Retrieval Augmented Generation (RAG) and Vector Databases

Understanding RAG, Vector Databases and its components

Contents:

- AL and ML
- Pre-Trained Models
- Large Language Models
- Retrieval Augmented Generation (RAG)
- Vector Databases

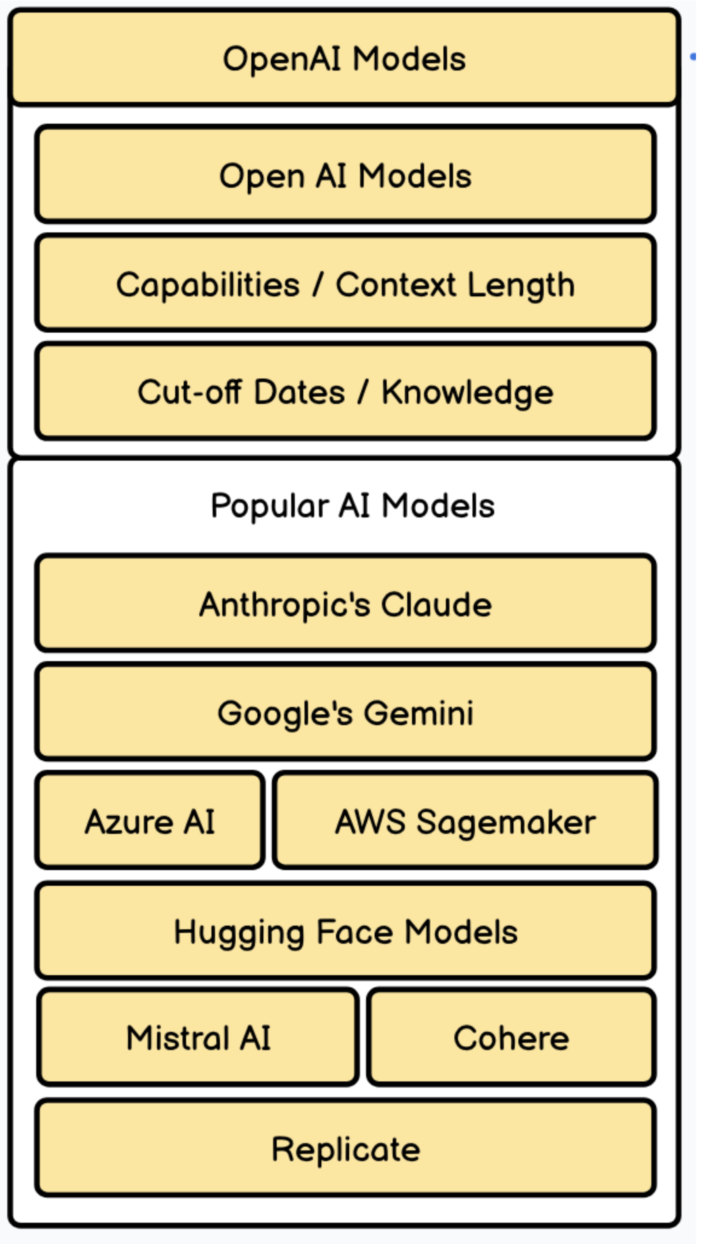
AI Engineer
Vs.
ML Engineer

AI engineers are professionals who specialize in designing, developing, and implementing artificial intelligence (AI) systems. Their work is essential in various industries, as they create applications that enable machines to perform tasks that typically require human intelligence, such as problem-solving, learning, and decision-making.

Machine learning engineers are responsible for building artificial intelligence systems. This fascinating branch of artificial intelligence involves creating models trained on data sets that can predict and adapt to outcomes.

Pre-Trained Models

Pre-trained models are Machine Learning (ML) models that have been previously trained on a large dataset to solve a specific task or set of tasks. These models learn patterns, features, and representations from the training data, which can then be fine-tuned or adapted for other related tasks. Pre-training provides a good starting point, reducing the amount of data and computation required to train a new model from scratch.



