DEVOXX FRANCE

Using Generative AI with Java.

/ Go Beyond “simple” chatbots

CEDRICK LUNVEN
DATASTAX

ABDEL SGHIOUAR
GOOGLE

DEEP DIVE
3H

Cédrick Lunven

Developer Advocate @DataStax

GOOD BARDS

- Stuff I do
  - Creator of ff4j (ff4j.org)
  - Distributed Systems &
  - C*
  - Tools (clients, sdk, cli)
- AI
  - DataStax Products
  - Dev Ecosystem
  - Langchain4j
  - Spring AI

Abdel Sghiouar

Developer Advocate @Google

Google Cloud

- Stuff I do
  - Kubernetes & Istio
  - Kubernetes Podcast
  - Networking & Security
- AI
  - AI On Kubernetes
  - KubeRay
  - Langchain4j

1. Introduction (20 min) - Abdel
   - Machine learning and Transformers
   - Generative AI
   - Large Language Models

2. Google AI Ecosystem (20 min) - Abdel
   - Google AI Portfolio Gemini
   - Vertex AI

3. Deploying LLM locally (20 min) - Abdel
   - Gemma and OSS Google models
   - GKE Deployments

4. Prompt Engineering (20 min) - Cedrick
   - Building effective prompts
   - Techniques and Demos
   - Best Practices

5. Vector Search
   - Definitions and Algorithms
   - Vector Databases
   - JVector & Apache Cassandra™
   - Vector Stores

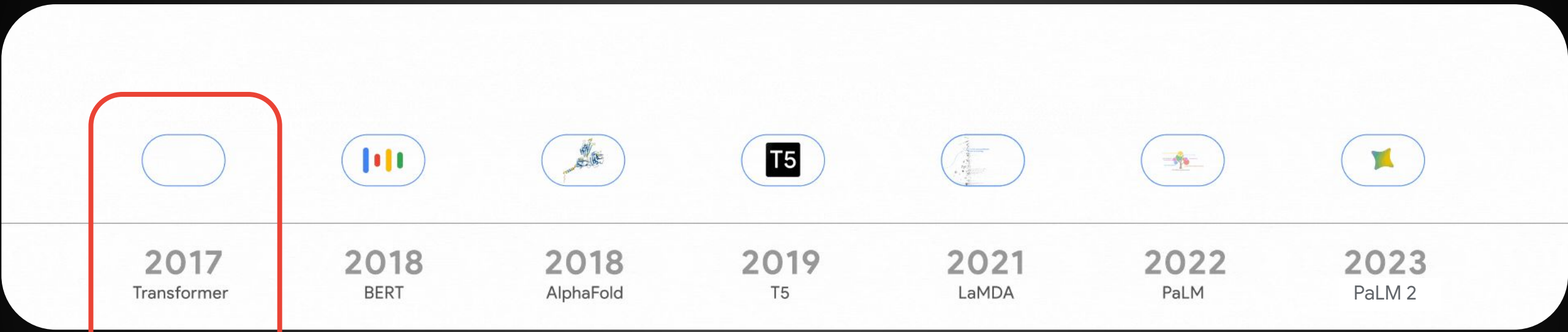
6. Retrieval Augmented Generation
   - Ingestion or Vectorization
   - Vector Searches & metadata filtering
   - Advanced RAG

7. Agents and other use cases
   - Function Calling
   - Vertex AI extension

Q & A

Introduction
Generative AI and LLM

Google invented the Transformer architecture



Google invents Transformer kickstarting LLM revolution

Google's groundbreaking large language model, BERT

AlphaFold predicts 3D models of protein structures

Text-to-Text Transfer Transformer LLM 10B P model open sourced

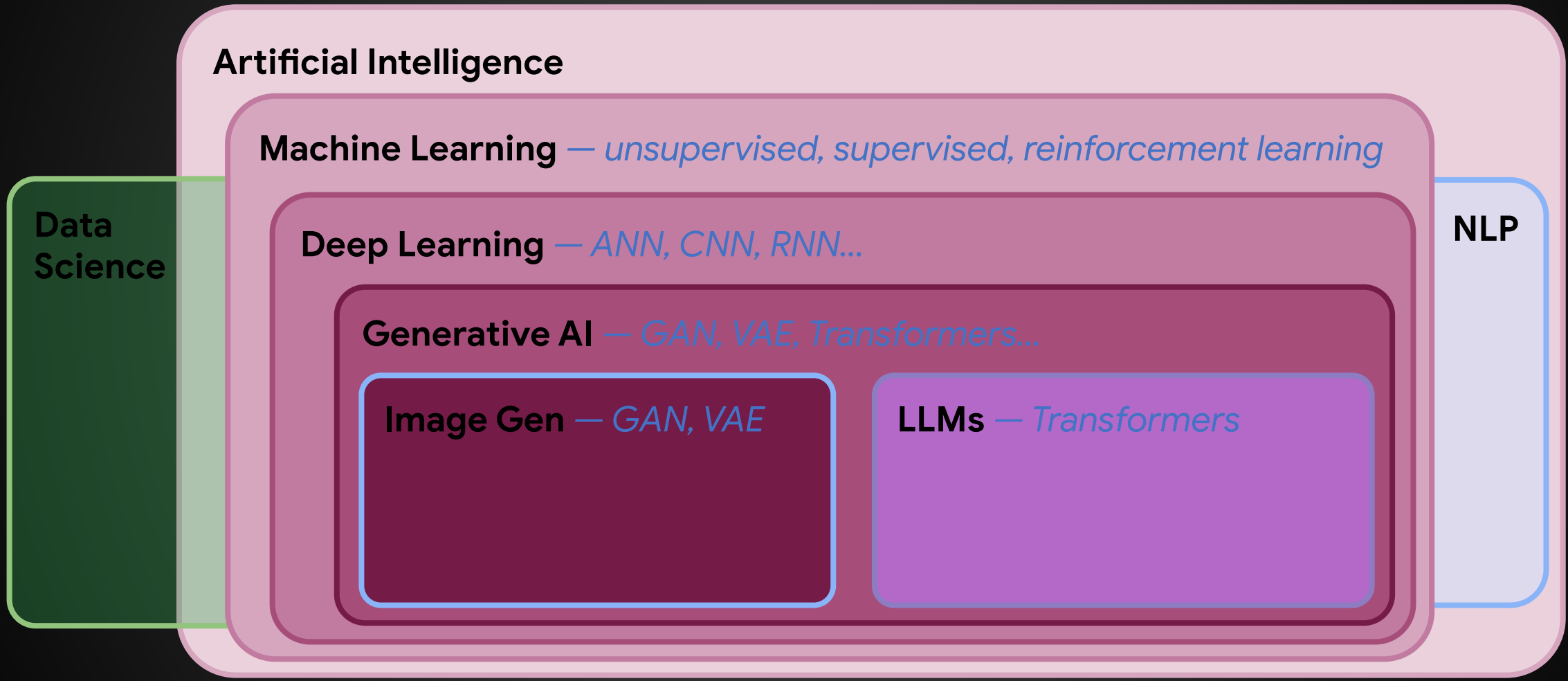
Google LaMDA model trained to converse

Google PaLM single model to generalize across domains

Google PaLM 2 model is the SOTA LLM

Responsible AI at the foundation

Generative AI



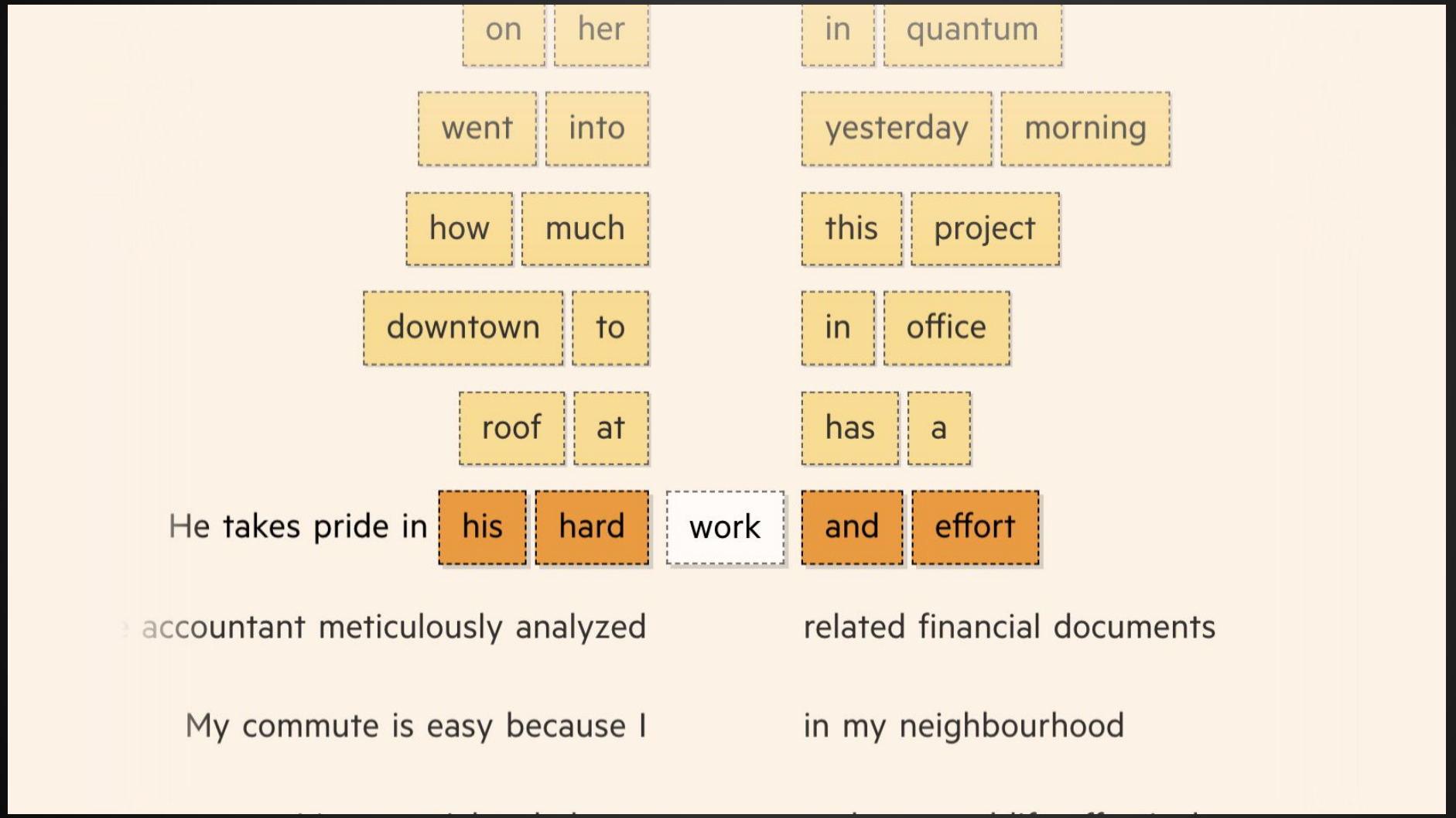
So what are Large Language Models?

- Transformer-based neural network architecture that can **recognize**, **predict**, and **generate** human language
- Trained on huge corpuses of text, in various languages and domains
  - *Ex: PaLM 2 learned 340 billion **parameters**, and trained over 3.6 trillions of **tokens***
- Learn the **statistical relationships between words and phrases**, as well as the patterns of human language
- Can be **fine-tuned** for specific tasks or domain knowledge

Transformers & Tokens

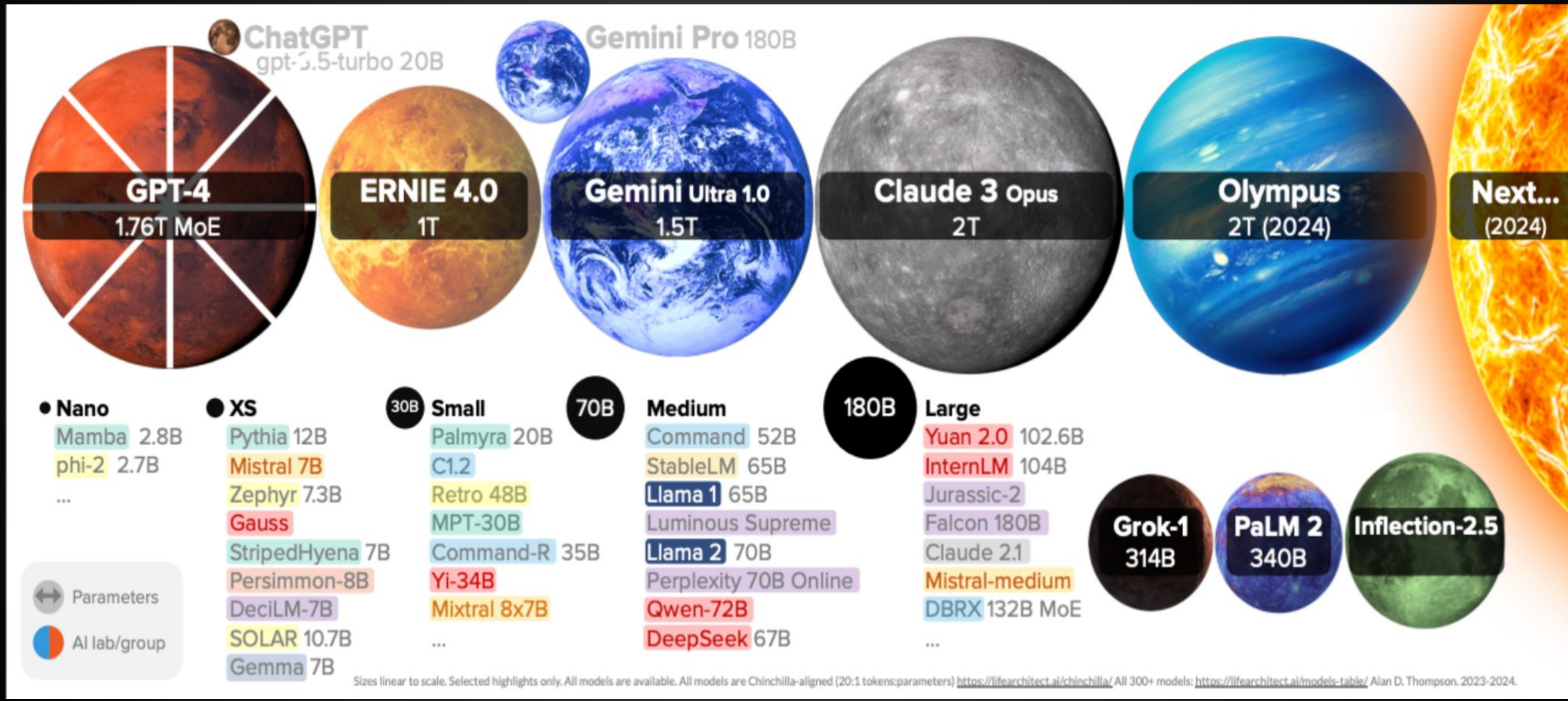
https://ig.ft.com/generative-ai/

demo

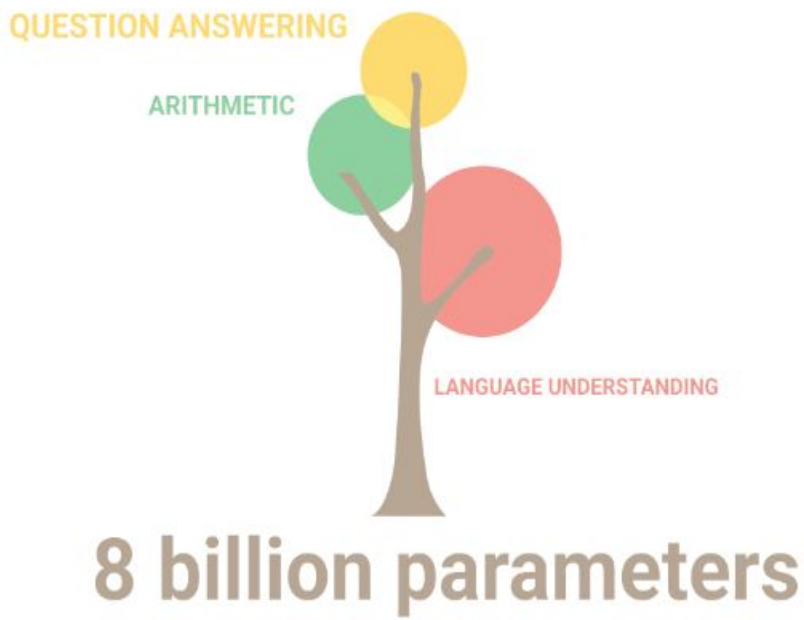


Large Language models - March 2024

https://lifearchitect.ai/models/



With larger models emerge new capabilities



https://blog.research.google/2022/04/pathways-language-model-palm-scaling-to.html

Generative AI use cases 2024

Language

- Writing
- Summarization
- Ideation
- Classification
- Sentiment analysis
- Extraction
- Chat
- Search

Code

- Code generation
- Code completion
- Code chat
- Code conversion

Speech

- Speech to text
- Text to speech

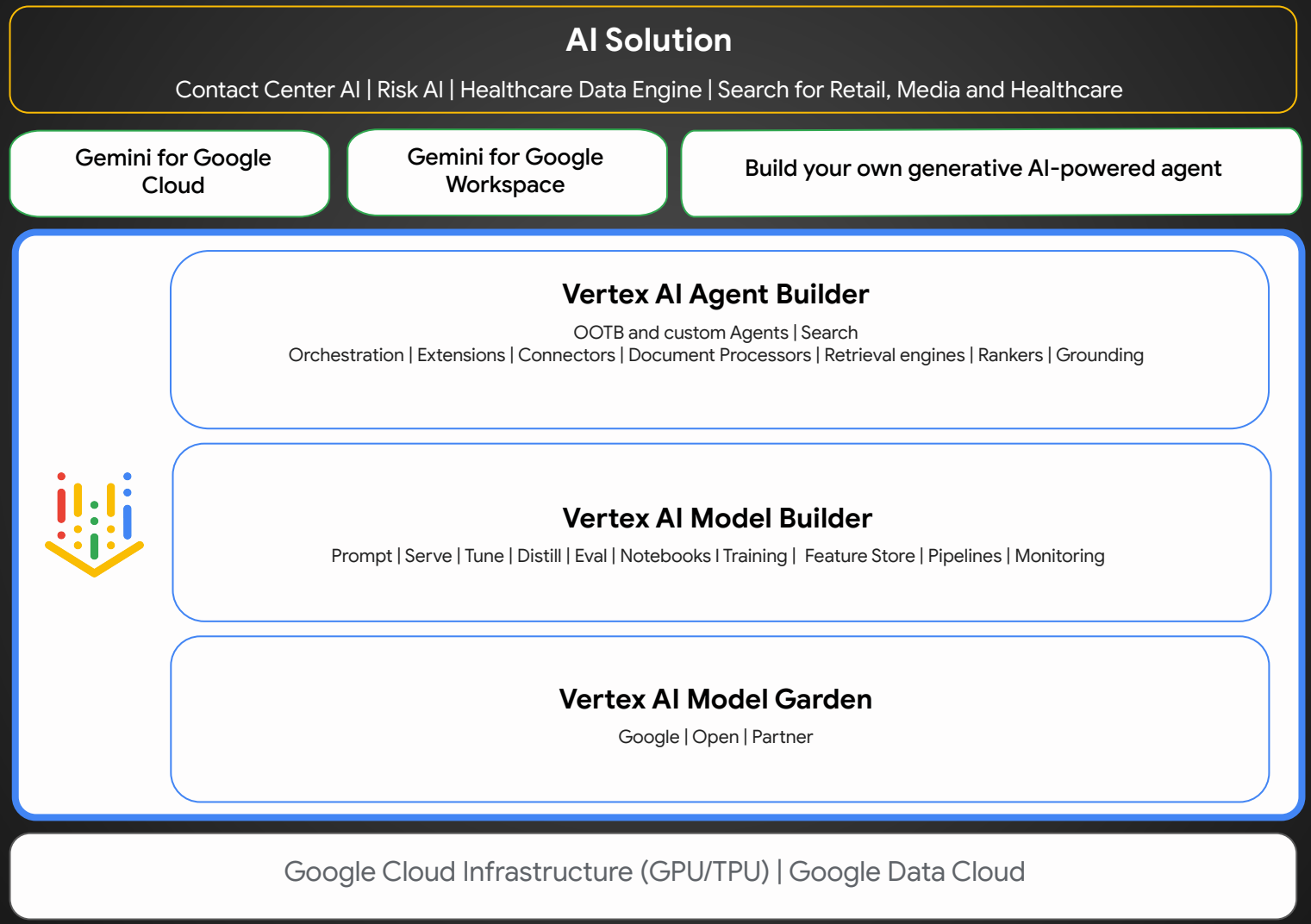
Vision

- Image generation
- Image editing
- Captioning
- Image Q&A
- Image search
- Video descriptions

