users could not verify answers.
❑ Businesses in legal, medical, or academic domains **require references** to build trust.

5. Lack of Personalization or Domain Adaptation

❑ Pre-trained LLMs could not understand custom business documents, product manuals, or internal knowledge bases.
❑ Fine-tuning was expensive and time-consuming for every update.

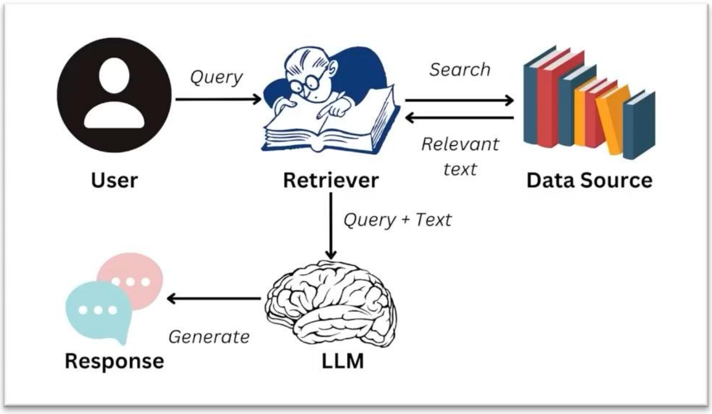
What is Retrieval-Augmented Generation?

Retrieval Augmented Generation (RAG) is a technique that enhances LLMs by integrating them with external data sources. By combining the generative capabilities of models like GPT-4 with precise information retrieval mechanisms, RAG enables AI systems to produce more accurate and contextually relevant responses.

RAG is a hybrid system that combines:

✅ **Retrieval**: Search for relevant information from an external knowledge base
✅ **Generation**: Use a language model to generate a response based on that information

RAG = Retriever + Generator


Query
Search
Relevant text
User
Retriever
Data Source
Query + Text
Generate
Response
LLM