OpenAI Models

Open AI Models

Capabilities / Context Length

Cut-off Dates / Knowledge

Popular AI Models

Anthropic's Claude

Google's Gemini

Azure AI

AWS Sagemaker

Hugging Face Models

Mistral AI

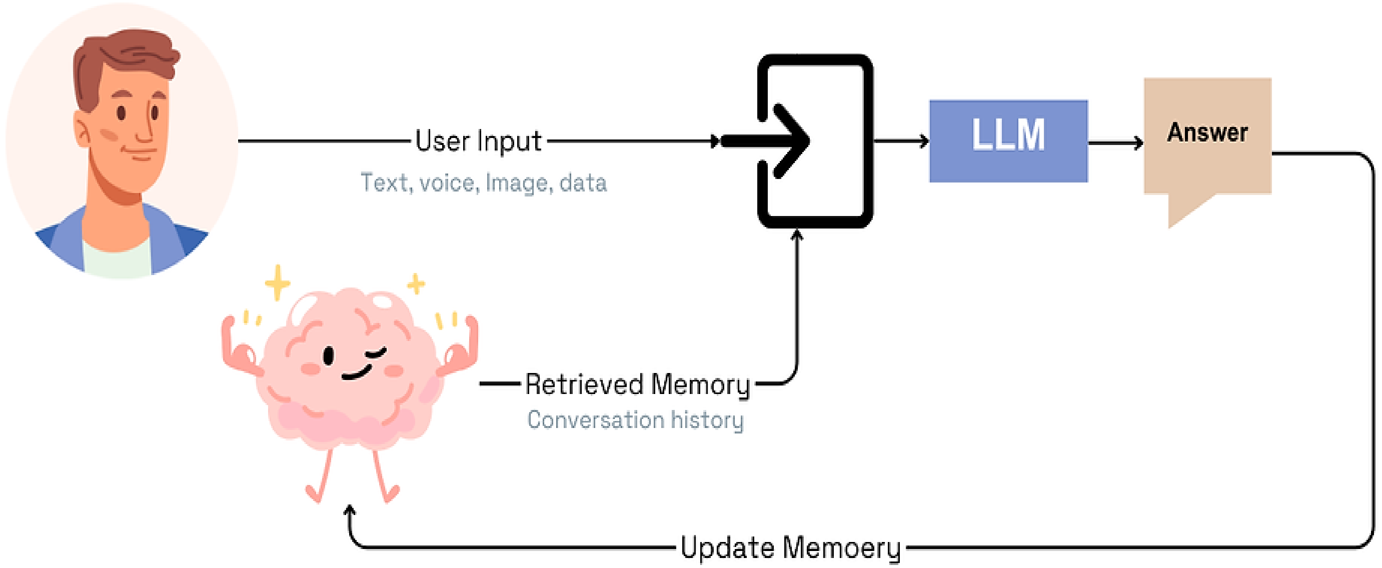
Cohere

Replicate

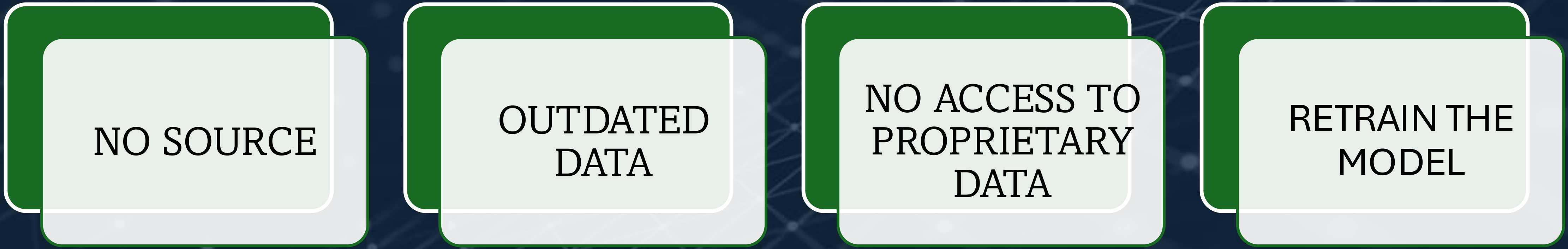
Large Language Models

In the world of artificial intelligence, LLMs are a specially designed subset of machine learning known as deep learning, which uses algorithms trained on large data sets to recognize complex patterns. LLMs learn by being trained on massive amounts of text.

Architectures of LLM Applications



LLM Challenges



- NO SOURCE
- OUTDATED DATA
- NO ACCESS TO PROPRIETARY DATA
- RETRAIN THE MODEL

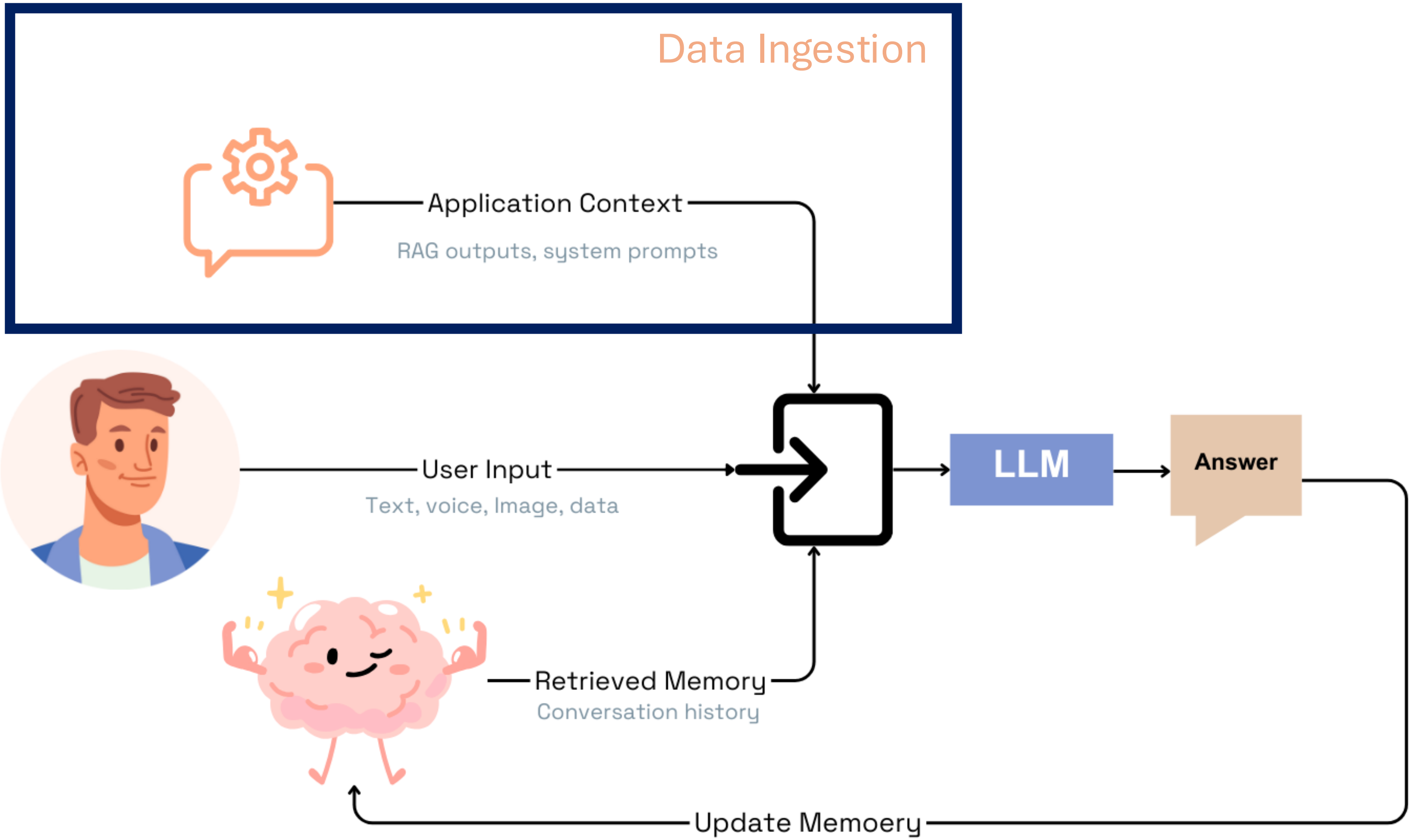
Retrieval Augmented Generation (RAG)