2

Google AI Ecosystem
Gemini and Vertex AI

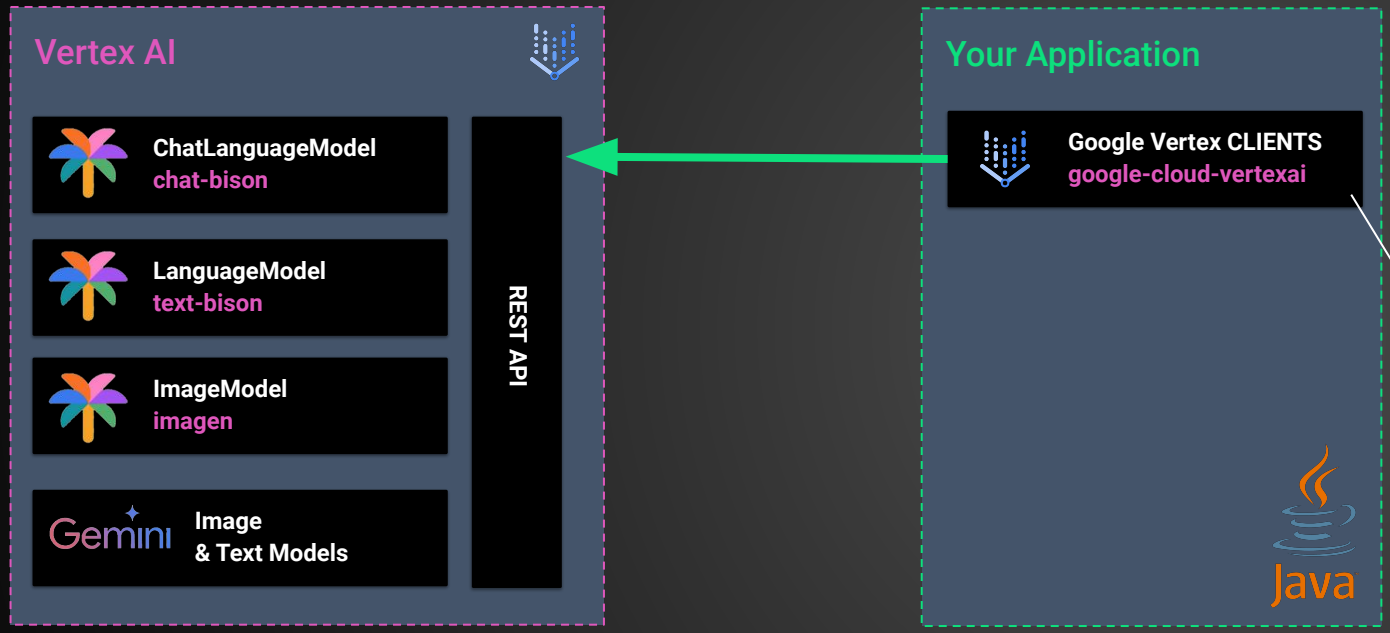
Gemini, PaLM, Codey, Imagen, Vertex AI...



Using Vertex AI in your Applications

demo

https://cloud.google.com/vertex-ai/generative-ai/docs/start/quickstarts/quickstart-multimodal?hl=en#gemini-beginner-samples-java



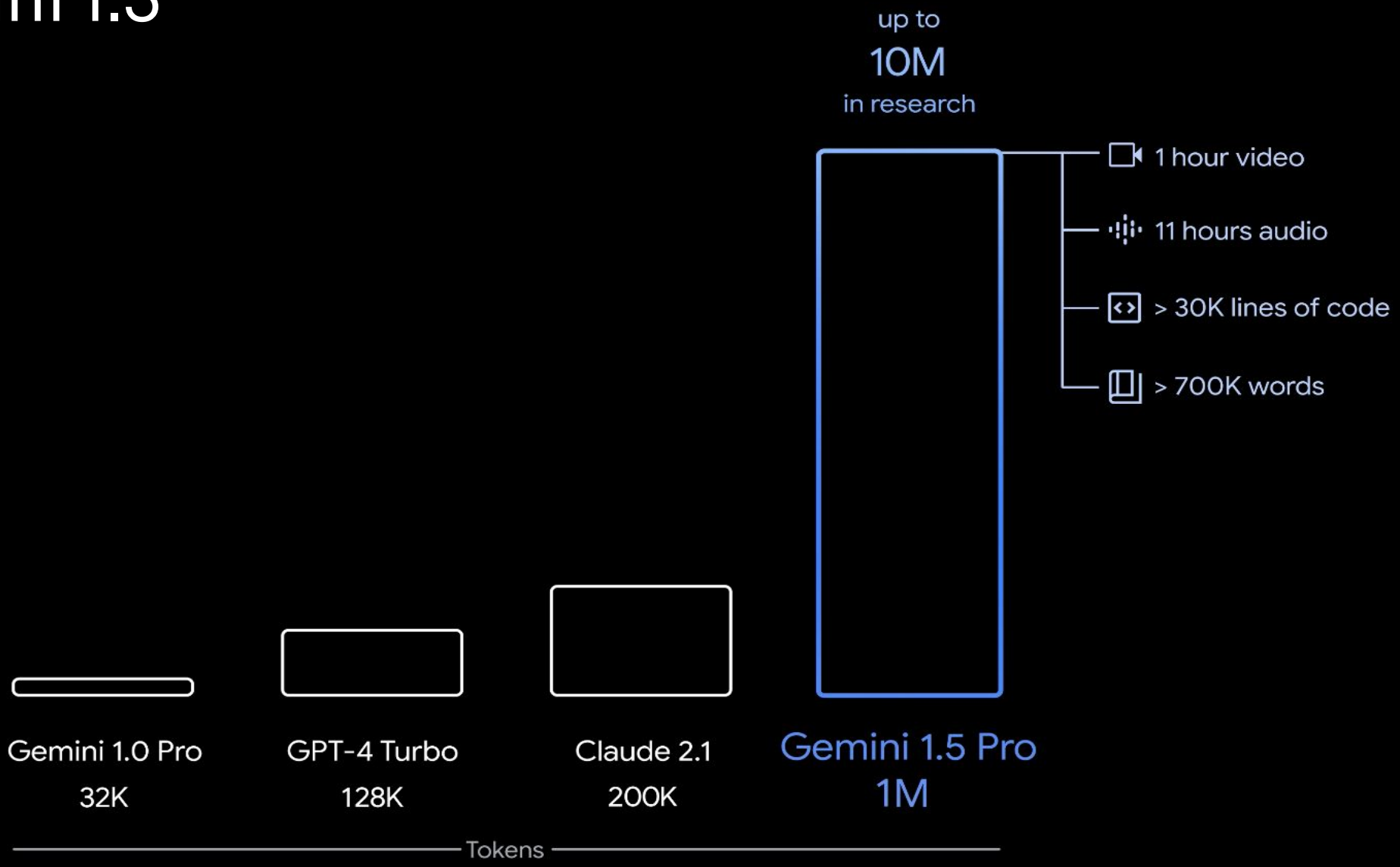
https://ai.google.dev/api/rest#rest-resource:-v1.models

```
<dependency>
 <groupId>com.google.cloud</groupId>
 <artifactId>google-cloud-vertexai</artifactId>
</dependency>
```

What is Gemini?

- **Gemini** is Google Deep Mind's most capable AI model
- It's a **multimodal** large language model: text, **images**, videos
- Comes in **3 sizes**: Nano, Pro, and Ultra
- Supports **function calling**
- Ranks at the top of the various LLM benchmarks *(general knowledge, translation, image understanding, reasoning, math, coding, and more...)*

Gemini 1.5


up to
10M
in research
1 hour video
11 hours audio
> 30K lines of code
> 700K words
Gemini 1.0 Pro
32K
GPT-4 Turbo
128K
Claude 2.1
200K
Gemini 1.5 Pro
1M
Tokens

demo

Let’s play with Gemini and Vertex AI Studio

- https://console.cloud.google.com/vertex-ai/model-garden
- https://ai.google.dev
- https://ai.google.dev/examples

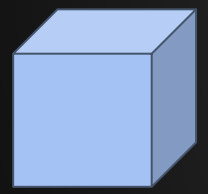
What goes under Gemini ?

How to customize a large model with Vertex AI

Simple, cost efficient

Complex, more expensive

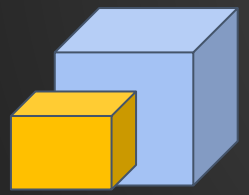
Prompt design



LLM

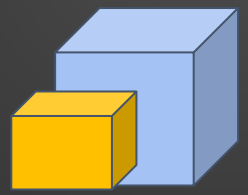
Prompt

Supervised Tuning (PEFT*)



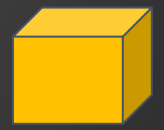
LLM with task-specific tuned parameters

Reinforcement Learning with Human Feedback (PEFT*)



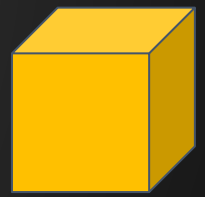
LLM with task-specific tuned parameters

Distillation step-by-step



Task-specific small model

Full fine tuning



Task-specific large model

*PEFT: Parameter Efficient Tuning

3.

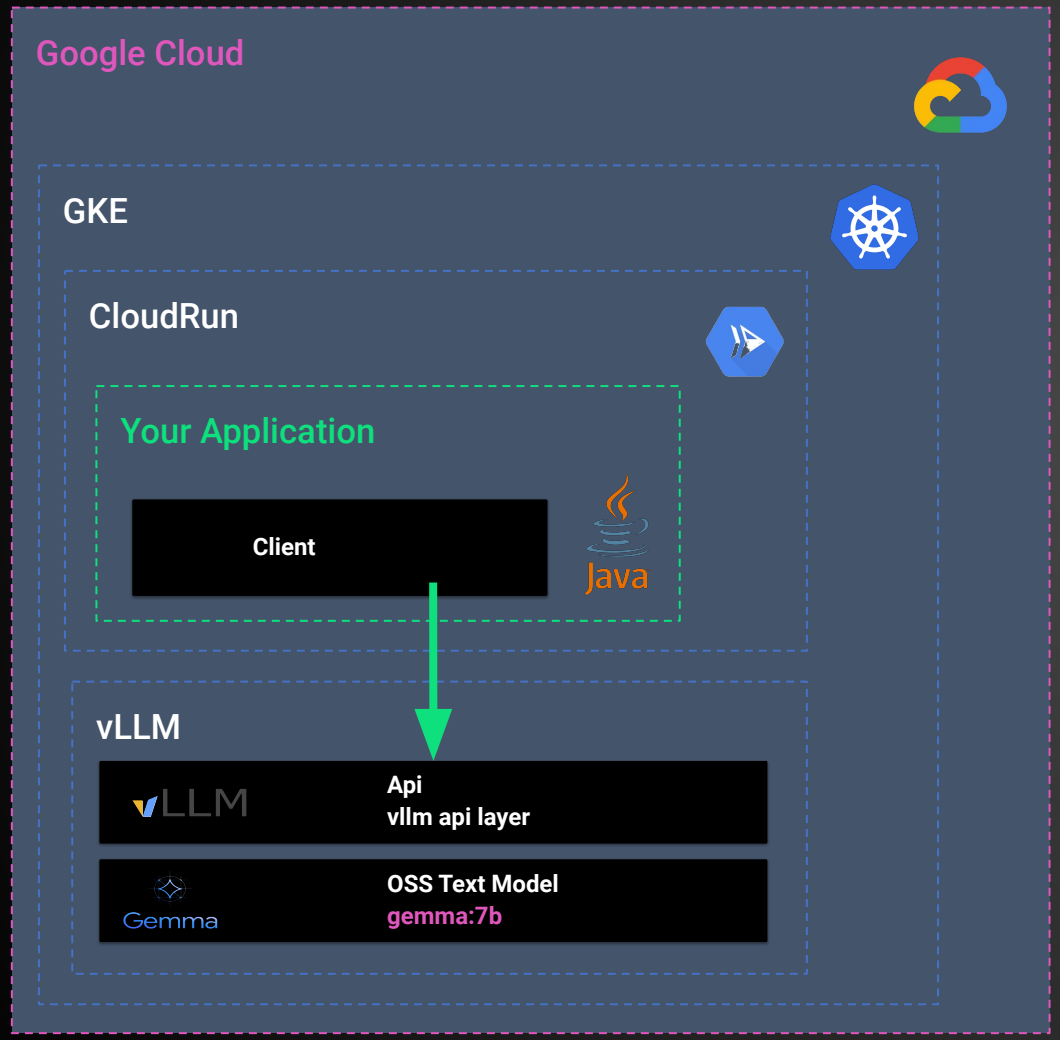
Deploy Local LLM
Gemma et GKE

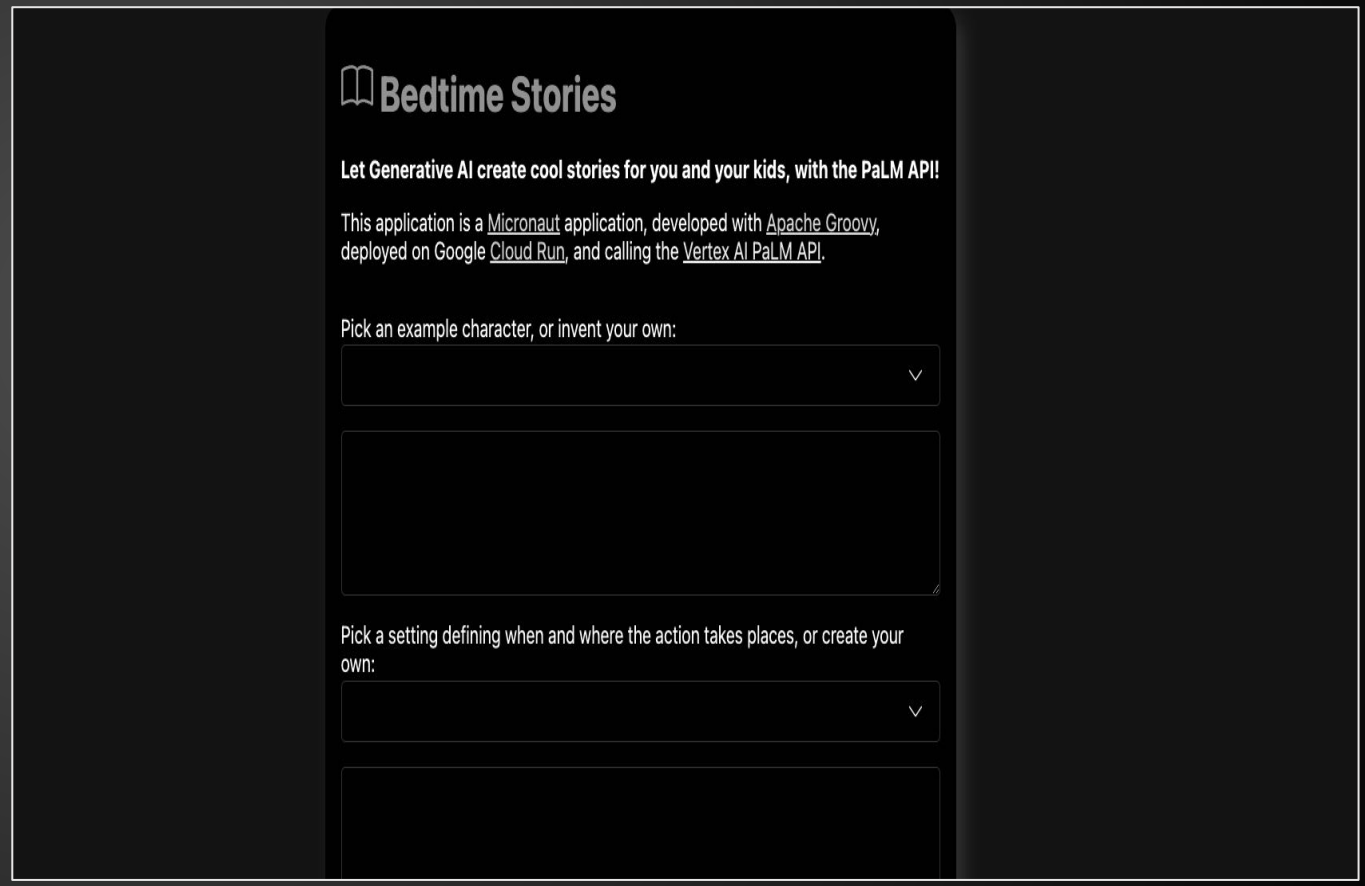
Gemma

What it Gemma ?
Why should you use it ?
how to deploy your OSS model to GKE ?
How to connect with Java applications ?

Demo bedtimes stories with Models on GKE

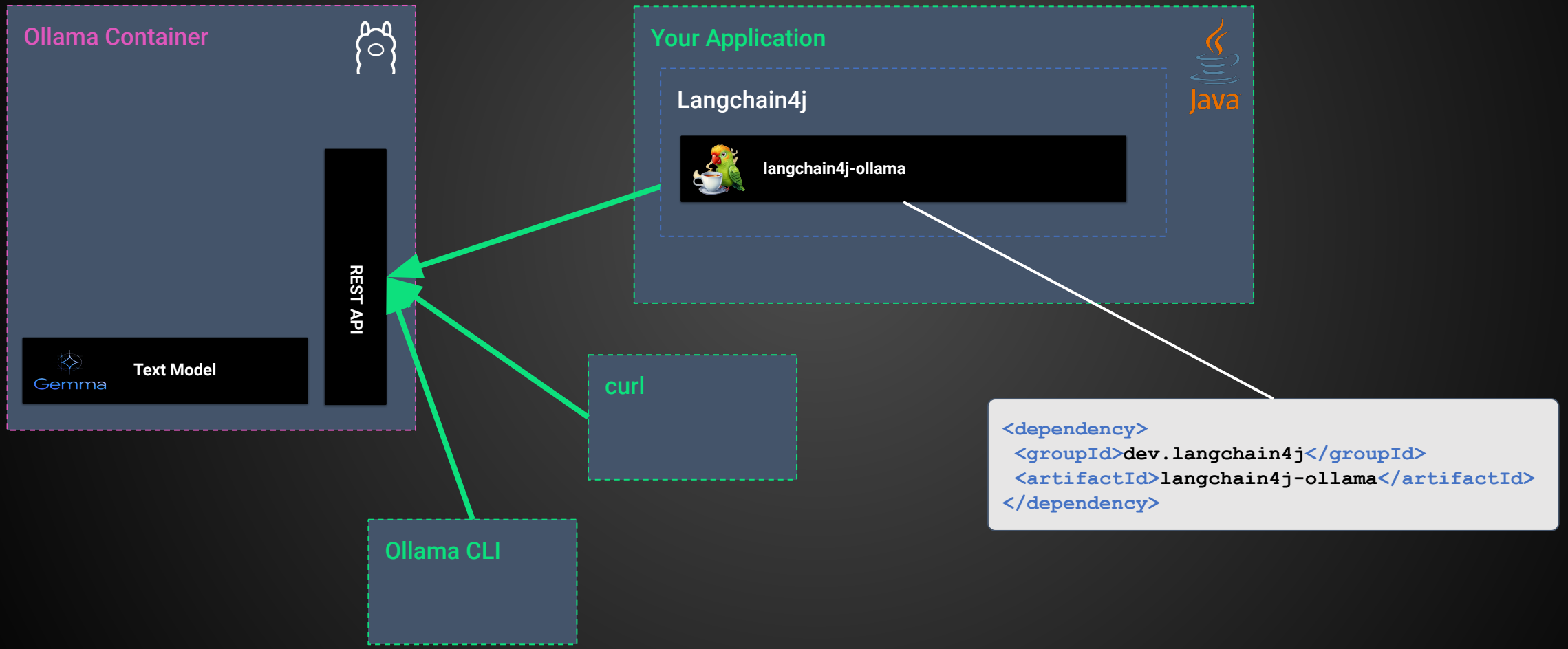
demo





Using (local) Gemma in your java applications

demo



4. Prompt Engineering

- Introduction
- Build effective Prompts
- Advanced techniques
- Best practices
- Limitations

Langchain4j: Build GenAI Apps with Java

Gemini

Language Model

text completion

ChatLanguage Model

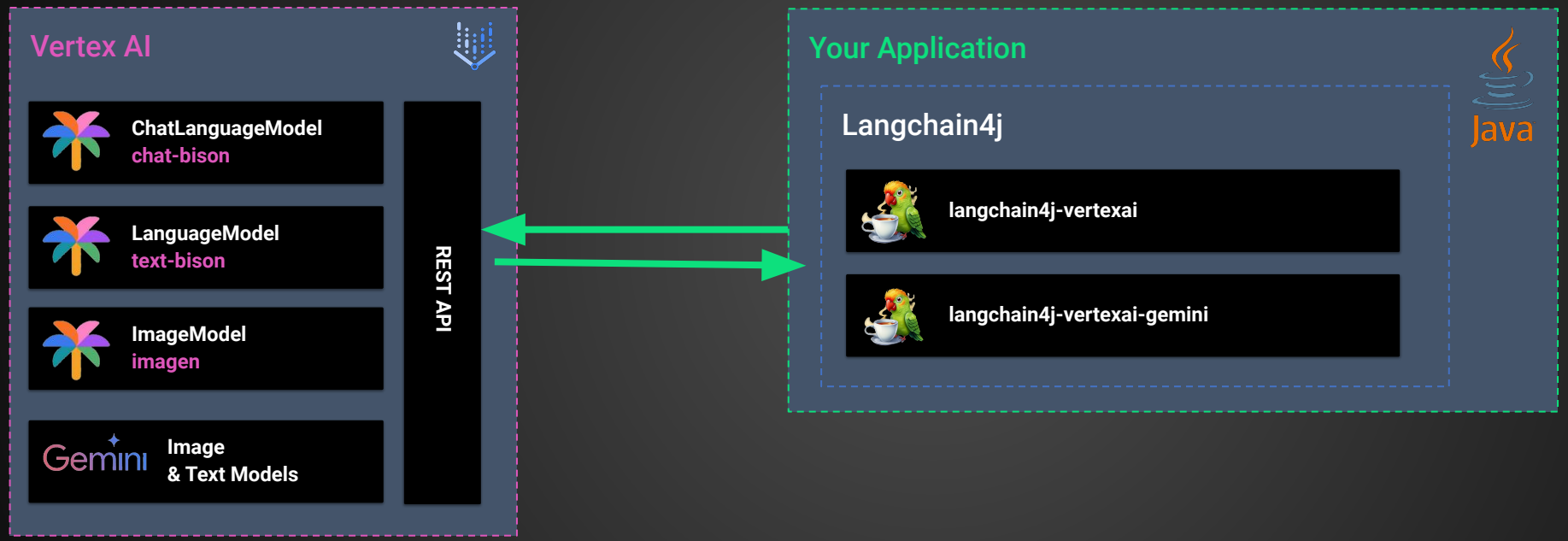
chat completion
vision-pro (multi modal)

ImageModel

text to image

Langchain4j and Vertex AI

demo



Spring AI: Build GenAI Apps with Java

Gemini








ChatClient

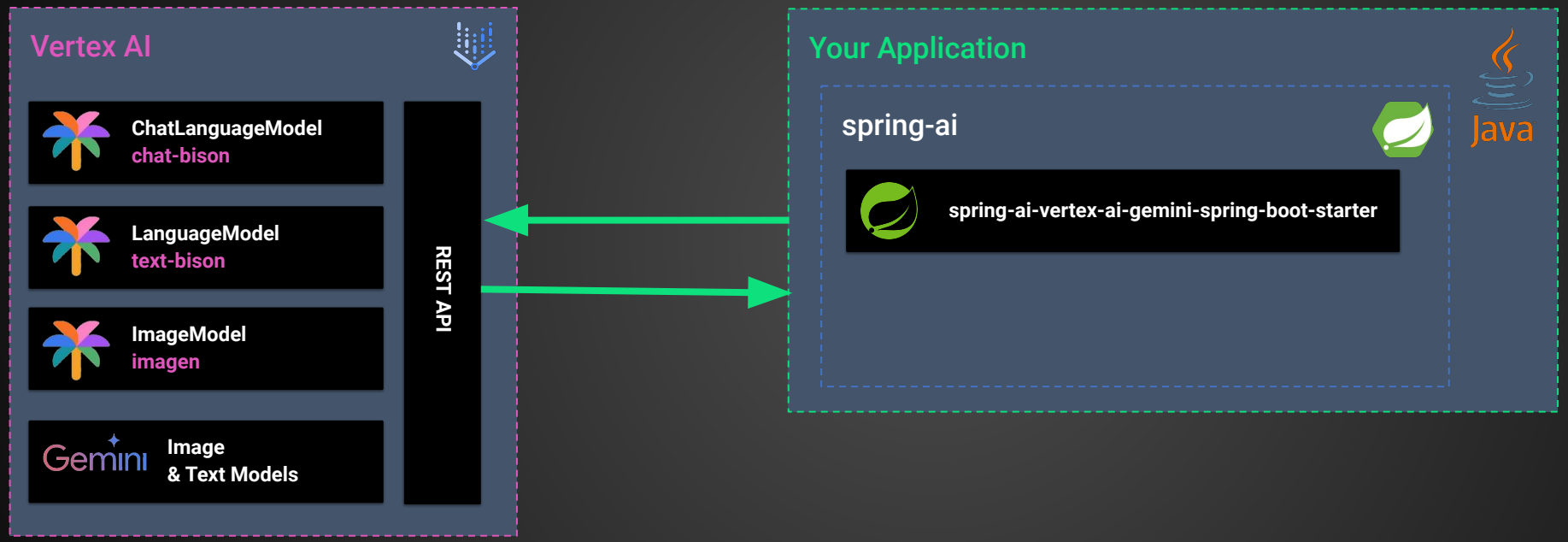
chat completion

ImageClient

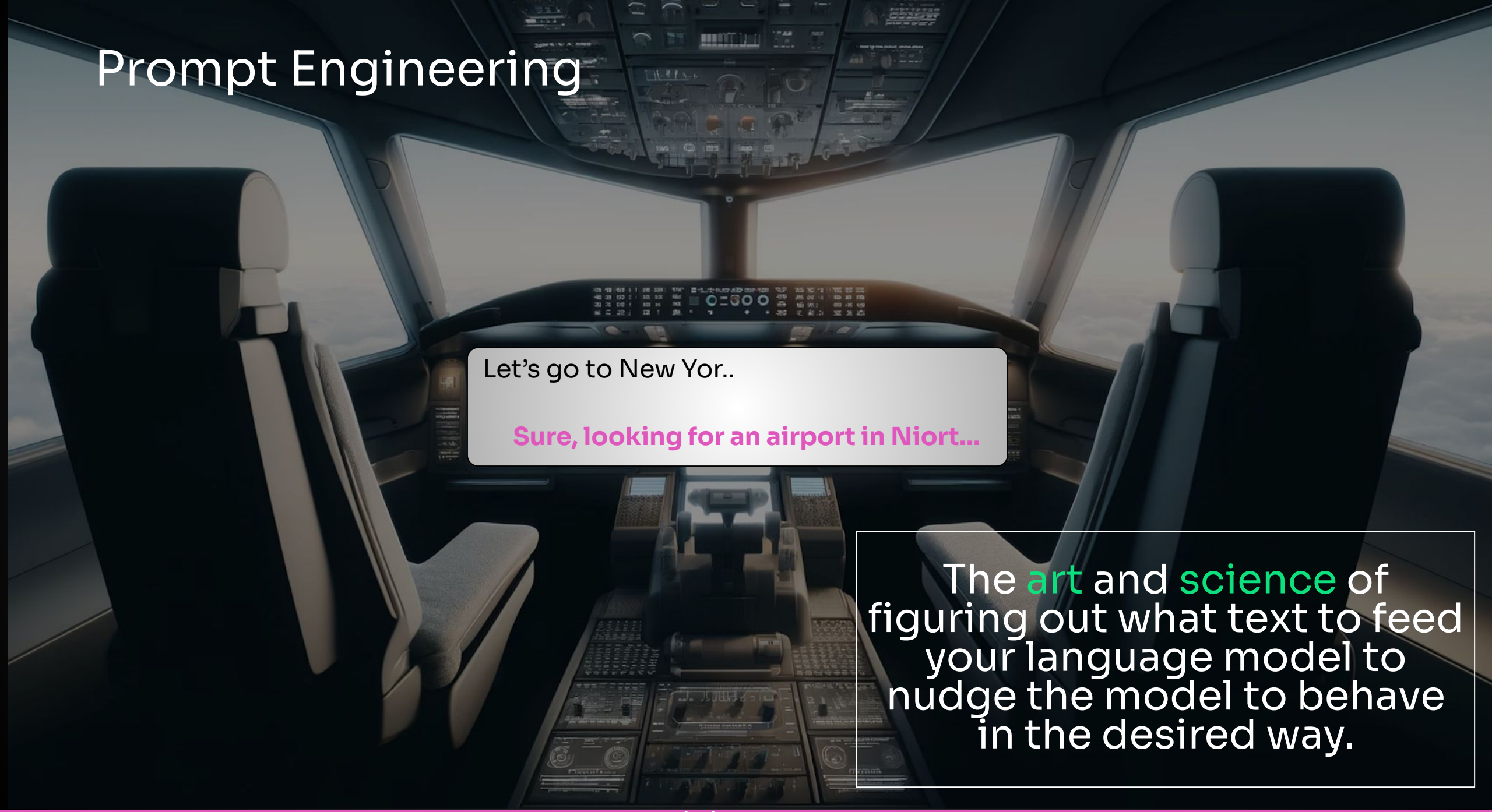
text to image


Spring AI and Vertex AI

demo



Prompt Engineering



The art and science of figuring out what text to feed your language model to nudge the model to behave in the desired way.

How to build effective prompts ?

CONTEXT

[Roles, Persona, Audience] : You are an assistant targeting Java developers

[Objectives] : Your mission is to provide helpful answers

[Constraints] : Format, Style, Must have, Boundaries

[Question] **(inputs)** Question, Task, Entity, Completions

SAMPLE

[Techniques] One-shot Prompt, few shots prompts, check questions

[RAG] Your documents

Tune your requests

TEMPERATURE

Tune the degree of randomness.

1

More Creative tasks
- Content Generation
- Can hallucinate more

More Accurate tasks
- Summarization
- Q&A

0

TOP P

Smallest set of words whose cumulative probability >= P

```
p = .8
```

```
java          .51
ia            .23
langchain     .11
spring        .08
…
```

TOP K

Smallest set of words whose cumulative probability >= P

```
K = 2
```

```
java          .51
ia            .23
langchain     .11
spring        .08
…
```

TOKENS

Size of the generated response.

PRO

- **Detailed/In-Depth**
- **Comprehensive**
- **Completion**

CONS

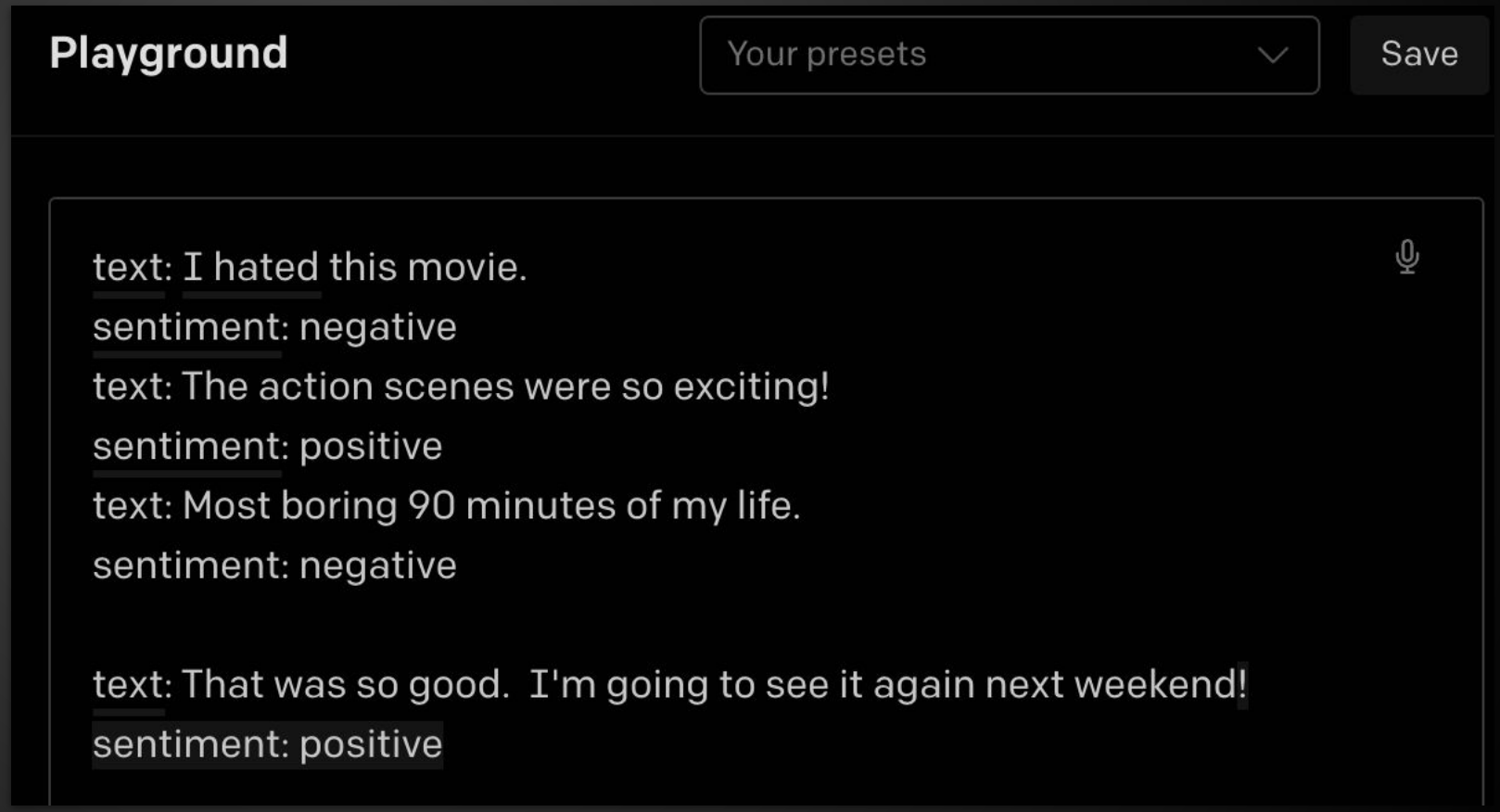
- **Lower Precision**
- **Processing Time**
- **Cost**
- **Repetitions**
- **Milleniums** 😜

Advanced Techniques
Few-shot Learning

Providing LLMs with a small number of examples is enough for them to learn specific tasks.