Bridging the gap between Large Language Models (LLMs) and Enterprises

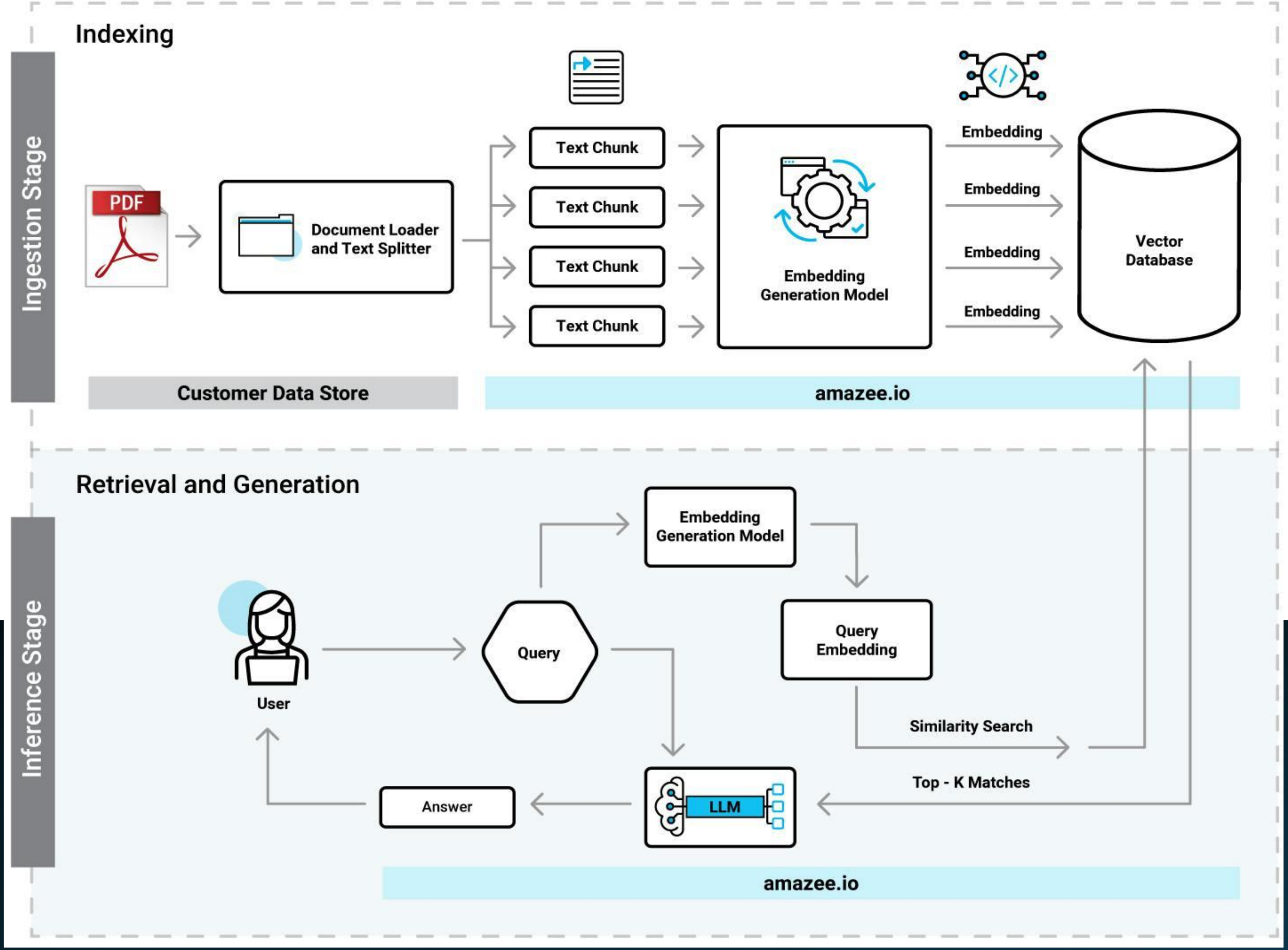
What is RAG (Retrieval Augmented Generation)?

Retrieval augmented generation (RAG) is an architecture for optimizing the performance of an artificial intelligence (AI) model by connecting it with external knowledge bases. RAG helps large language models (LLMs) deliver more relevant responses at a higher quality.