Different Names

- zero-shot prompting
- one-shote prompting
- few-shot prompting

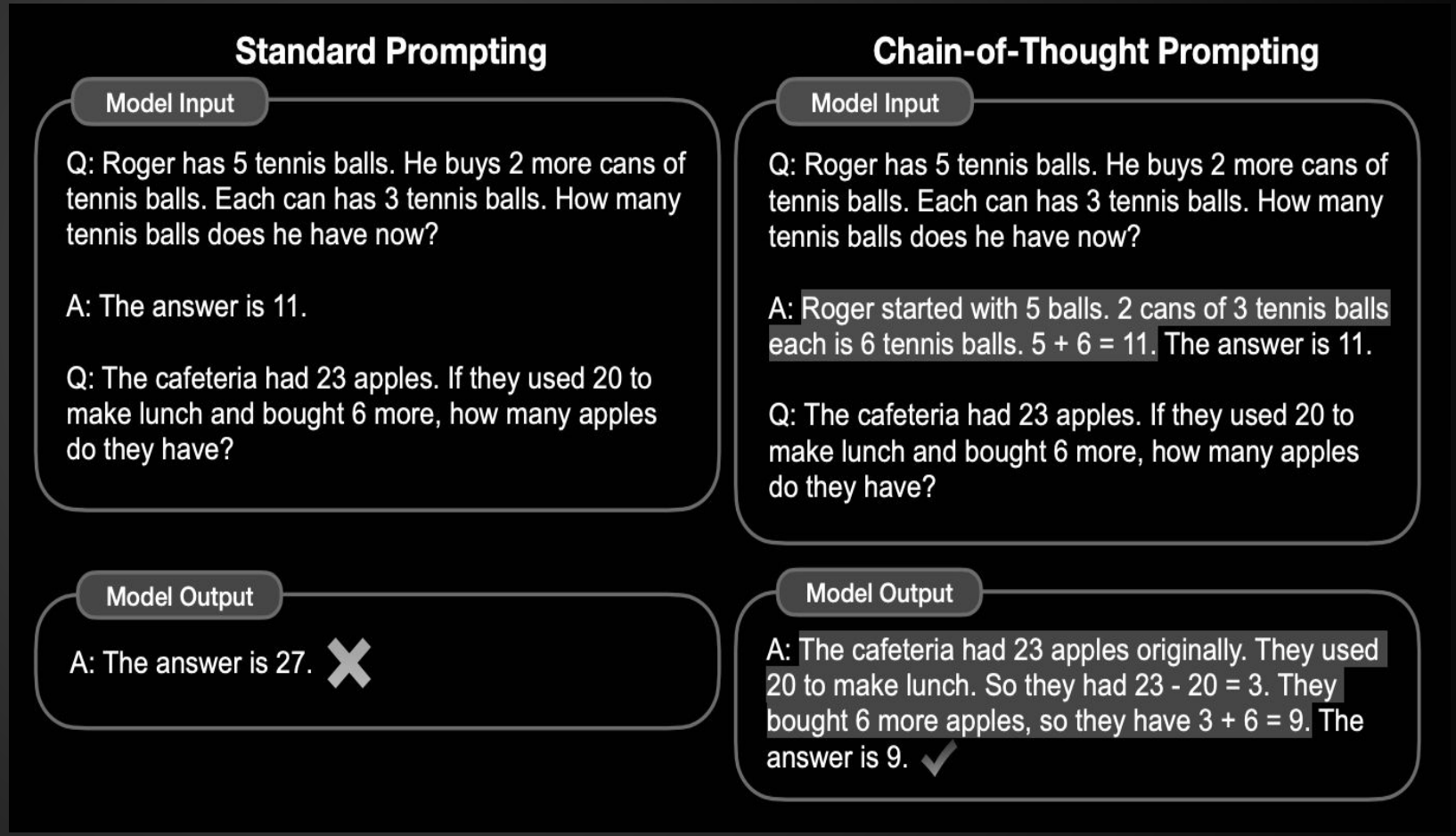


Advanced Techniques
Chain of Thoughts

LLM Can reason

Pro

- Easy
- Effective on multiple tasks



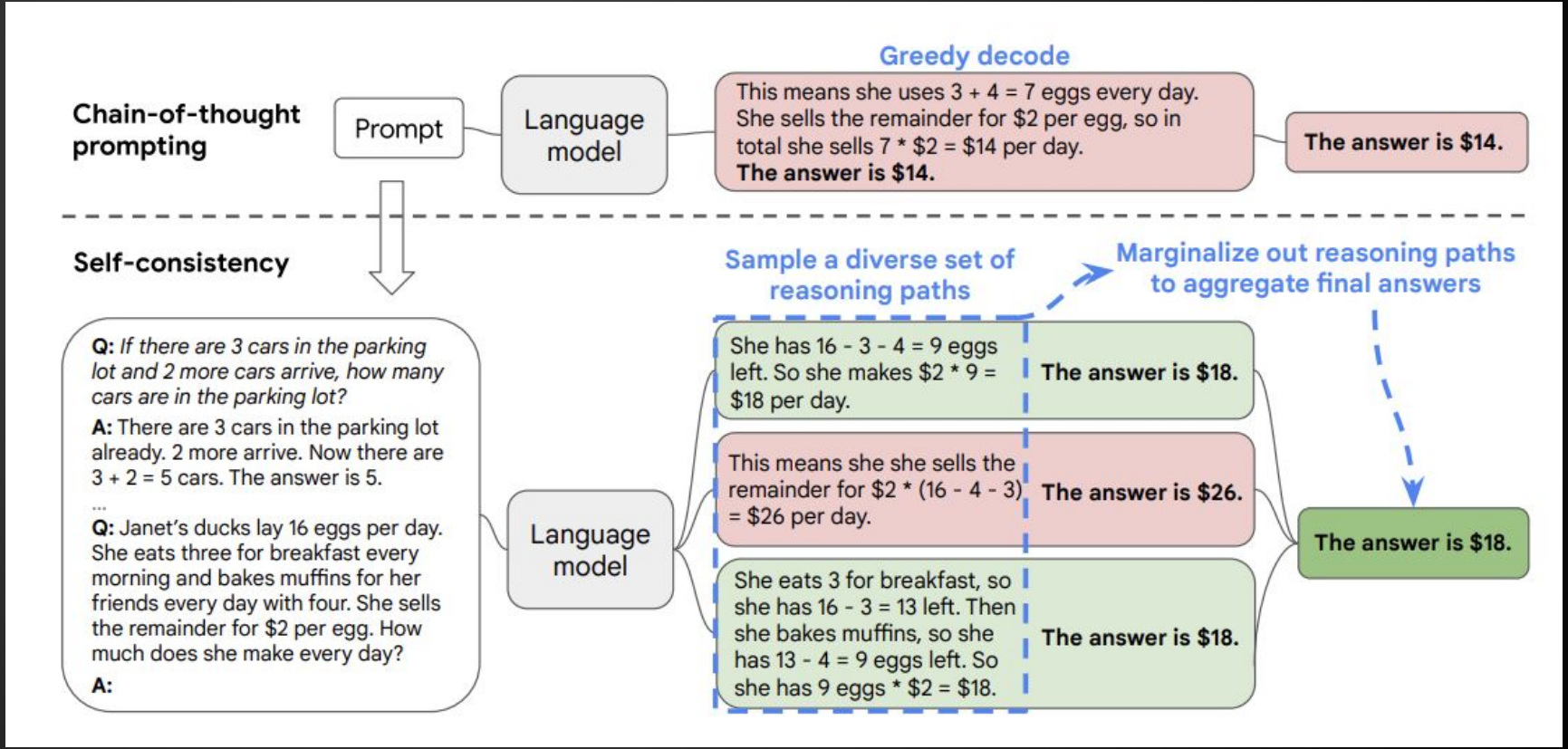
Advanced Techniques
CoT + Self Consistency

Sampling multiple reasoning paths can produce better results.

March 2022

How does it work in practice?

- Task needs to have a correct answer
- Different reasoning paths need to be explored
- Effectively achieved by increasing temperature, top p, and top k parameters with LLMs
- Need to sample 5-20 reasoning paths



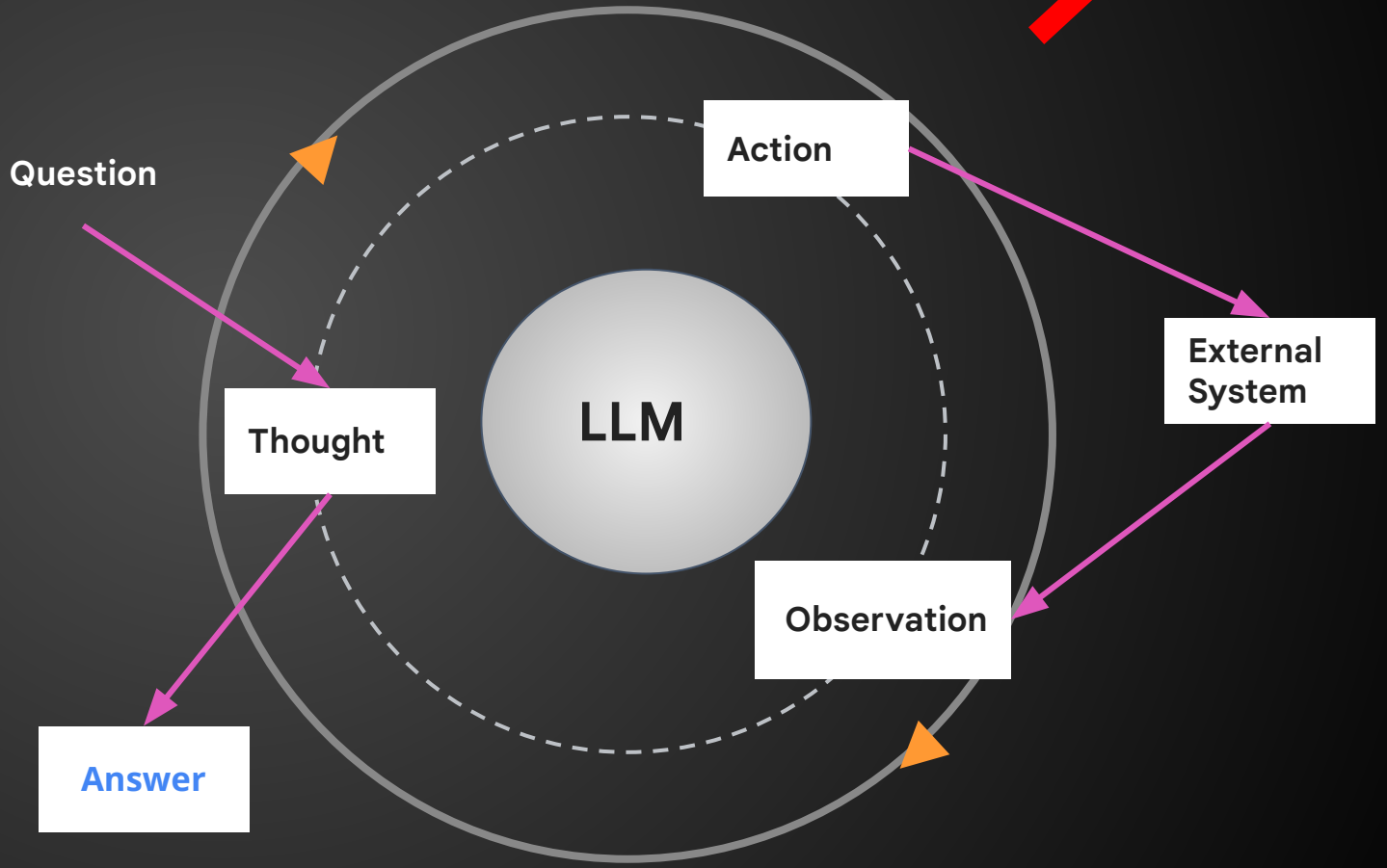
Advanced Techniques
ReAct

ReAct short for Reasoning and Acting

Combines chain of thought and tool usage together to reason through complex tasks by interacting with external systems

ReAct is particularly useful if you want the LLM to reason and take action on external systems

Used to improve the accuracy of LLMs when answering questions



Prompt Best Practices 1/2

1. 1 examples worth 100 instructions
2. DARE Determine Appropriate Response
   a. Rôles, Personas, Audience => your vision
   b. Objectives => your mission
   c. Scope: If you do not know...say it
3. Adapt temperature to tasks (creative vs accurate)
4. Use detailed natural language to unveil chain of prompts
5. Structured your prompts && orders Matters

Prompt Best Practices 2/2

6. Responsible AI and Filters

7. Test, Measure, Improve, Repeat

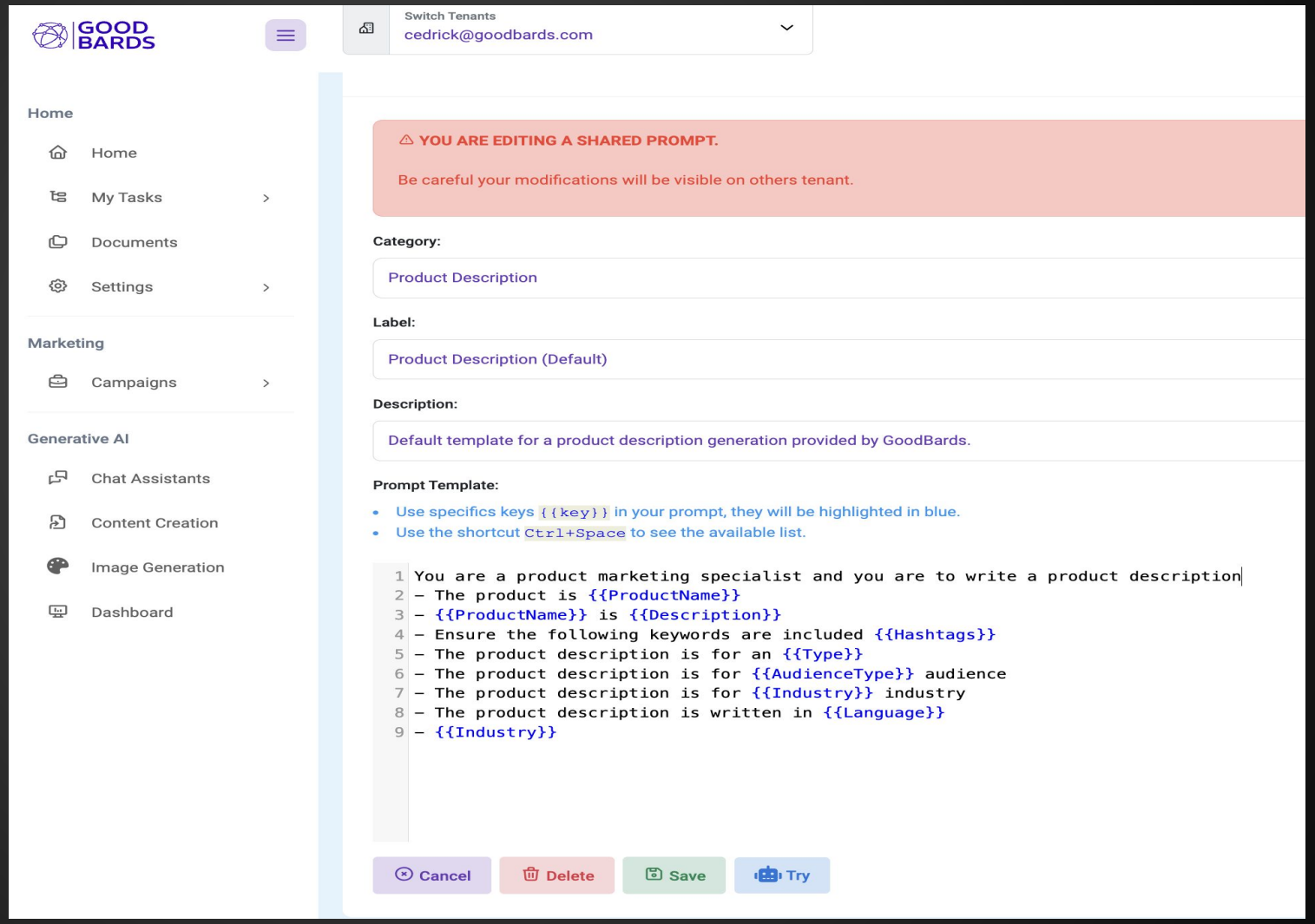
8.Be specific, no open questions

9. Review from multiple people

10. Detailed algorithms and reasoning problems

Prompt Templates with Java

- Langchain4j
- SpringAI
- GoodBards



demo

Can we do better ?
Limitations of Prompt Engineering

- LLM .....can be outdated
- LLM .....Does not know *your* data
- LLM .....is not tuned = hard steerability
- LLM .....Hallucinating if not properly prompted
- LLM .....works with limited Input windows (tokens)

=> You want to use your own documents & knowledge based.

=> Retrieval-Augmented Generation

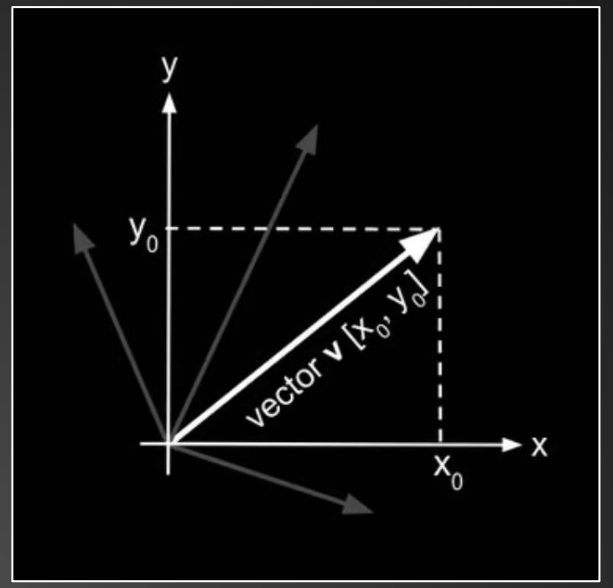
5. Vector Search

- Vector Search
- Vector Databases
- jVector
- Apache Cassandra™ & AstraDB
- Vector Stores

Vector

Denote a phenomenon with a direction and a length .

Formulated as a list made of numbers (components), list length is the dimensionality (d)



The "length" (or norm)

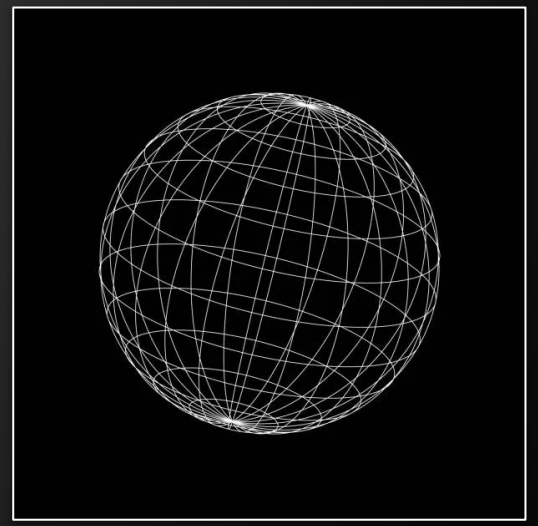
regardless which direction

some meaningful notion of "rotation"

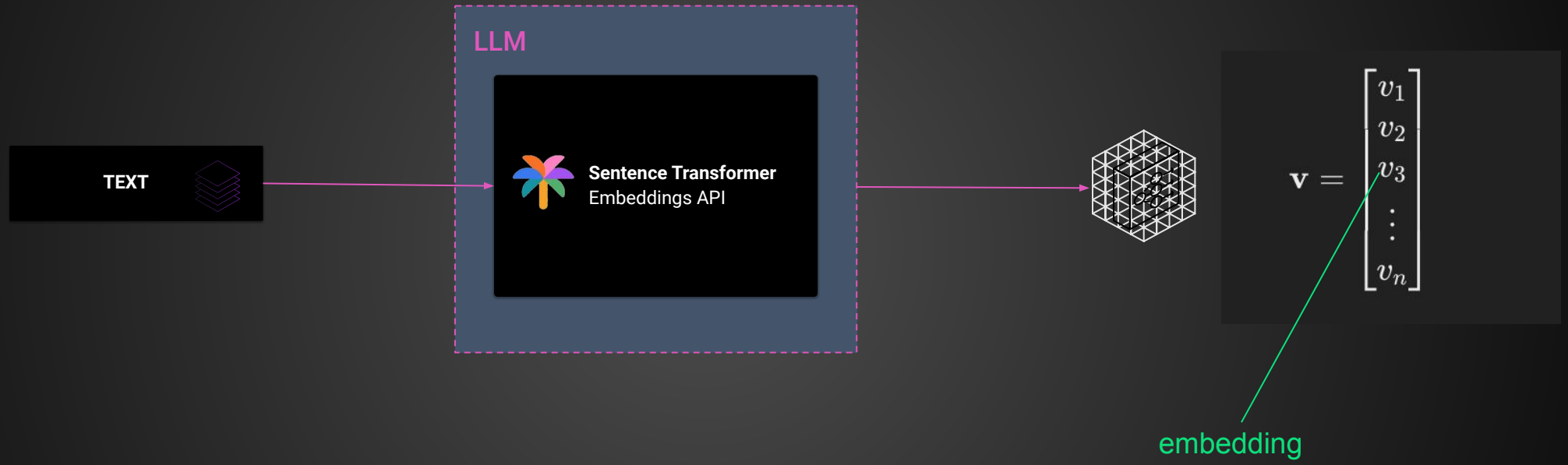
$$|\mathbf{v}| = \sqrt{\sum_i v_i^2}$$

With 3 dimensions, space representing by | v | = 1 ?

the unit sphere

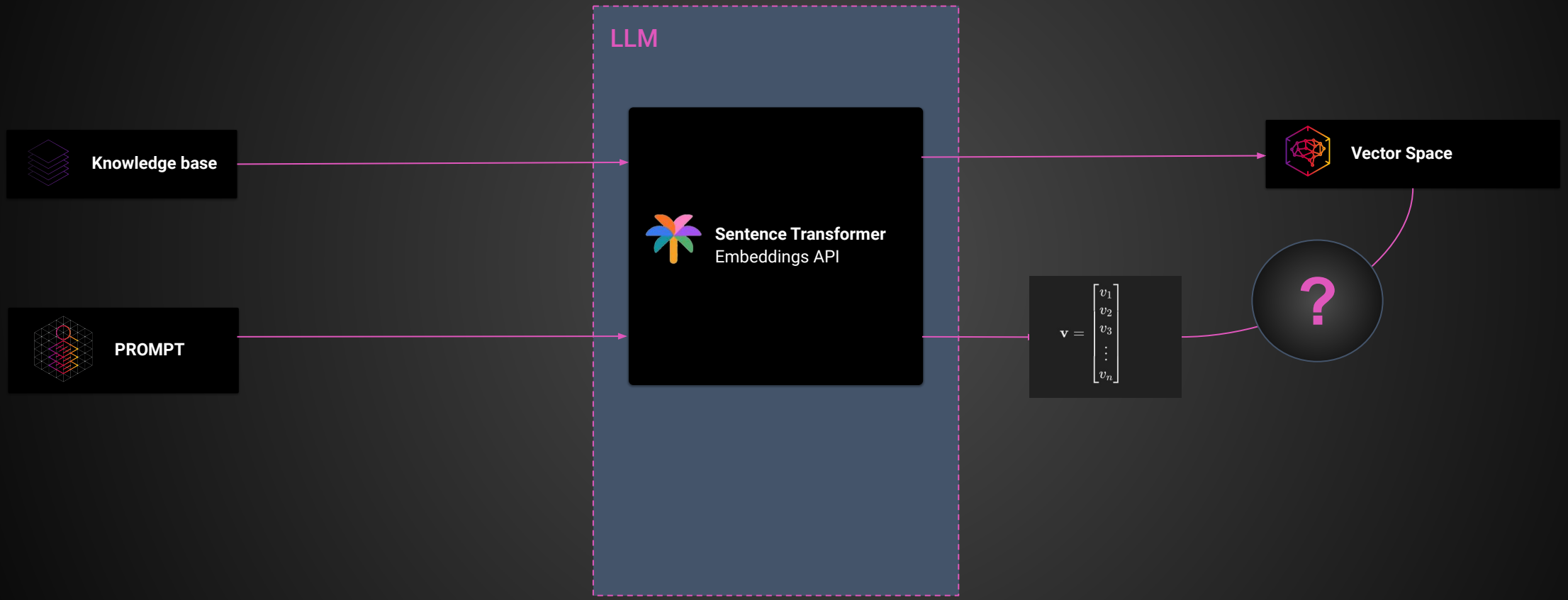


“Vectorization” (embed)



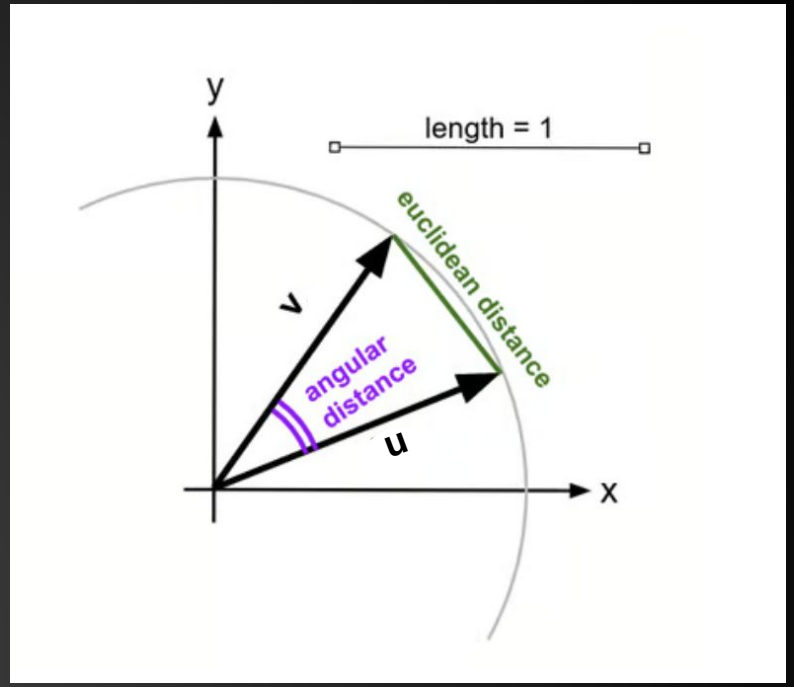
Vector *search*

demo



Vector *Similarity*

A numeric way to quantify how much two vectors v and u are close to each other, computed with some formula S(*v*, *u*).



Euclidean distance (L2)

$$d(\mathbf{v}, \mathbf{u}) = \sqrt{\sum_{i=1}^{n}(v_i - u_i)^2}$$

Angular distance, cosine similarity

$$\text{cosine similarity}(\mathbf{v}, \mathbf{u}) = \frac{\mathbf{v} \cdot \mathbf{u}}{\|\mathbf{v}\|\|\mathbf{u}\|}$$

$$\mathbf{v} \cdot \mathbf{u} = \sum_{i=1}^{n} v_i u_i$$

$$\|\mathbf{v}\| = \sqrt{\sum_{i=1}^{n} v_i^2}$$

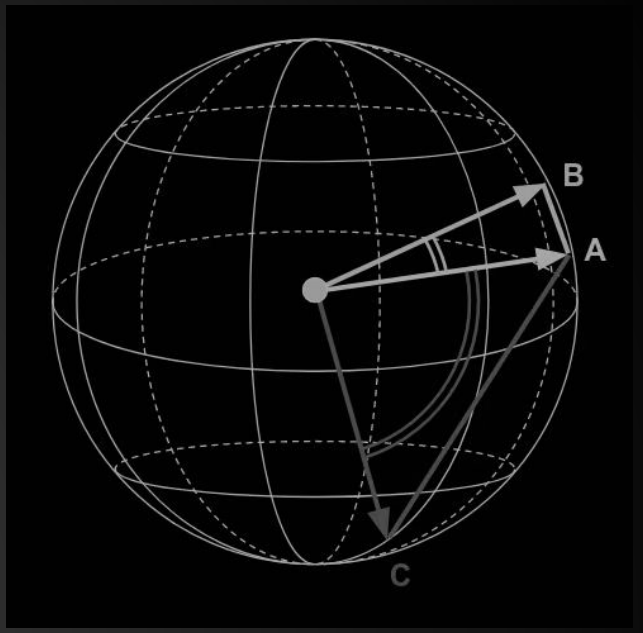
$$\|\mathbf{u}\| = \sqrt{\sum_{i=1}^{n} u_i^2}$$

Not all Vectorial Databases use same formulae

Similarity name	Definition (Cassandra / Astra DB)	Remarks
Euclidean	$S_{\text{eucl}}(\mathbf{x},\mathbf{y}) = \frac{1}{1+\sum_i (x_i-y_i)^2} = \frac{1}{1+\delta^2_{\text{eucl}}(\mathbf{x},\mathbf{y})}$	based on the Euclidean distance, $\delta_{\text{eucl}}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_i (x_i-y_i)^2}$
Cosine	$S_{\cos}(\mathbf{x},\mathbf{y}) = \frac{1+\frac{\sum_i x_i y_i}{\lvert\mathbf{x}\rvert\lvert\mathbf{y}\rvert}}{2} = \frac{1+\frac{\mathbf{x}\cdot\mathbf{y}}{\lvert\mathbf{x}\rvert\lvert\mathbf{y}\rvert}}{2} = \frac{1+S^*_{\cos}(\mathbf{x},\mathbf{y})}{2}$	a rescaling of the **S*cos** found on some textbooks (see later)
Dot-product	$S_{\text{dot}}(\mathbf{x},\mathbf{y}) = \frac{1+\sum_i x_i y_i}{2} = \frac{1+\mathbf{x}\cdot\mathbf{y}}{2}$	rarely the right choice, except on the unit sphere!

On the sphere, life is easy

On the sphere, no matter what similarity you use, you get the same top results in the same order.



$$\cos(\angle AB) > \cos(\angle AC)$$
$$\vec{a} \cdot \vec{b} > \vec{c} \cdot \vec{b}$$

Vector representation

Vector $\vec{AB} = \vec{b} - \vec{a}$

Vector $\vec{CB} = \vec{b} - \vec{c}$

$$\|\vec{b} - \vec{a}\|^2 = (\vec{b} - \vec{a}) \cdot (\vec{b} - \vec{a})$$
$$\|\vec{b} - \vec{a}\|^2 = \vec{b} \cdot \vec{b} - 2(\vec{a} \cdot \vec{b}) + \vec{a} \cdot \vec{a}$$
$$\|\vec{b} - \vec{a}\|^2 = 2(1 - \vec{a} \cdot \vec{b})$$

Distance dot product

$$\vec{a} \cdot \vec{b} = \|\vec{a}\|\|\vec{b}\| \cos(\theta)$$

Since $\vec{a} \cdot b > \vec{c} \cdot b$, we can conclude that:

$\|\vec{AB}\|^2 < \|\vec{CB}\|^2$

$\|\vec{AB}\| < \|\vec{CB}\|$

Distance L2

$$\text{distance}(A, B) = \|\vec{AB}\| = \|\vec{b} - \vec{a}\|$$
$$\text{distance}(C, B) = \|\vec{CB}\| = \|\vec{b} - \vec{c}\|$$

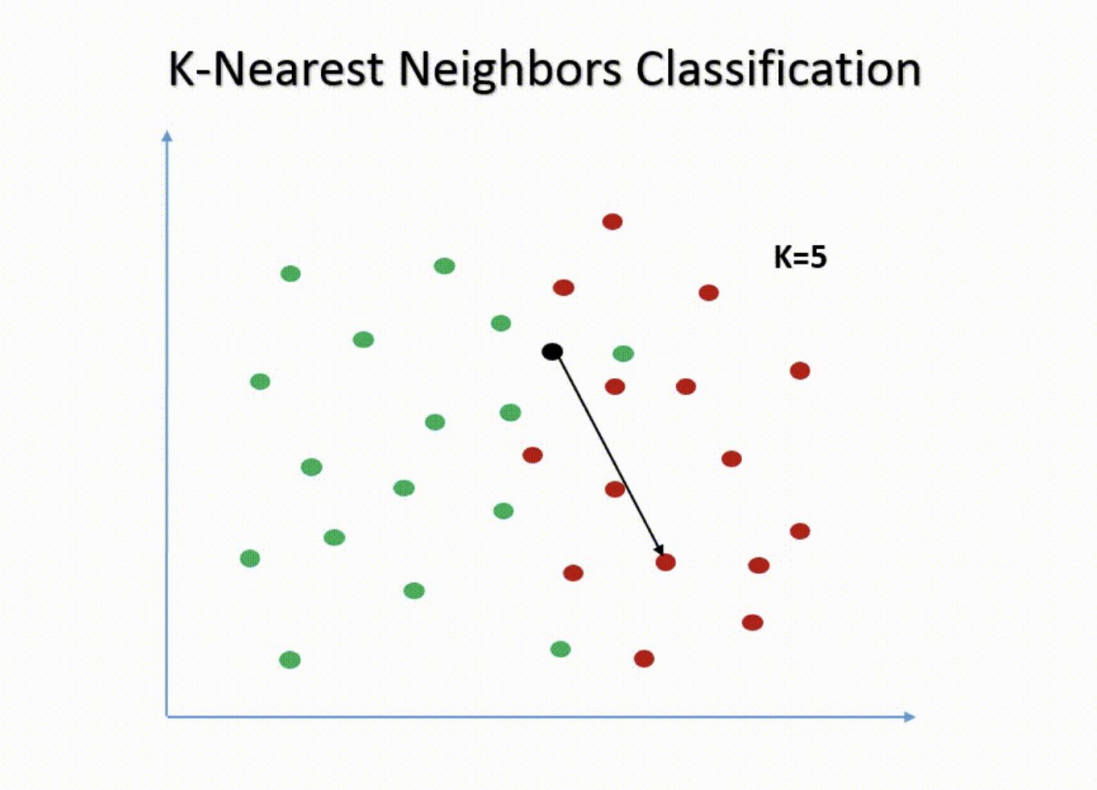
$$\angle AB < \angle AC \Rightarrow AB < AC.$$

When to use a metric or another ?

measure	domain	notes
Euclidean	sphere (unit-norm vectors)	Consider switching to Cosine or Dot-product (just the numeric similarities change, ordering unchanged)
Cosine	sphere (unit-norm vectors)	If you know you're on a sphere, switch to Dot-product (the only change is faster computation)
Dot-product	sphere (unit-norm vectors)	The best-performance measure on the sphere
Euclidean	arbitrary vectors	Use this if every aspect of the vector (incl. the norm) carries useful information
Cosine	arbitrary vectors	Consider normalizing vectors to unit norm at ingestion time and, once on the sphere, switching to Dot-product
Dot-product	arbitrary vectors	Probably not what you want (ensure there's a strong reason to fall in this case); performance may be tricky

Vector Search : KNN

k-nearest neighbours

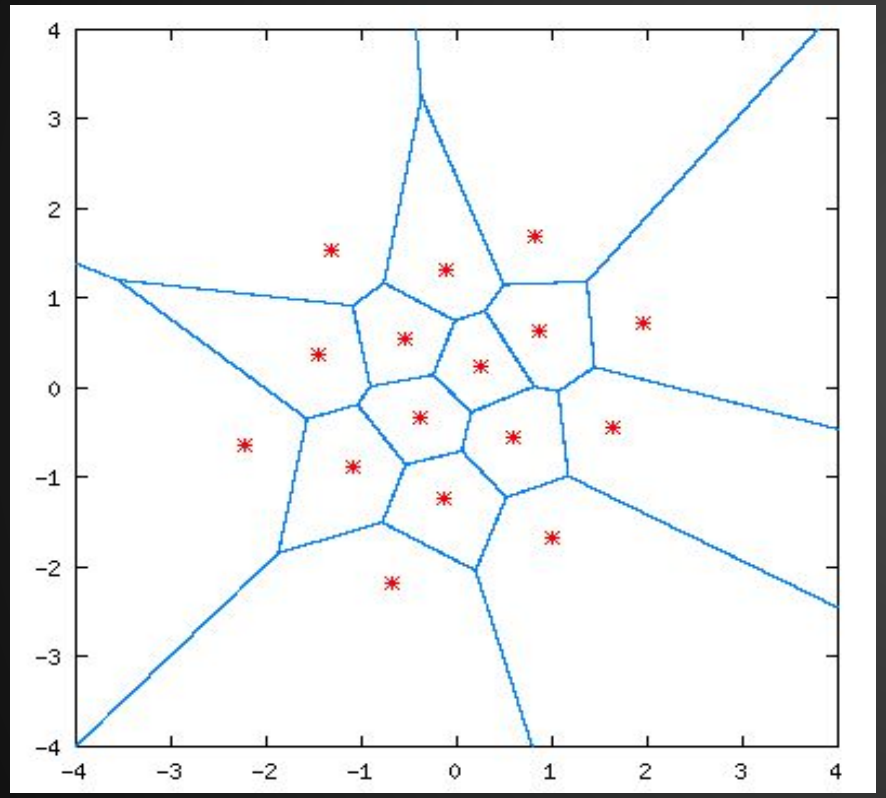


https://machinelearningknowledge.ai/k-nearest-neighbor-classification-simple-explanation-beginners/

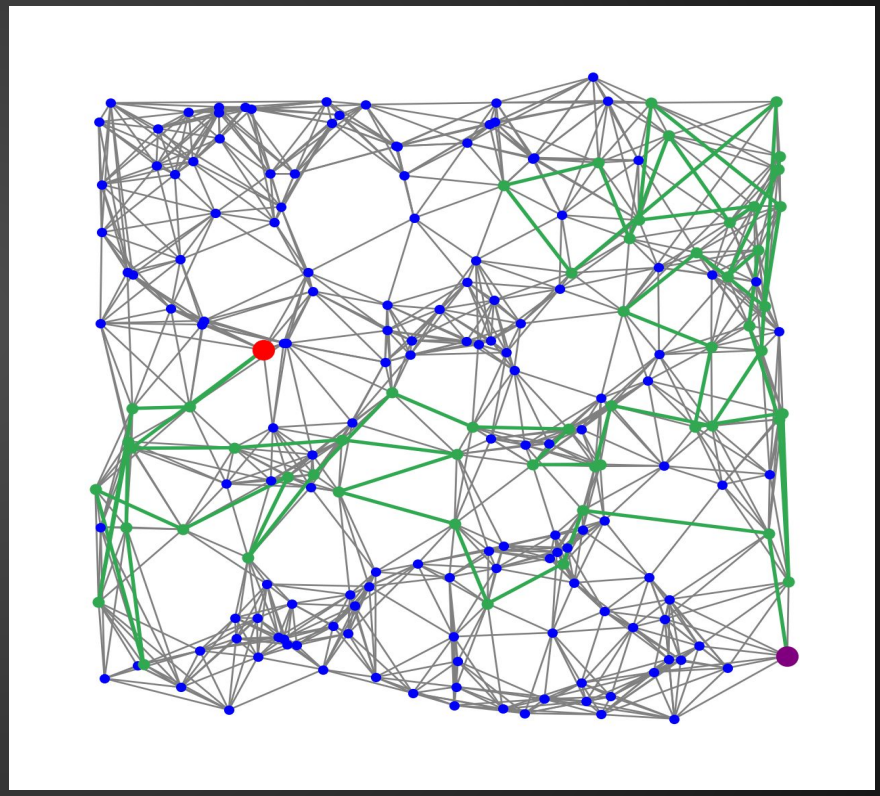
- Exhaustive
- Linear Complexity
- 1000+ dimensions
- Millions of Vector
- Good Luck !

Vector Search: ANN

Approximate nearest neighbours

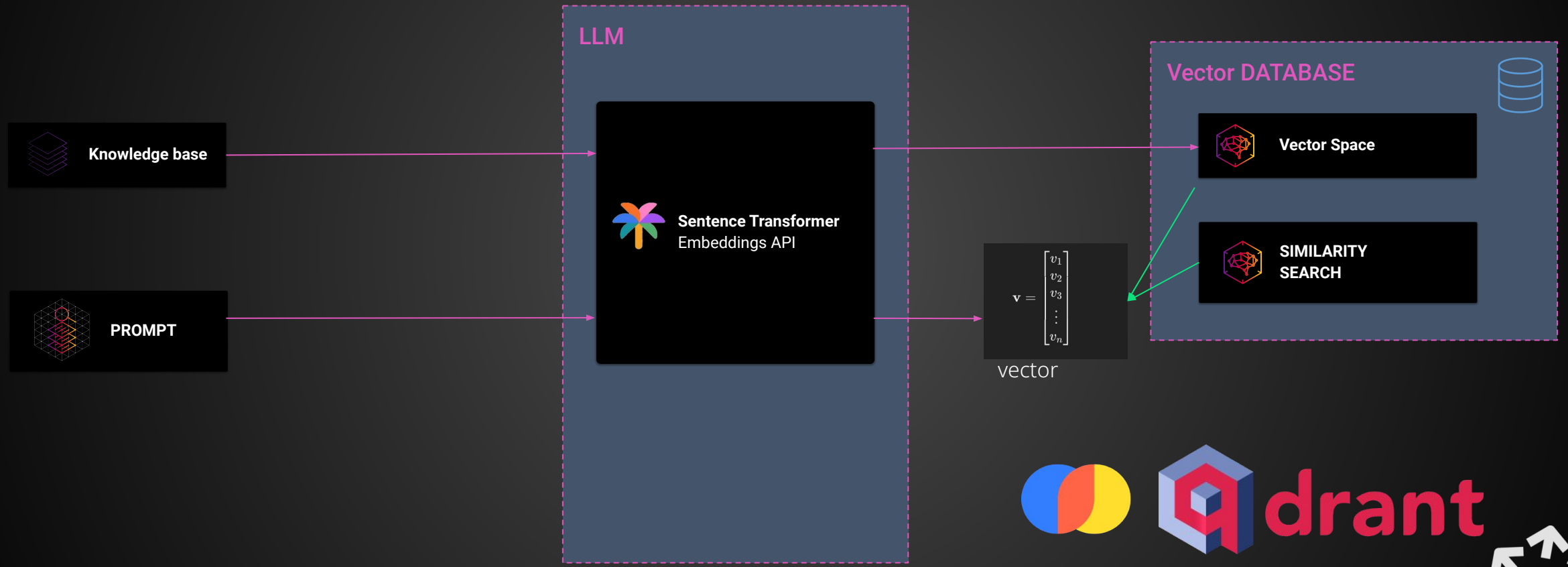


Partition

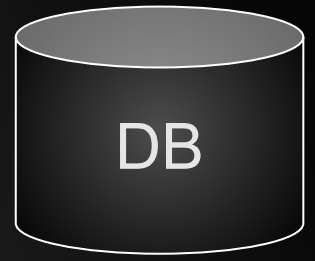


Graph

Vector *Database*



“Vectors” in existing *Databases ?*



- An organized collection of related data items
- Database Management System (DBMS)
- Navigational, hierarchical, network databases
- Relational databases
  - Row-oriented for online transaction processing or OLTP
  - Column-oriented or columnar for online analytical processing or OLAP
- Object databases
- NoSQL databases
- Multi-model databases