Emerging Architectures of LLM Applications

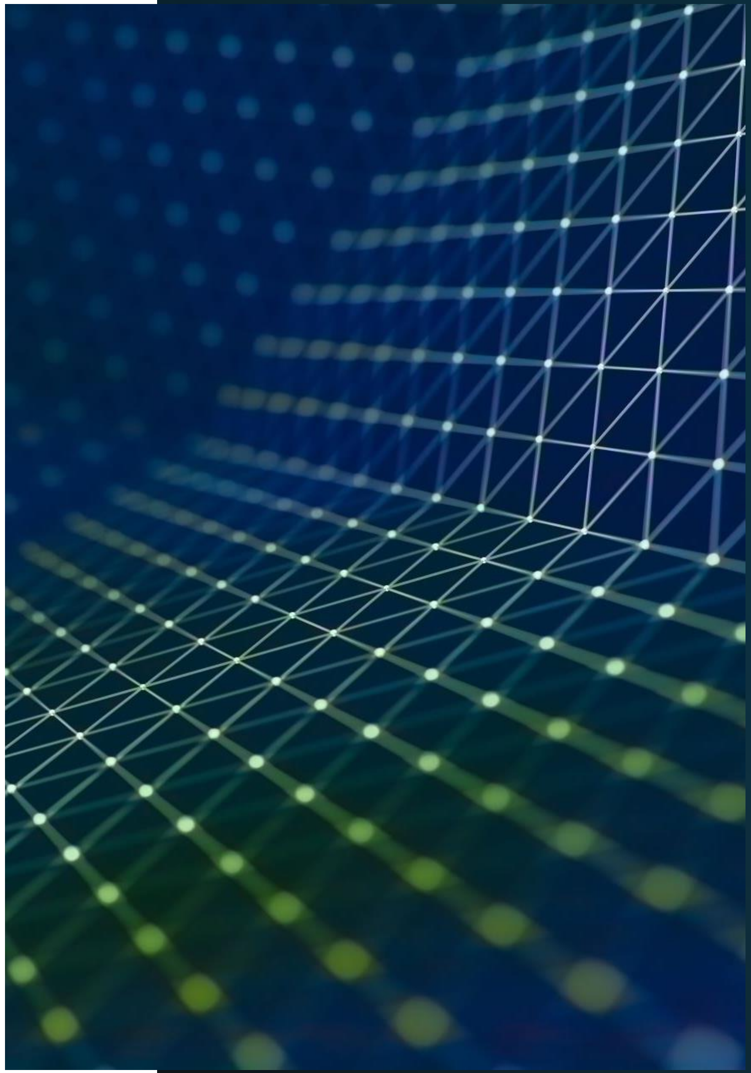


RAG Architecture



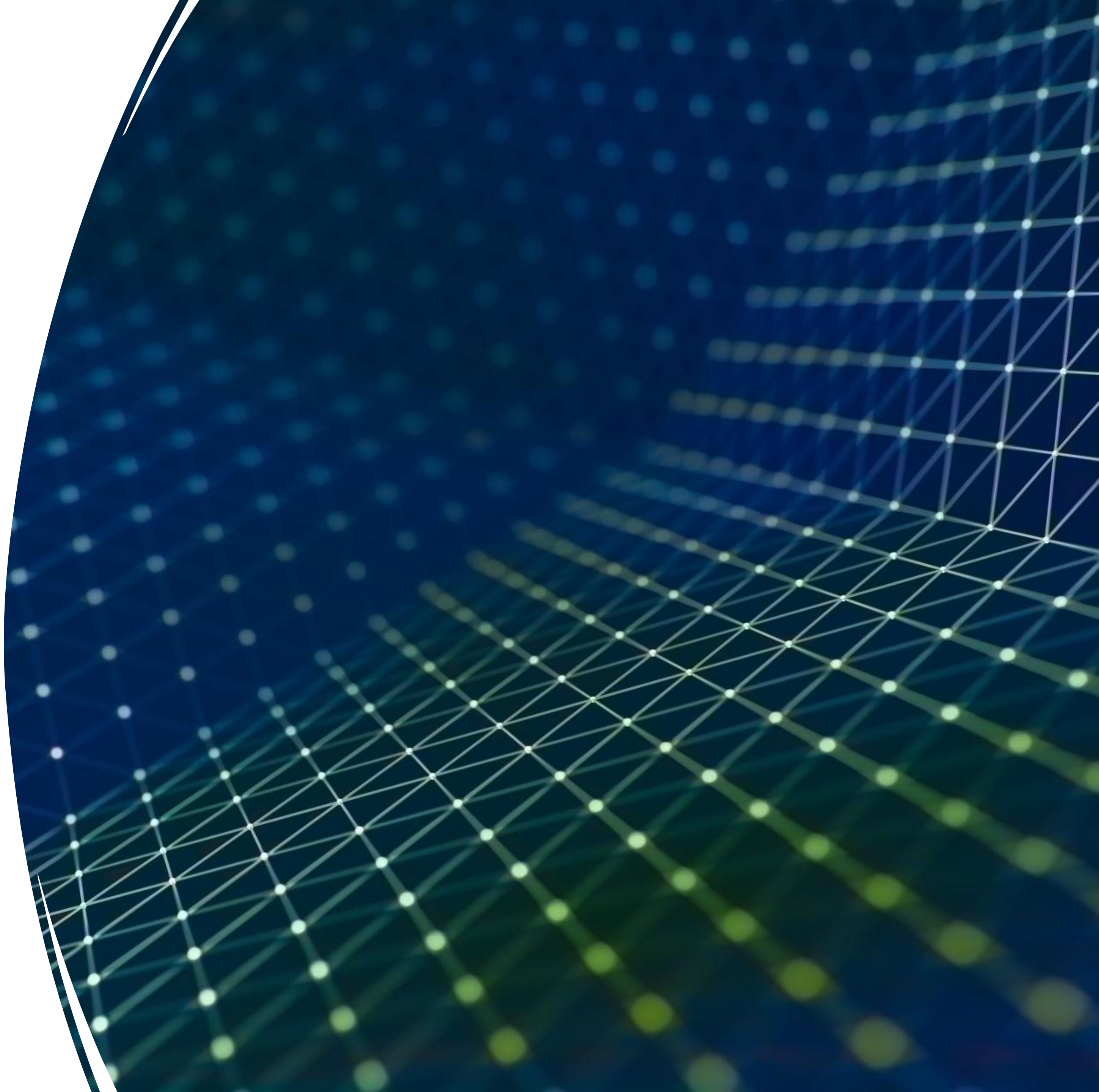
RAG Components

- CHUNKING
- EMBEDDINGS
- VECTOR EMBEDDING
- VECTOR DATABASE
- VECTOR INDEXING



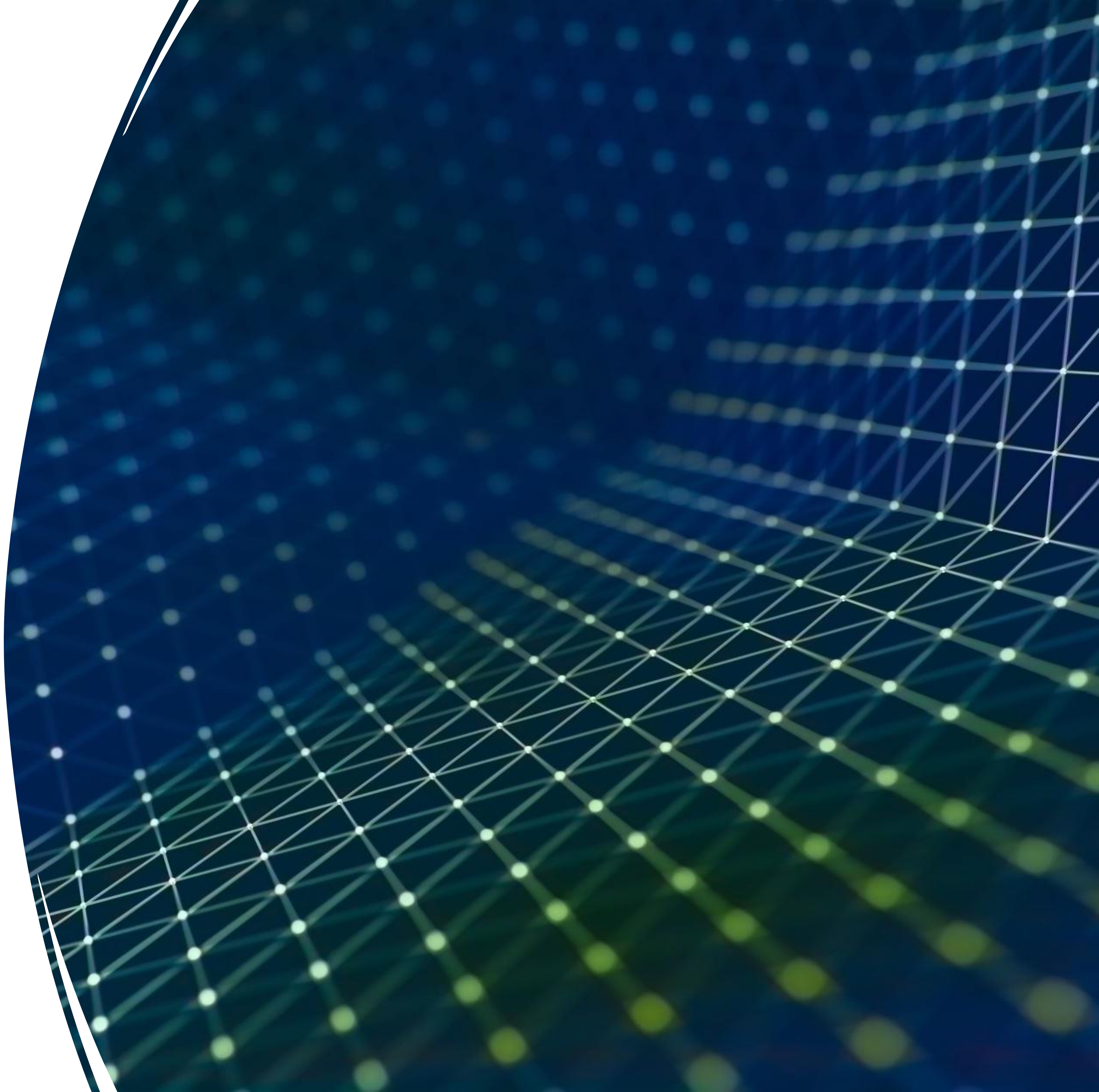
Chunking

- The chunking step in Retrieval-Augmented Generation (RAG) involves breaking down large documents or data sources