Tabular or wide-column


Document


Key-value

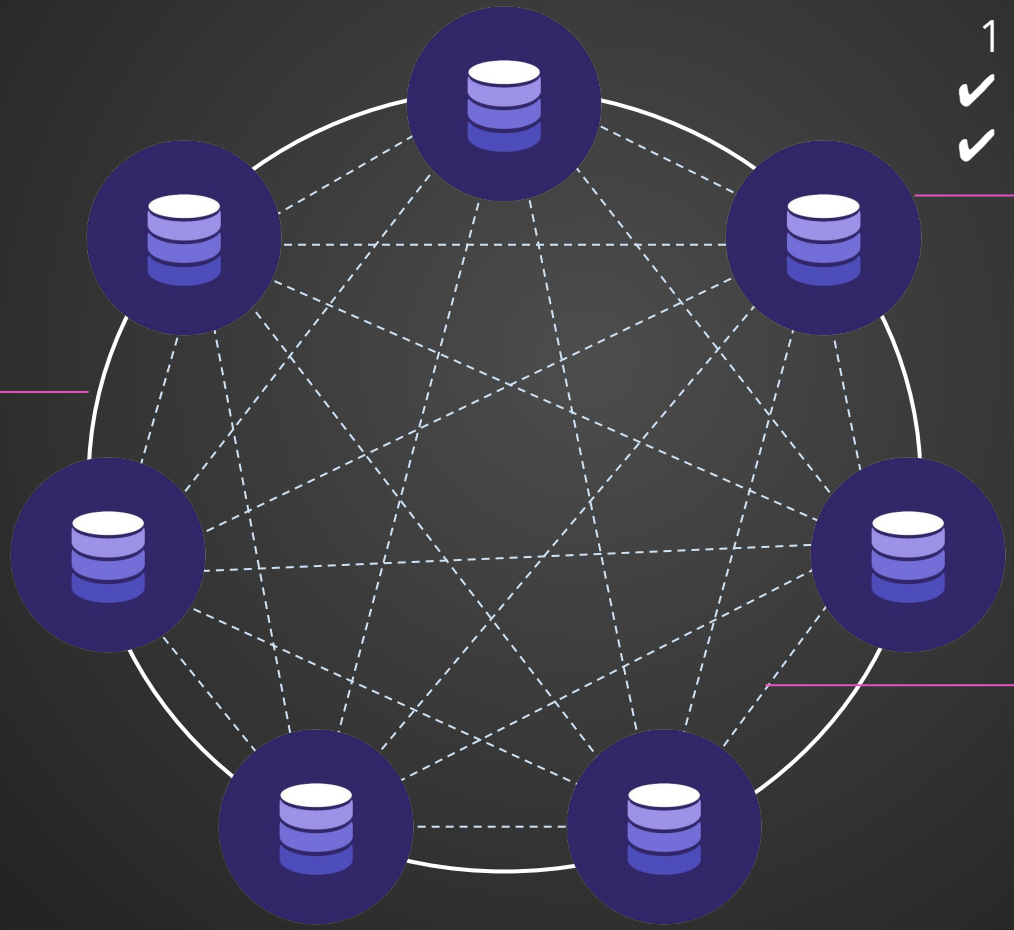

Graph

Apache Cassandra™

Undisputed Leader for Scale and reliability

Apache Cassandra™

NoSQL Distributed database



1 Installation = 1 NODE

- ✔ Capacity = ~ 2-4TB
- ✔ Throughput = LOTS Tx/sec/core

DataCenter (DC) | Ring

Communication:

- ✔ Gossiping
- ✔ No Master (peer-to-peer)

Apache Cassandra™

NoSQL Distributed database

High Availability

Always On

Every second of downtime translates into lost revenue

Linear Scalability

Hyper Scalability

Millions of operations per day, hour, or second

Low Latency

Faster Pace

Every millisecond of latency has consequence

Global Distribution

Data Everywhere

On-premises, hybrid, multi-cloud, centralized, or edge

Cassandra 5 as a Vector database

New Model

- New Vector type introduced

```
CREATE TABLE IF NOT EXISTS vsearch.products (
  id int PRIMARY KEY,
  name TEXT,
  description TEXT,
  item_vector VECTOR<FLOAT, 5> //5-dimensional embedding
);
```

Cassandra 5 as a Vector database

SAI Secondary indices

```
CREATE CUSTOM INDEX IF NOT EXISTS ann_index
ON vsearch.products(item_vector)
USING 'StorageAttachedIndex';
```

Cassandra 5 as a Vector database

Sample Neighbour Search

```
SELECT * FROM vsearch.products
ORDER BY item_vector ANN OF [0.15, 0.1, 0.1, 0.35, 0.55]
LIMIT 1;
```

Cassandra 5 as a Vector database

Integration

Global ANN everywhere

```
SELECT * FROM demo
ORDER BY
  embedding ANN OF ?
LIMIT 10
```

Composes with partitioning

```
SELECT * FROM demo
WHERE partition_id = ?
ORDER BY
  embedding ANN OF ?
LIMIT 100
```

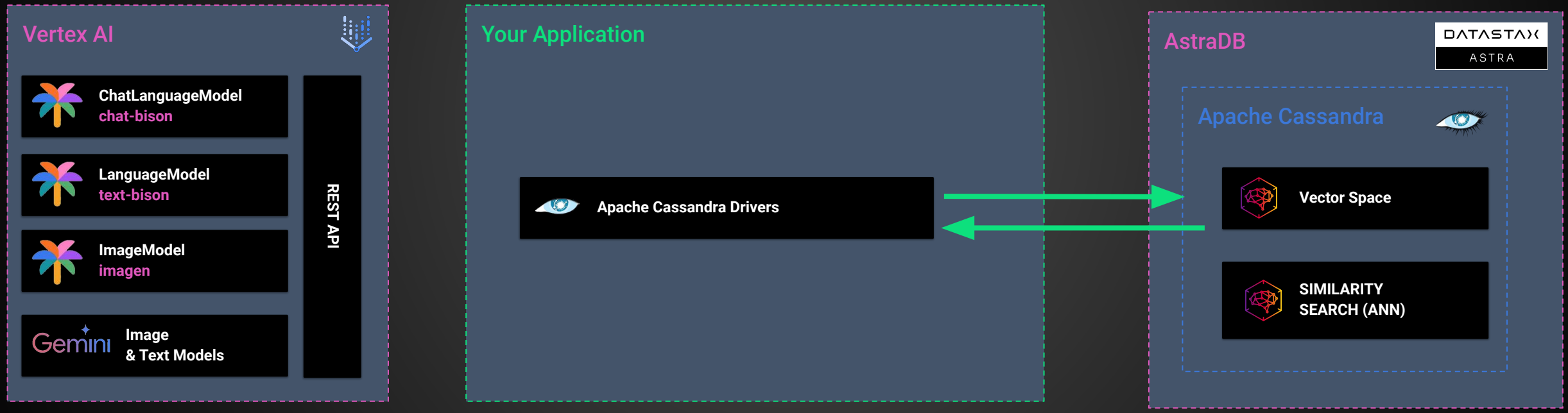
Composes with other SAI indexes

```
SELECT * FROM demo
WHERE (c1 = ? AND c2 = ?)
   OR c3 = ?
ORDER BY
  embedding ANN OF ?
LIMIT 5
```

Cassandra 5 as a Vector database

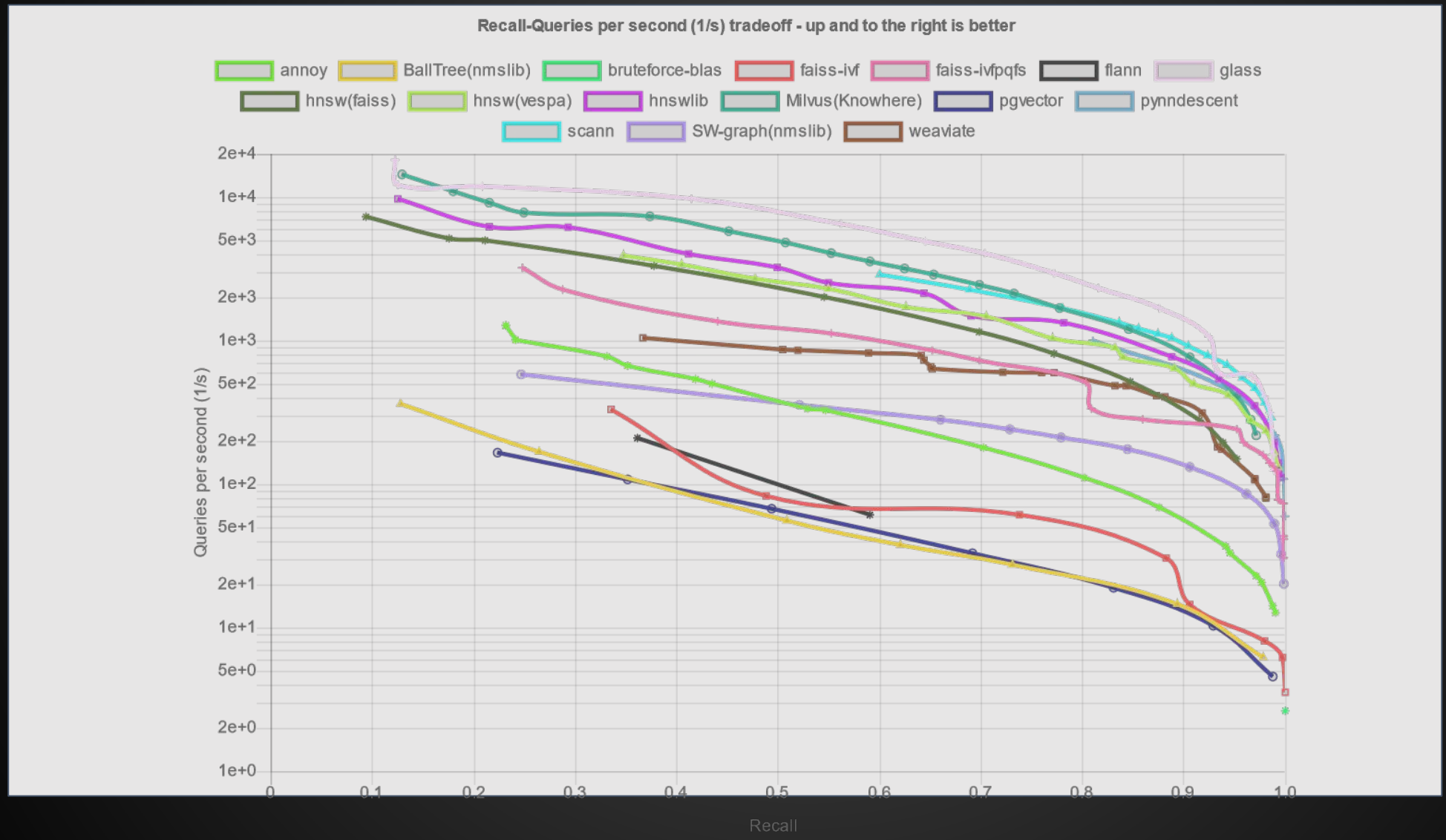
Cassandra Drivers

demo



Vector ANN Search Benchmark

benchmarks

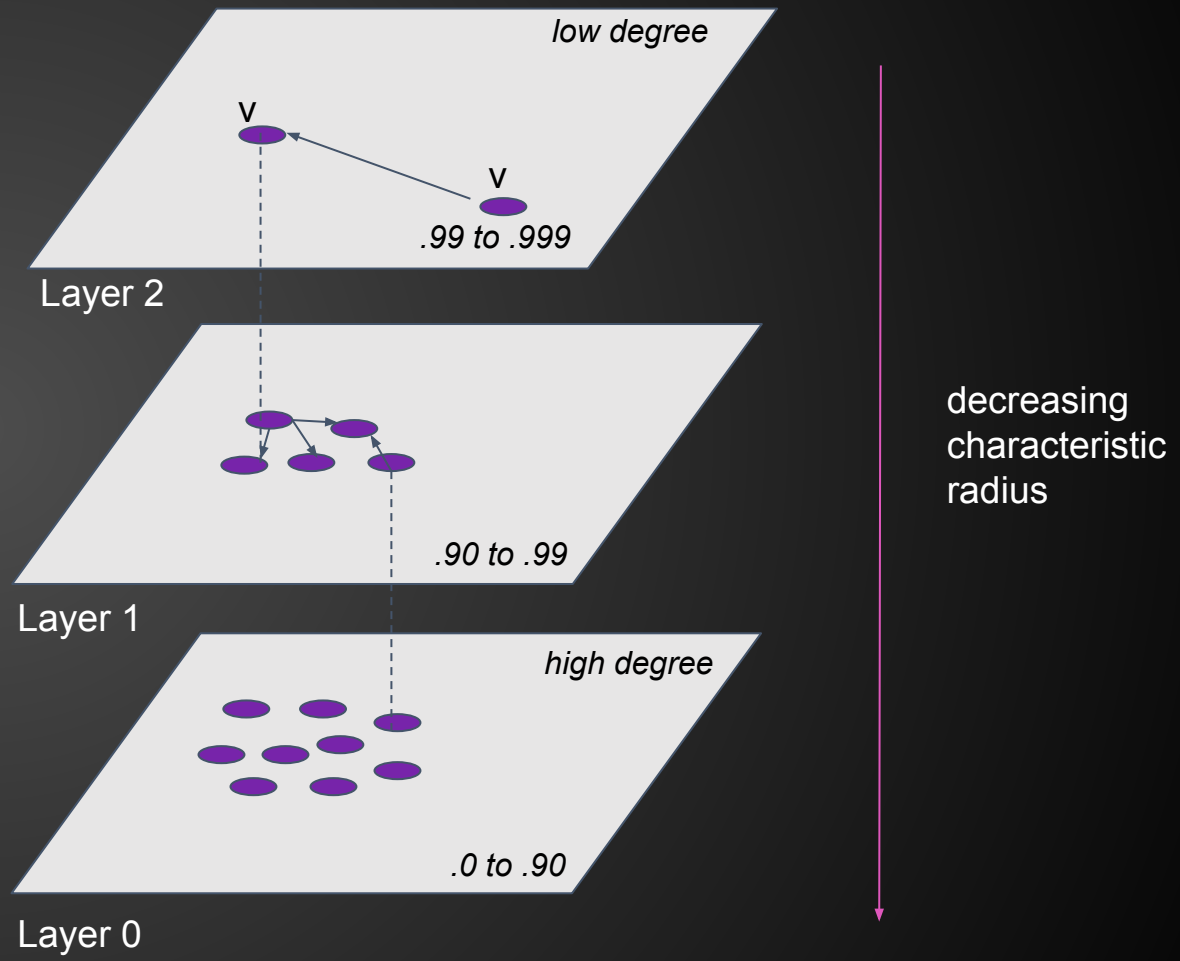


HNSW – Hierarchical Navigable Small World

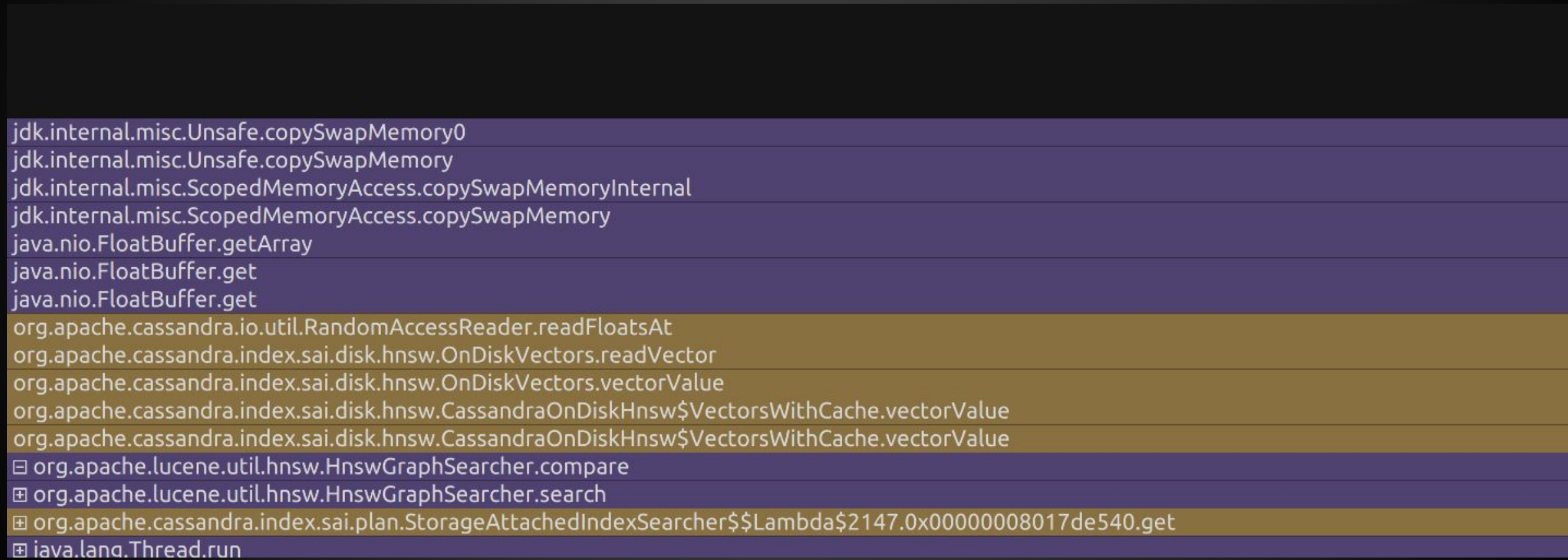
"Vector Index"

As seen in:

- Lucene (*Elastic, Solr, OpenSearch, MongoDB*)
- Weaviate
- Qdrant
- PGVector (August 2023)



HSNW on Cassandra with Wikipedia



jVector: State of the art

SPANN: Highly-efficient Billion-scale Approximate Nearest Neighbor Search

Qi Chen[1,*] **Bing Zhao**[1,2,†] **Haidong Wang**[1] **Mingqin Li**[1] **Chuanjie Liu**[1,3,†]
Zengzhong Li[1] **Mao Yang**[1] **Jingdong Wang**[1,4,*,†]
[1]Microsoft [2]Peking University [3]Tencent [4]Baidu
[1]{cheqi, haidwa, mingqli, jasol, maoyang}@microsoft.com
[2]its.bingzhao@pku.edu.cn [3]liu.chuanjie@outlook.com [4]wangjingdong@outlook.com

Abstract

The in-memory algorithms for approximate nearest neighbor search (ANNS) have achieved great success for fast high-recall search, but are extremely expensive when handling very large scale database. Thus, there is an increasing request for the hybrid ANNS solutions with small memory and inexpensive solid-state drive (SSD). In this paper, we present a simple but efficient memory-disk hybrid indexing and search system, named SPANN, that follows the inverted index methodology. It stores the centroid points of the posting lists in the memory and the large posting lists in the disk. We guarantee both disk-access efficiency (low latency) and high recall by effectively reducing the disk-access number and retrieving high-quality posting lists. In the index-building stage, we adopt a hierarchical balanced clustering algorithm to balance the length of posting lists and augment the posting list by adding the points in the closure of the corresponding clusters. In the search stage, we use a query-aware scheme to dynamically prune the access of unnecessary posting lists. Experiment results demonstrate that SPANN is 2× faster than the state-of-the-art ANNS solution DiskANN to reach the same recall quality 90%

DiskANN: Fast Accurate Billion-point Nearest Neighbor Search on a Single Node

Suhas Jayaram Subramanya* Carnegie Mellon University suhas@cmu.edu
Devvrit* University of Texas at Austin devvrit.03@gmail.com
Rohan Kadekodi* University of Texas at Austin rak@cs.texas.edu

Ravishankar Krishaswamy Microsoft Research India rakri@microsoft.com
Harsha Vardhan Simhadri Microsoft Research India harshasi@microsoft.com

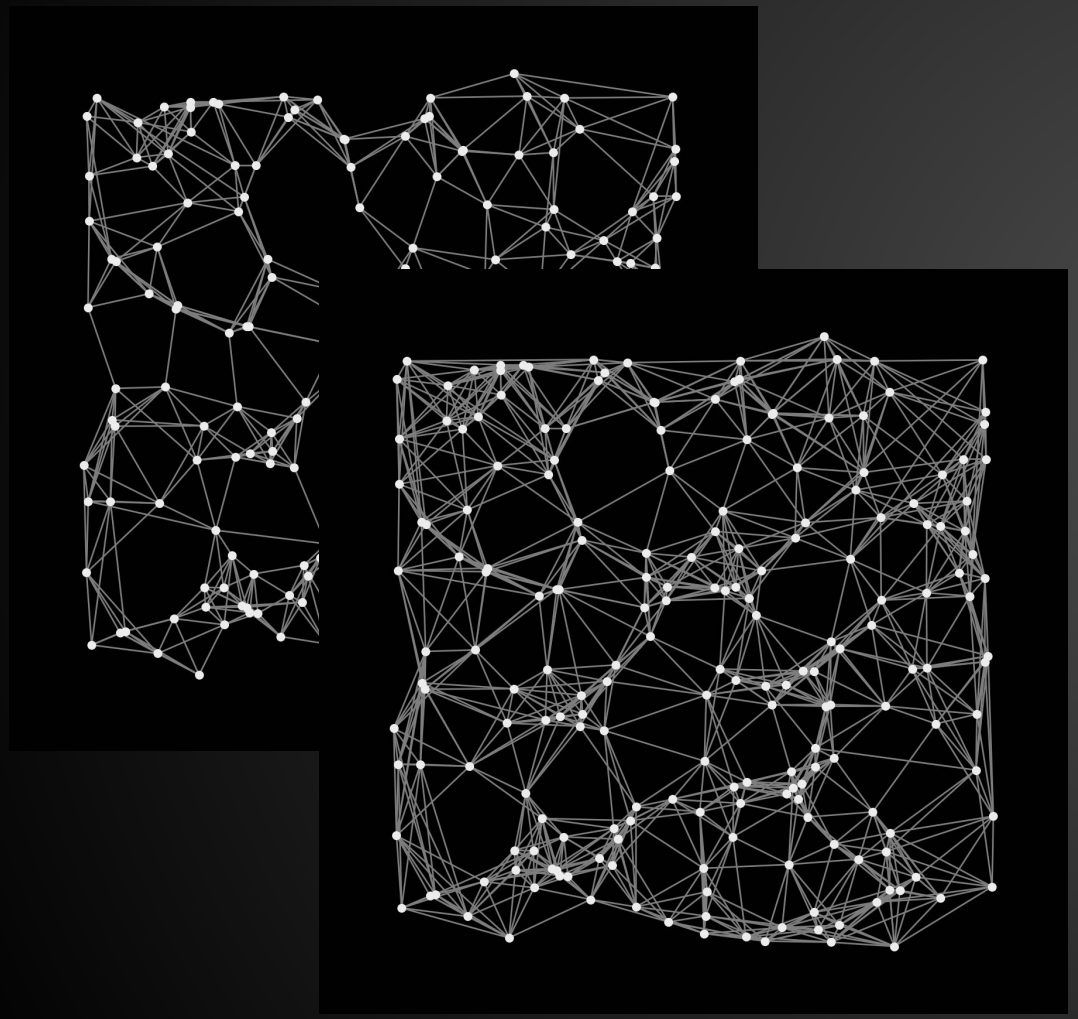
Abstract

Current state-of-the-art approximate nearest neighbor search (ANNS) algorithms generate indices that must be stored in main memory for fast high-recall search. This makes them expensive and limits the size of the dataset. We present a new graph-based indexing and search system called DiskANN that can index, store, and search a billion point database on a single workstation with just 64GB RAM and an inexpensive solid-state drive (SSD). Contrary to current wisdom, we demonstrate that the SSD-based indices built by DiskANN can meet all three desiderata for large-scale ANNS: high-recall, low query latency and high density (points indexed per node). On the billion point SIFT1B *bigann* dataset, DiskANN serves > 5000 queries a second with < 3ms mean latency and 95%+ 1-recall@1 on a 16 core machine, where state-of-the-art billion-point ANNS algorithms with similar memory footprint like FAISS [18] and IVFOADC+G+P [8] plateau at around 50% 1-recall@1. Alternately, in the high recall regime, DiskANN can

jVector: DiskANN

= Vamana + PQ + oversampling

Vamana

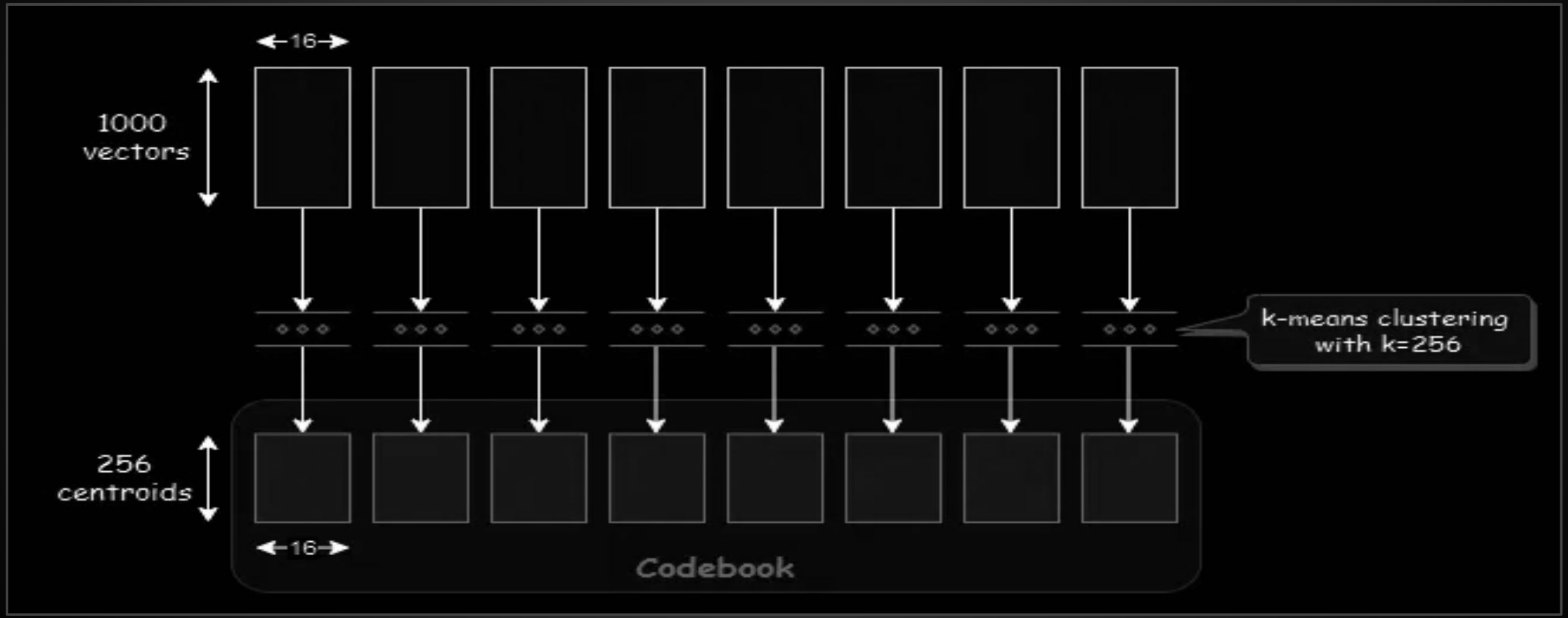


- Built with NNDecent
- Data Compressed in memory
- Full vector on disk, less lookup
- Way better in high recall (no cap)

$$\text{recall} = \frac{\text{Number of docs matchings retrieved}}{\text{Number of docs matchings available}}$$

Product quantization

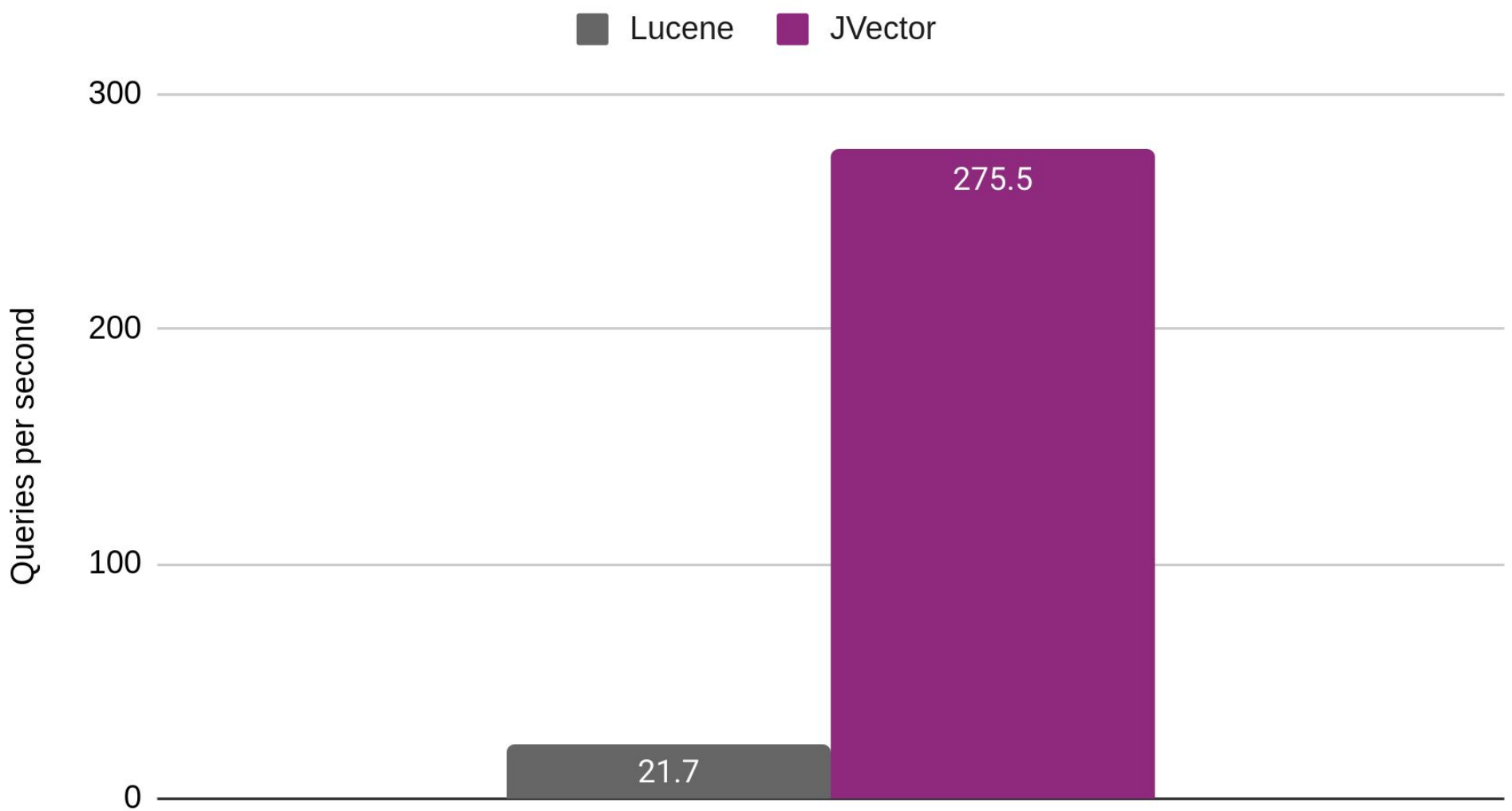
Lossy Compression for vectors



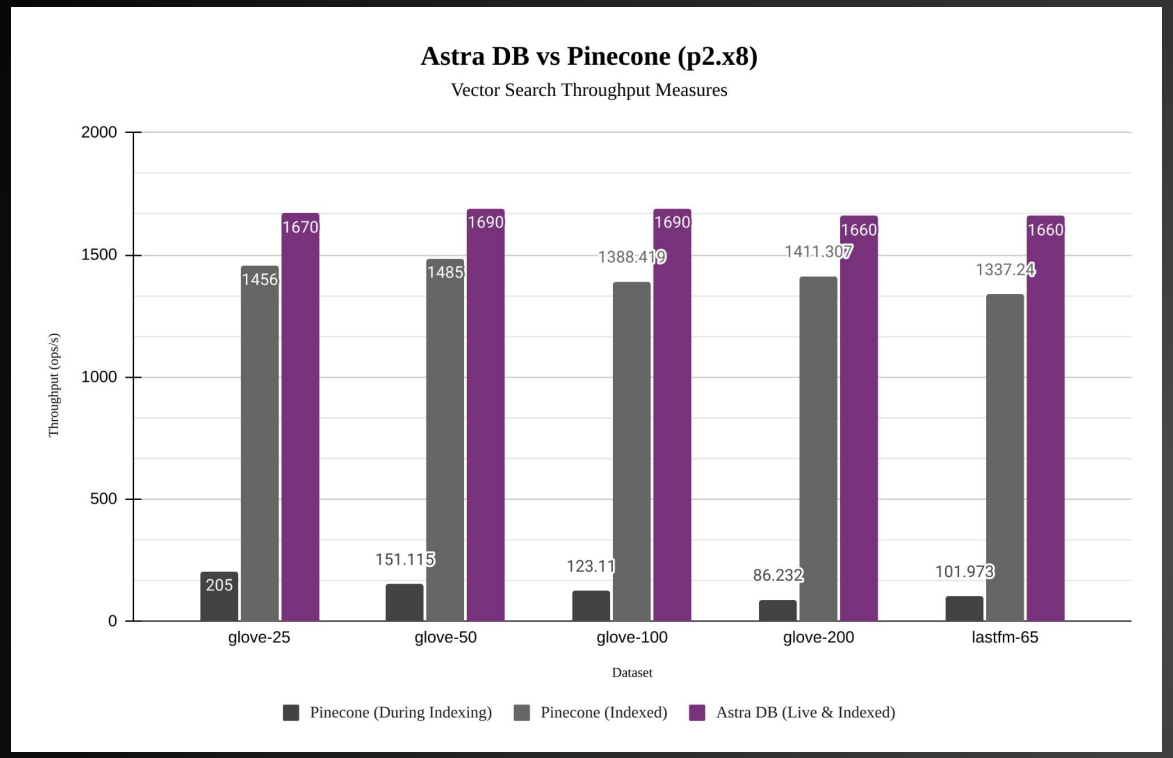
Oversampling

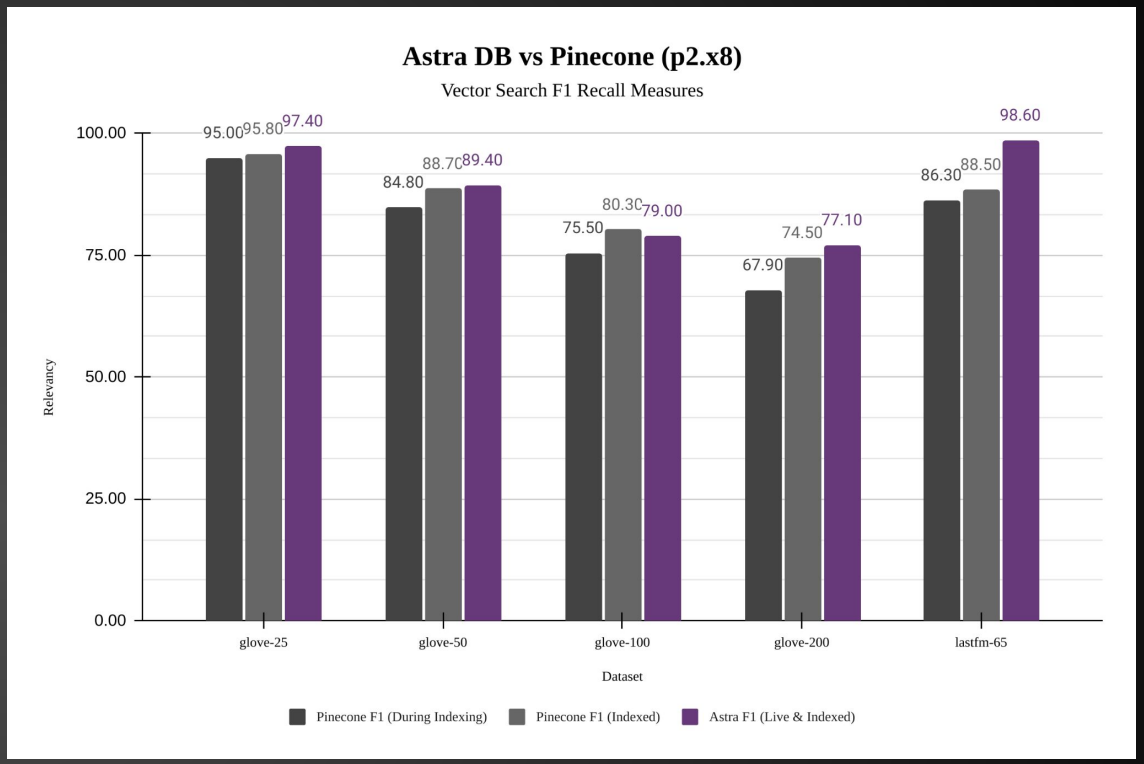
- **Instead of searching for closest K, search for closest 2K (using compressed comparisons)**
- **Read uncompressed vectors from disk during search whenever a candidate is added to the resultset**
- **Reorder the resultset (of 2K) using uncompressed vectors, and return the top K**

QPS on Deep100M dataset (24GB Macbook)



Concurrency World





$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$F1 = 2 \times \left(\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \right)$$

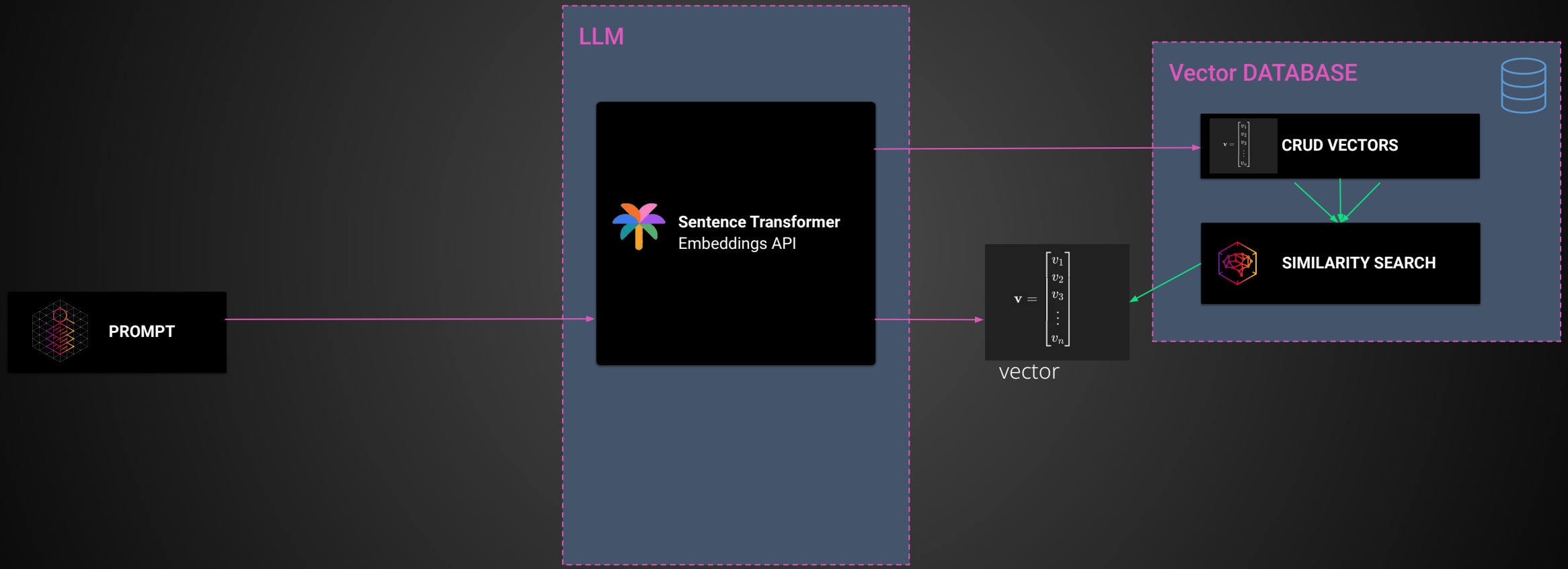
jVector: 5 Hard Problems We're Solving

- **Scale-Out Capabilities**: No upper limits
- **Garbage Collection**: Pruning obsolete index information
- **Effective Use of Disk:** Enabling high throughput
- **Composability**: Predicates, term-based searches. Aka Hybrid Search
- **Concurrency**: Non-blocking, multi-threaded index construction