into smaller, manageable chunks. This is done to ensure that the retriever can efficiently search through large volumes of data while staying within the token or input limits of the model.



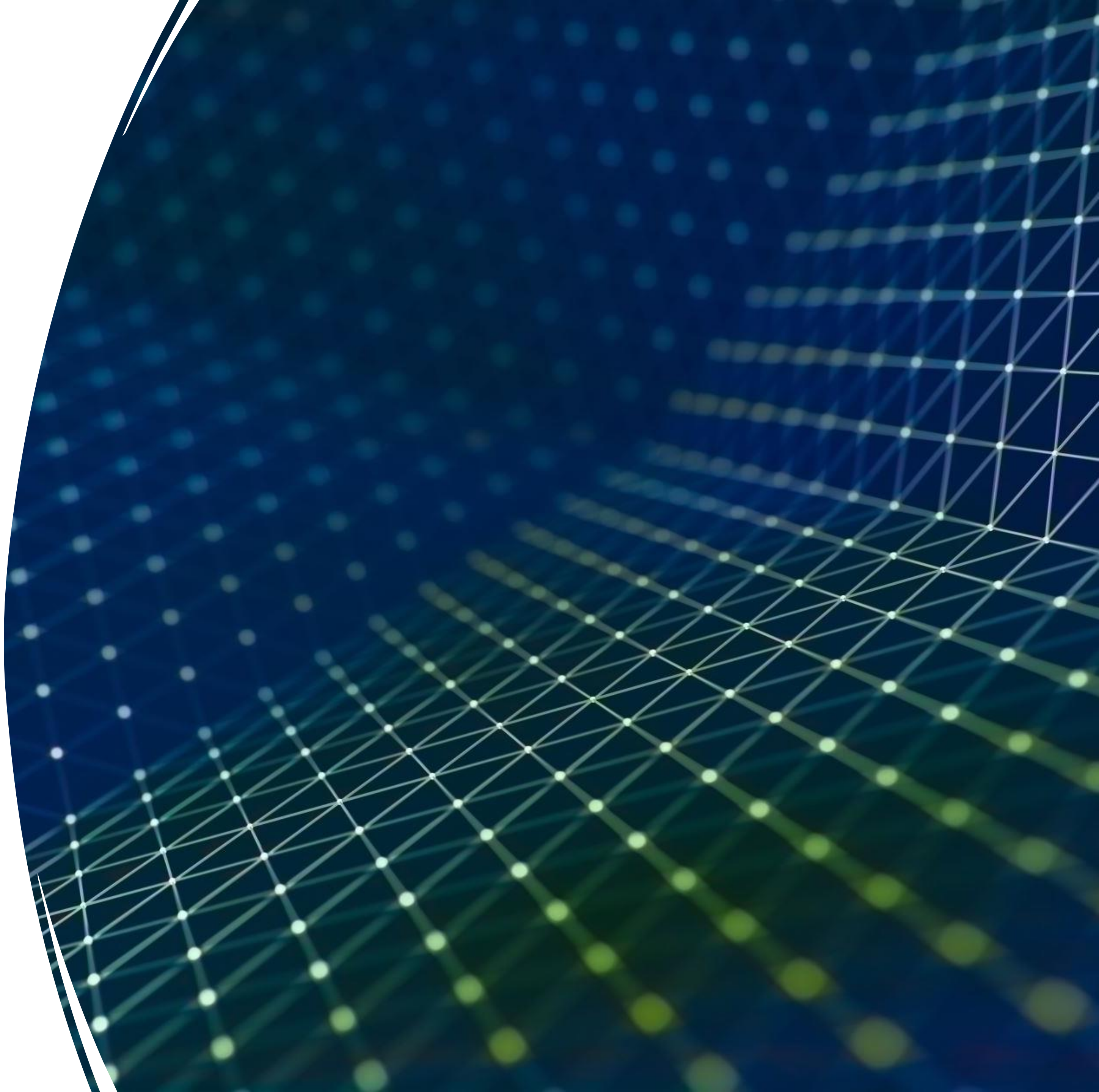
Embeddings

- A numerical representation of data that captures its relevant qualities. Embeddings are designed to capture the underlying structure or properties of the data



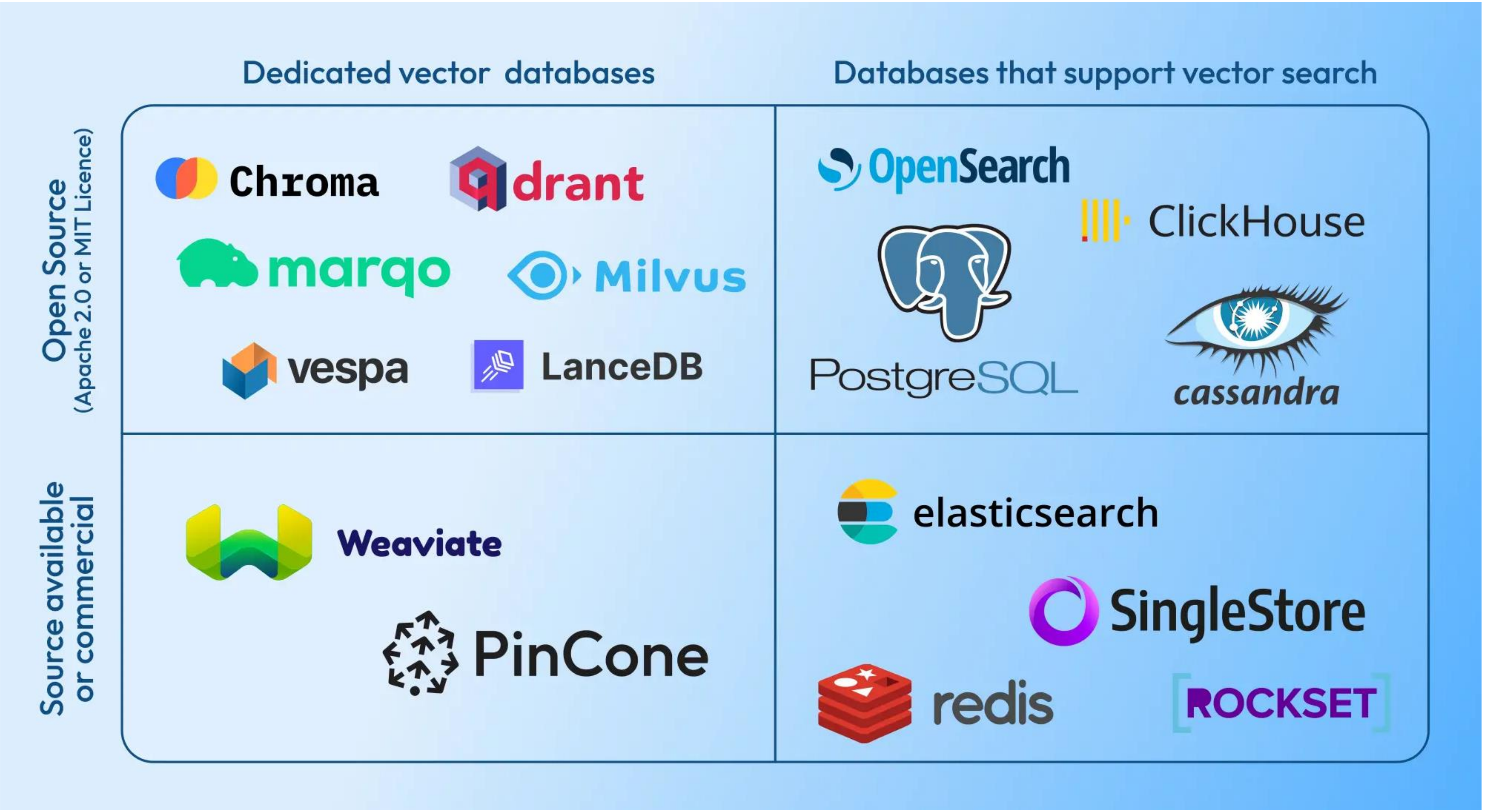
Vector Embeddings

- Vector embeddings are numerical representations of data points that express different types of data, including nonmathematical data such as words or images.



Vector Databases

When implementing Retrieval-Augmented Generation (RAG), a vector database is used to store and efficiently retrieve embeddings, which are vector representations of data like documents, images, or other knowledge sources. During the RAG process, when a query is made, the system converts it into an embedding and searches the vector database for the most relevant, similar embeddings

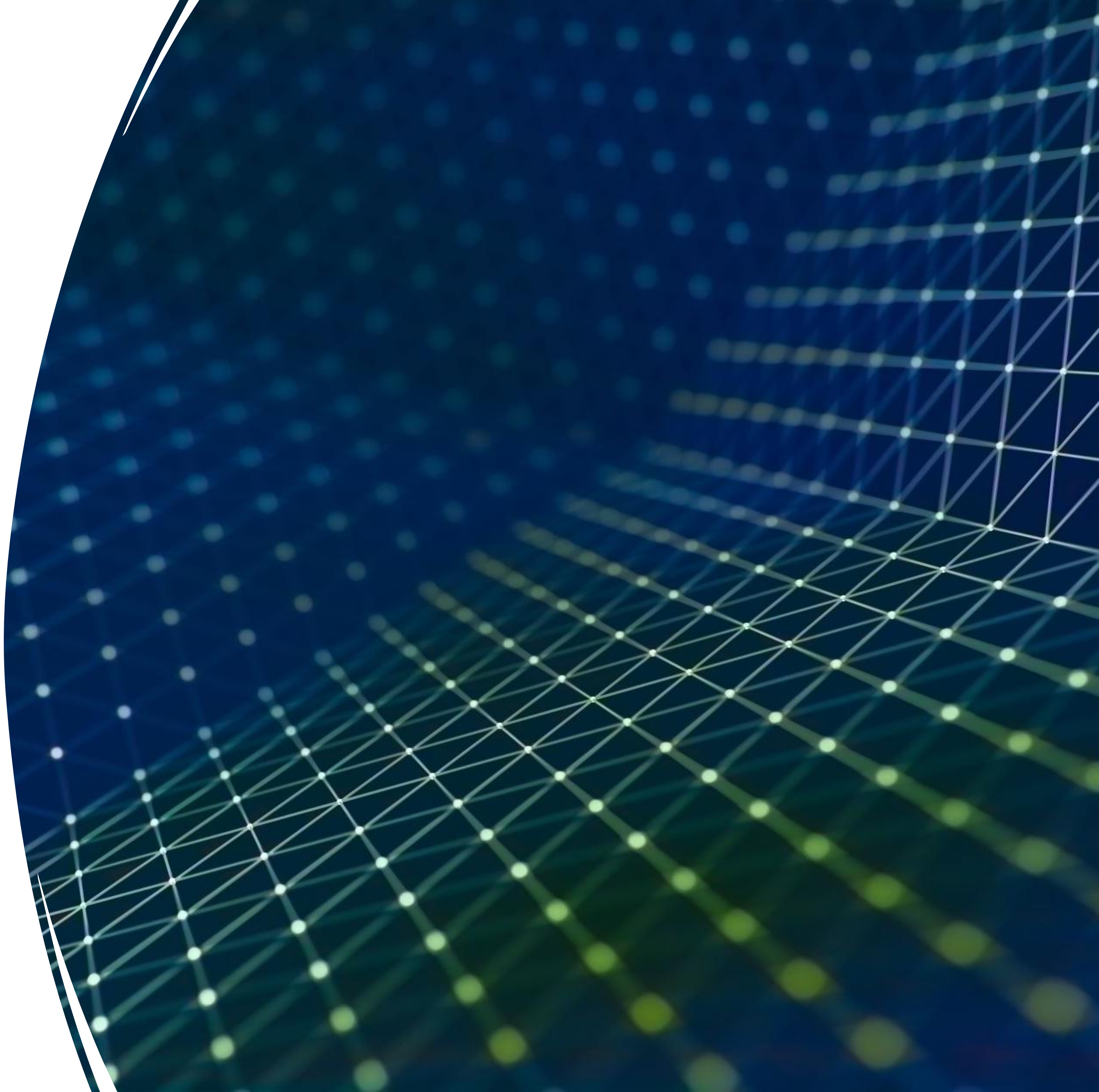


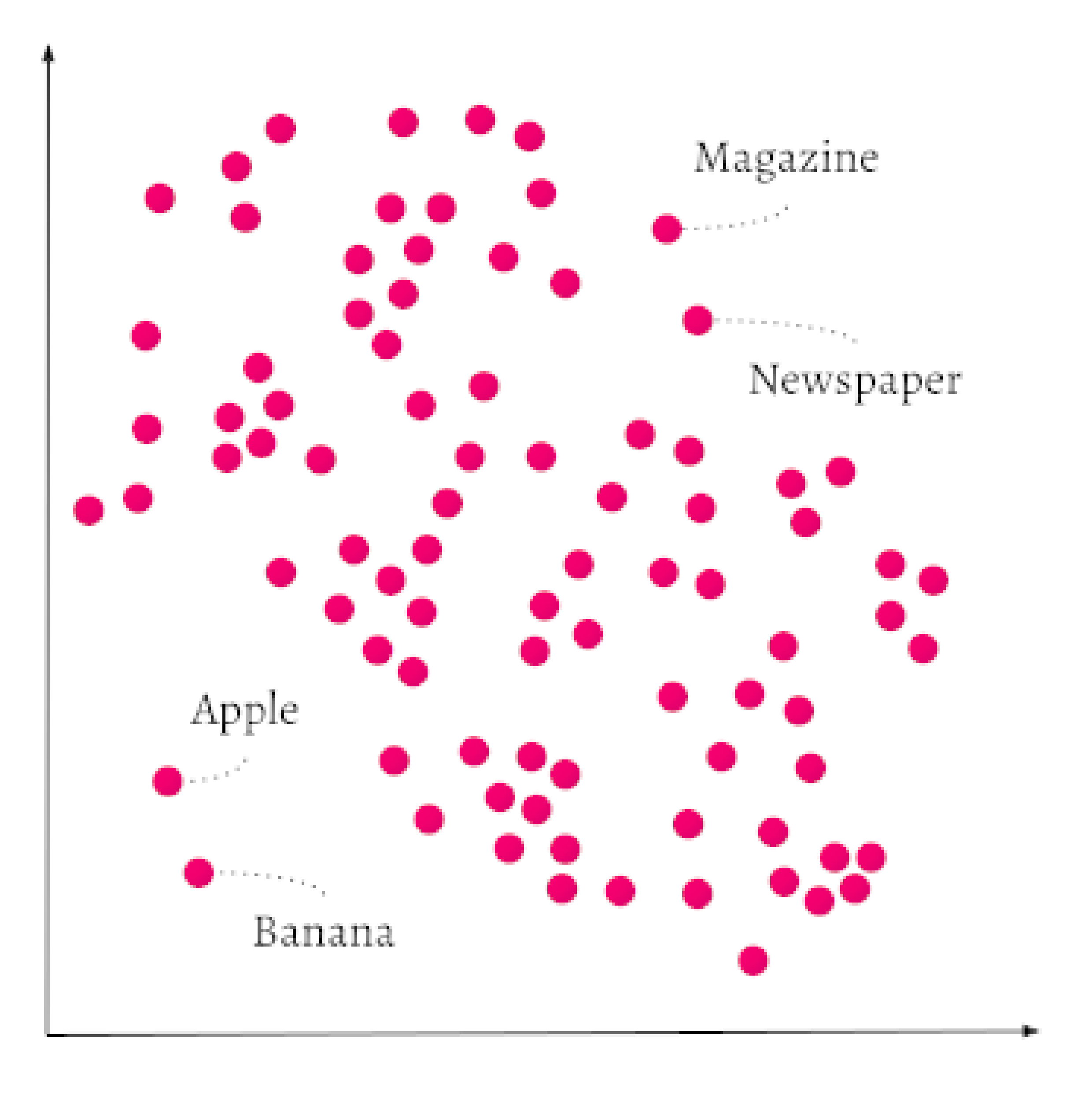
	Dedicated vector databases	Databases that support vector search
Open Source (Apache 2.0 or MIT Licence)	Chroma qdrant marqo Milvus vespa LanceDB	OpenSearch ClickHouse PostgreSQL cassandra
Source available or commercial	Weaviate PinCone	elasticsearch SingleStore redis ROCKSET

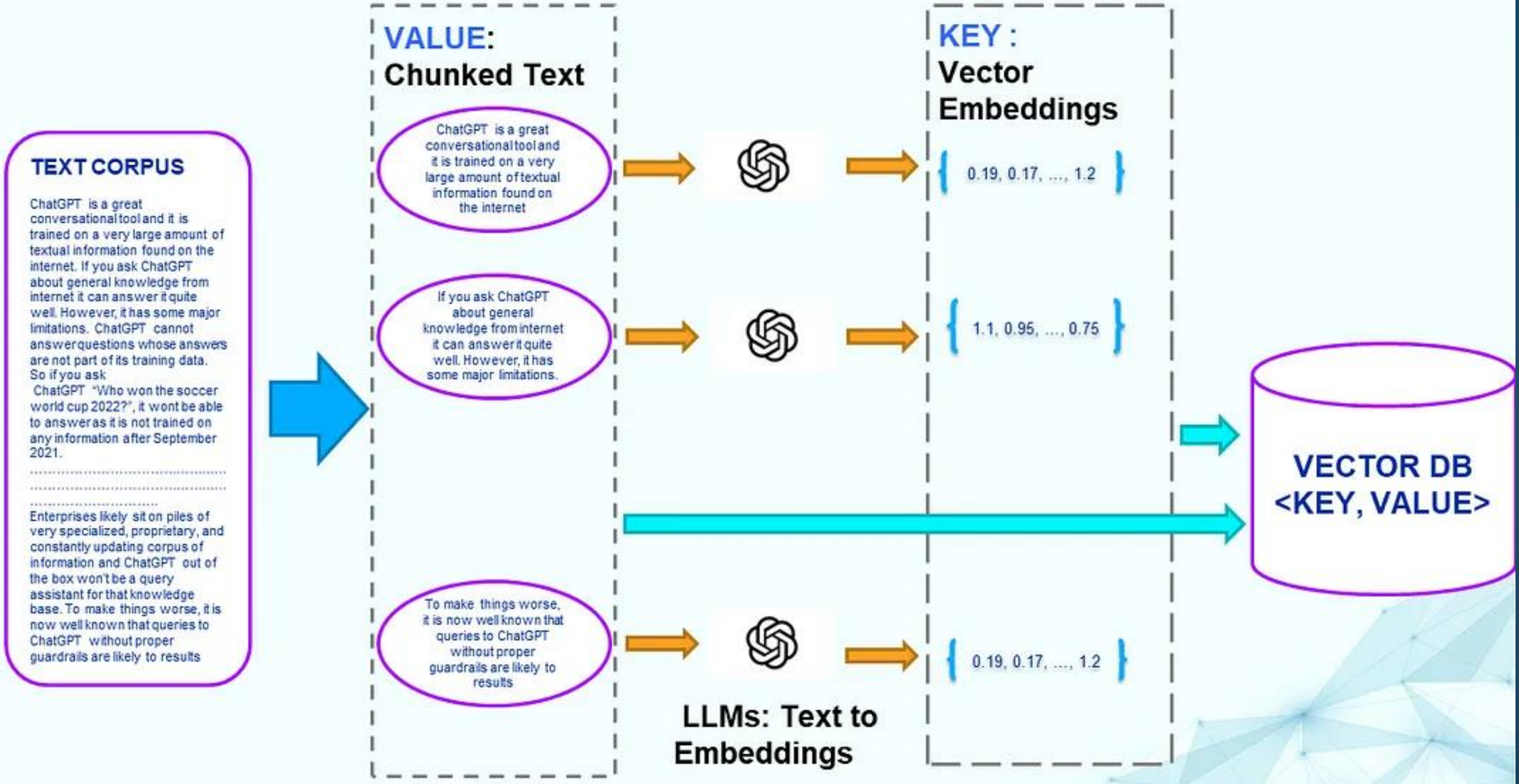
Indexing in Vector Databases

Once our data is converted into vector embeddings, the data is organized in a multi-dimensional vector space for efficient similarity search.

Vector indexing is not just about storing data, it's about intelligently organizing the vector embeddings to optimize the retrieval process. This technique involves advanced algorithms to neatly arrange the high-dimensional vectors in a searchable and efficient manner. This arrangement is not random; it's done in a way that similar vectors are grouped together, by which vector indexing allows quick and accurate similarity searches and pattern identification, especially for searching large and complex datasets.




Magazine
Newspaper
Apple
Banana



TEXT CORPUS

ChatGPT is a great conversational tool and it is trained on a very large amount of textual information found on the internet. If you ask ChatGPT about general knowledge from internet it can answer it quite well. However, it has some major limitations. ChatGPT cannot answerquestions whose answers are not part of its training data. So if you ask ChatGPT "Who won the soccer world cup 2022?", it wont be able to answeras it is not trained on any information after September 2021.

....

....

....

Enterprises likely sit on piles of very specialized, proprietary, and constantly updating corpus of information and ChatGPT out of the box won't be a query assistant for that knowledge base. To make things worse, it is now well known that queries to ChatGPT without proper guardrails are likely to results

VALUE: Chunked Text

ChatGPT is a great conversational tool and it is trained on a very large amount of textual information found on the internet

If you ask ChatGPT about general knowledge from internet it can answer it quite well. However, it has some major limitations.

To make things worse, it is now well known that queries to ChatGPT without proper guardrails are likely to results

LLMs: Text to Embeddings

KEY : Vector Embeddings

{ 0.19, 0.17, ..., 1.2 }

{ 1.1, 0.95, ..., 0.75 }

{ 0.19, 0.17, ..., 1.2 }

VECTOR DB <KEY, VALUE>

ThirdAI

Thank You!