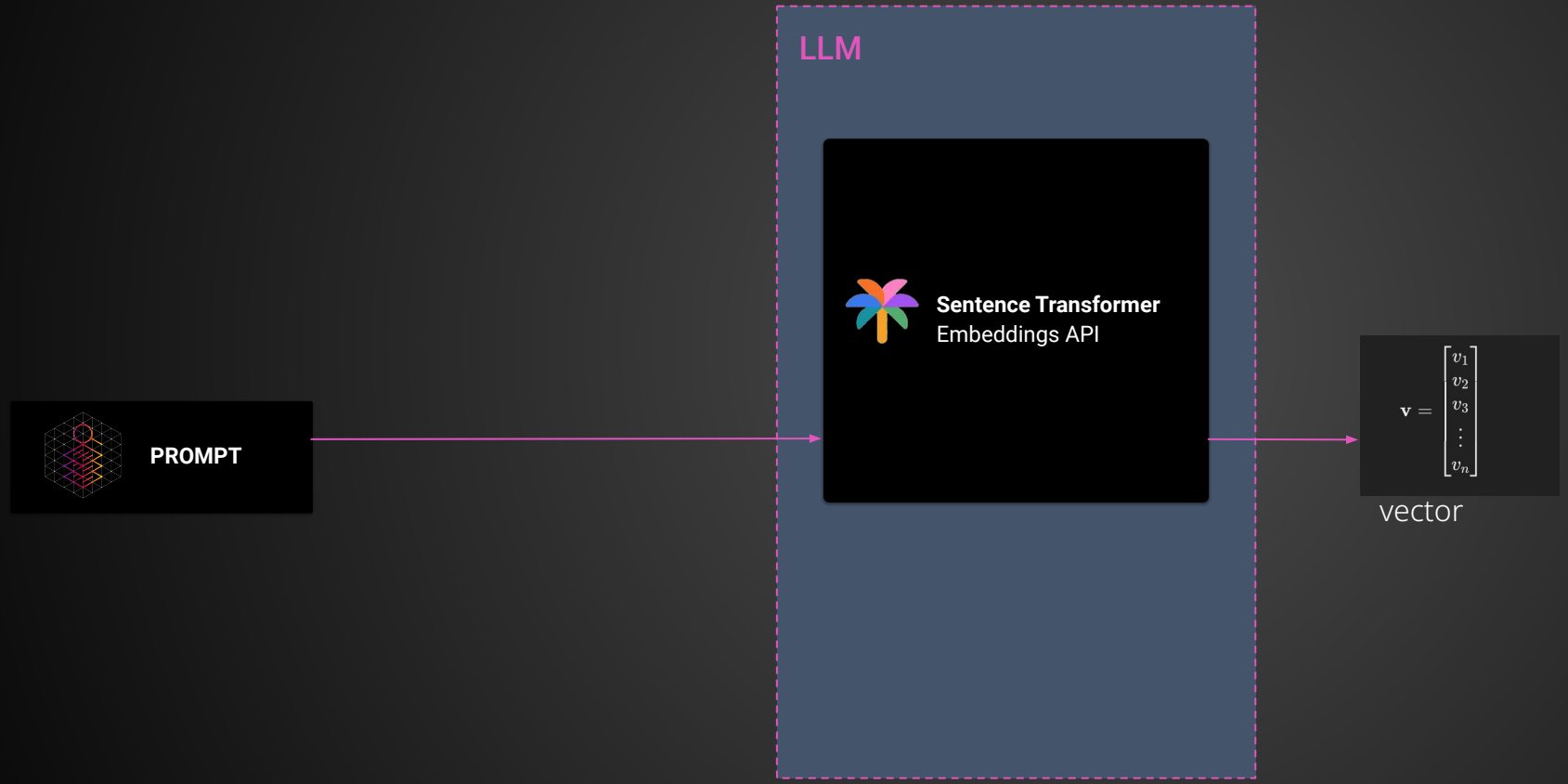
https://thenewstack.io/5-hard-problems-in-vector-search-and-how-cassandra-solves-them/

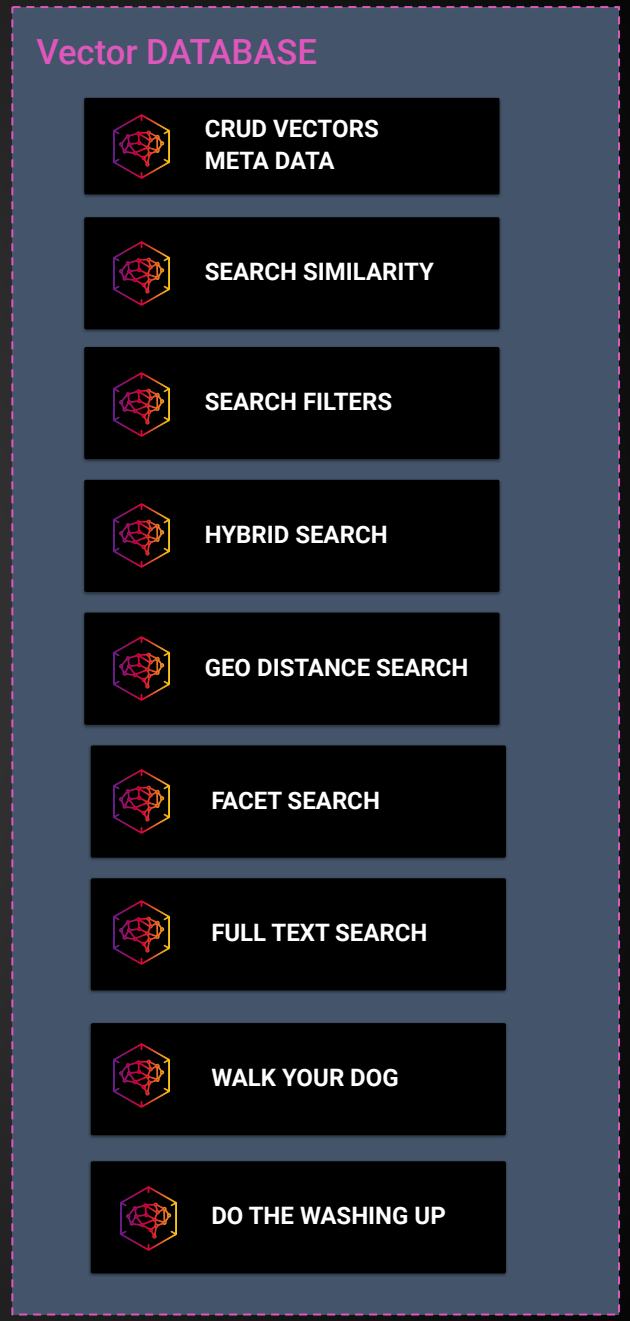
https://github.com/jbellis/jvector

Vector *Database*



Vector *Database*





Vector Database Features

https://superlinked.com/vector-db-comparison

Vector DB Comparison

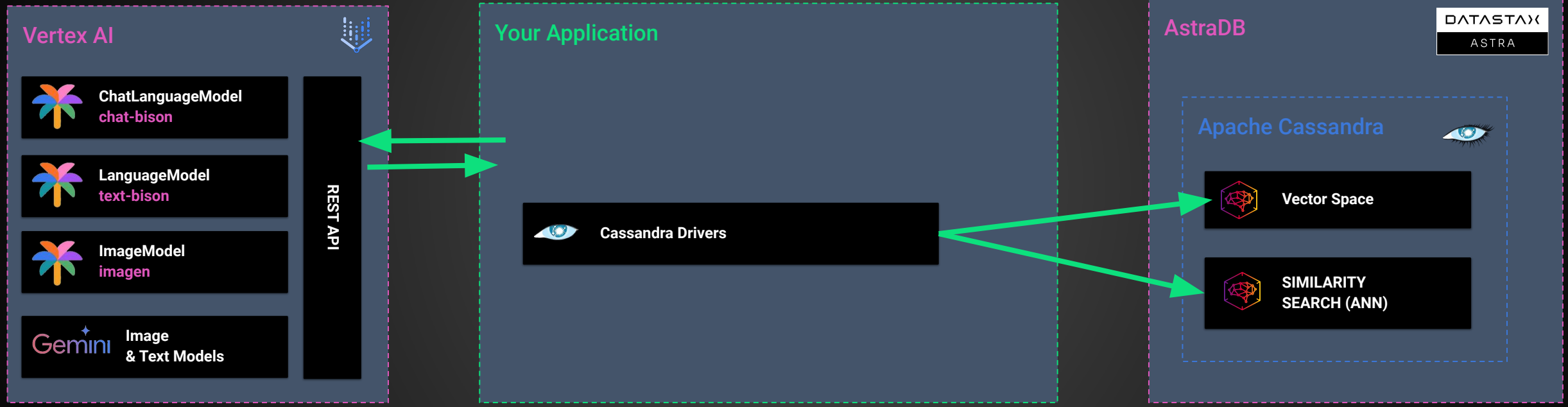
by Superlinked | Last Updated : Today

Search | Get insights | Give us a star

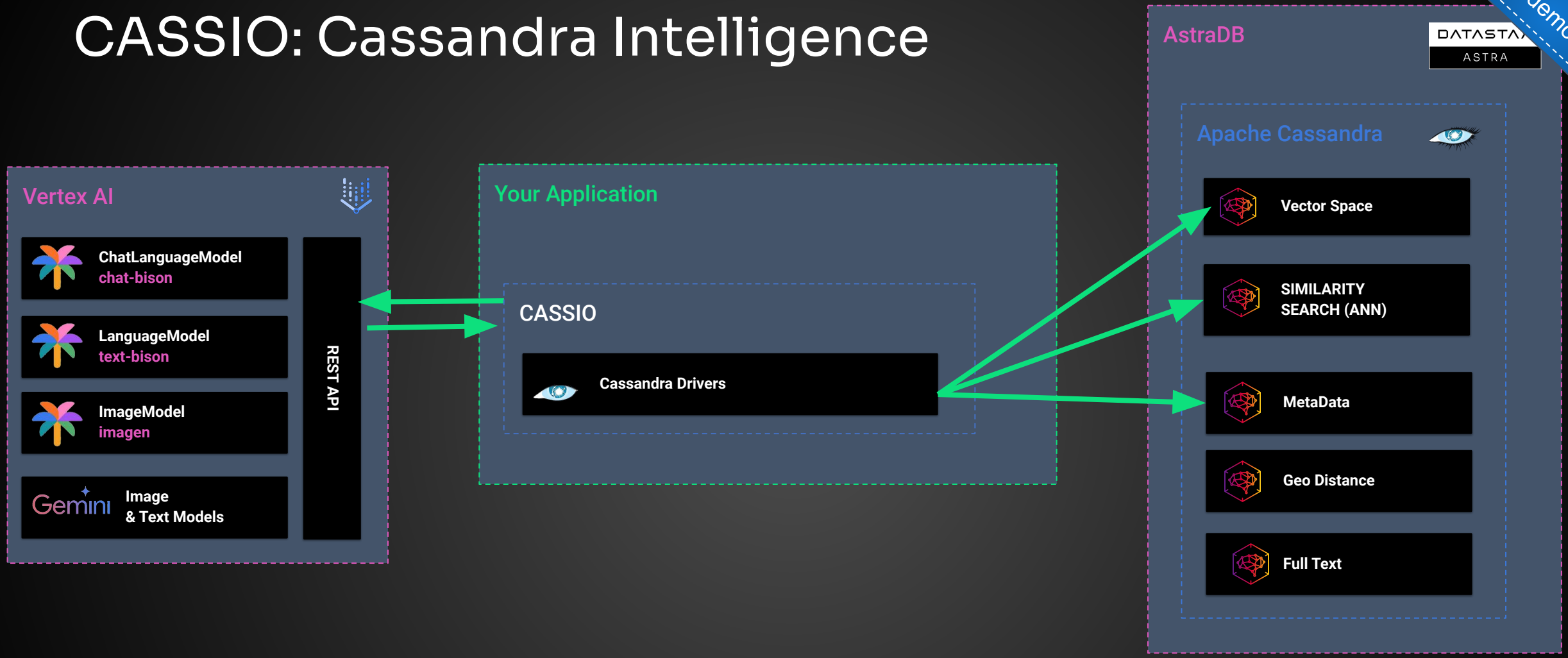
Vendor	About				Search							
	OSS	License	Dev Lang	VSS Launch	Filters	Hybrid Search	Facets	Geo Search	Multi-Vector	Sparse	BM25	Full-Text
Activeloop Deep Lake	✓	MPL 2.0	python c++	2023	✓	-	-	✗	✓	■	✗	■
Anari AI	✗	Proprietary	-	2023	-	✗	-	✗	-	-	✗	-
Apache Cassandra	✓	Apache-2.0	java	2023	✓	✓	-	-	-	✗	-	✗
Apache Solr	✓	Apache-2.0	java	2022	✓	✓	✓	✓	✓	-	✓	✓
ApertureDB	-	-	-	-	-	-	-	-	-	-	-	-
Azure AI Search	✗	Proprietary	c++	2023	✓	✓	✓	✓	✓	✗	✓	✓
Chroma	✓	Apache-2.0	python	2022	✓	✗	-	-	✗	✗	✗	✗
ClickHouse	✓	Apache 2.0	c++	2022	✓	✗	✓	✓	✓	✗	✗	✓
CrateDB	✓	Apache 2.0	java	2023	✓	-	✓	✓	✓	-	✓	✓
DataStax Astra DB	✓	Proprietary	java go	2023	✓	✓	✓	✓	✗	✗	✗	✓
Elasticsearch	✗	Elastic Lice...	java	2021	✓	✓	✓	✓	✓	✓	✓	✓

Cassandra Vector Store

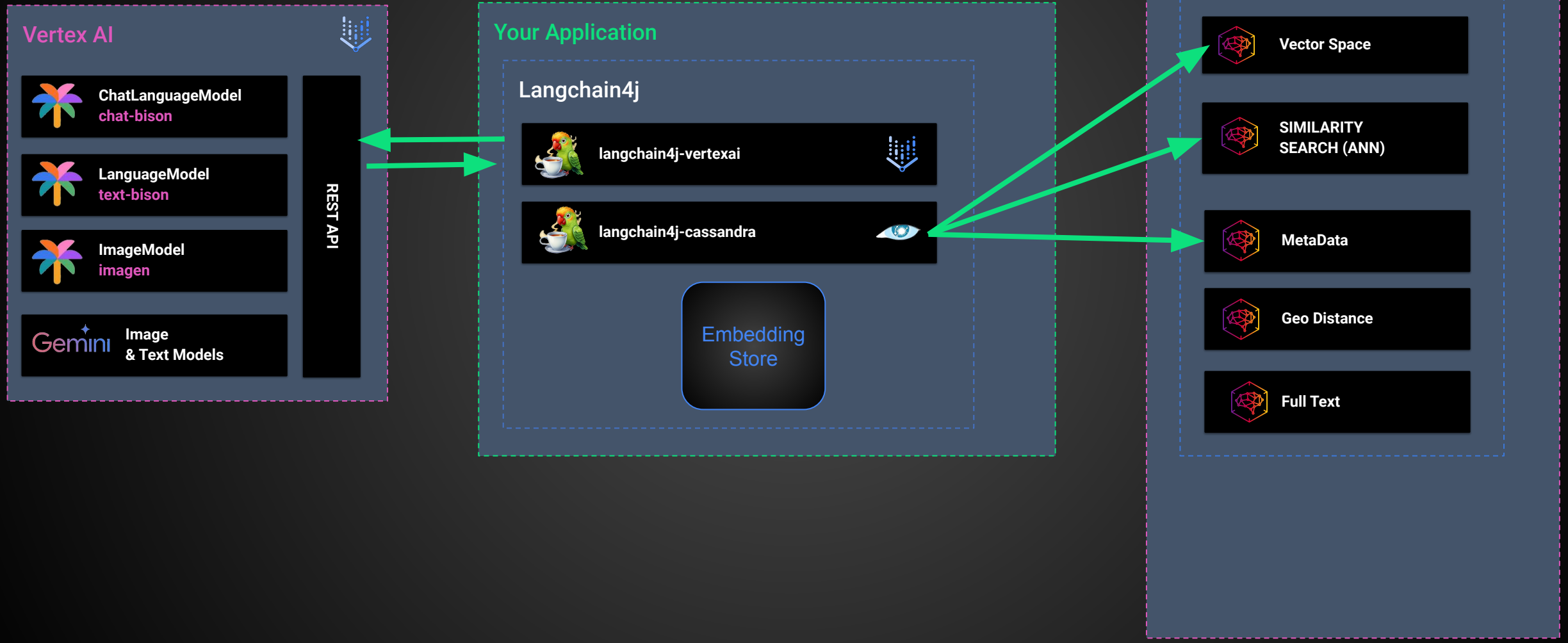
demo



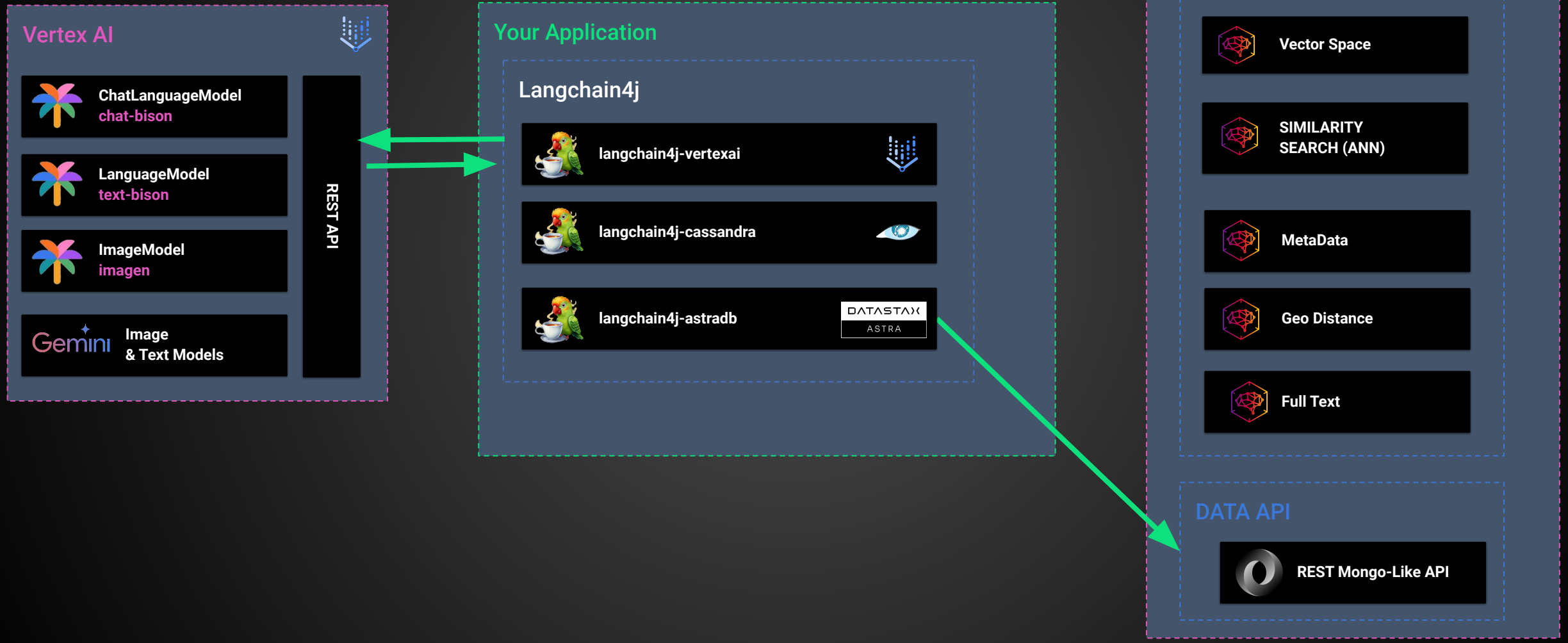
CASSIO: Cassandra Intelligence



Cassandra Vector Store



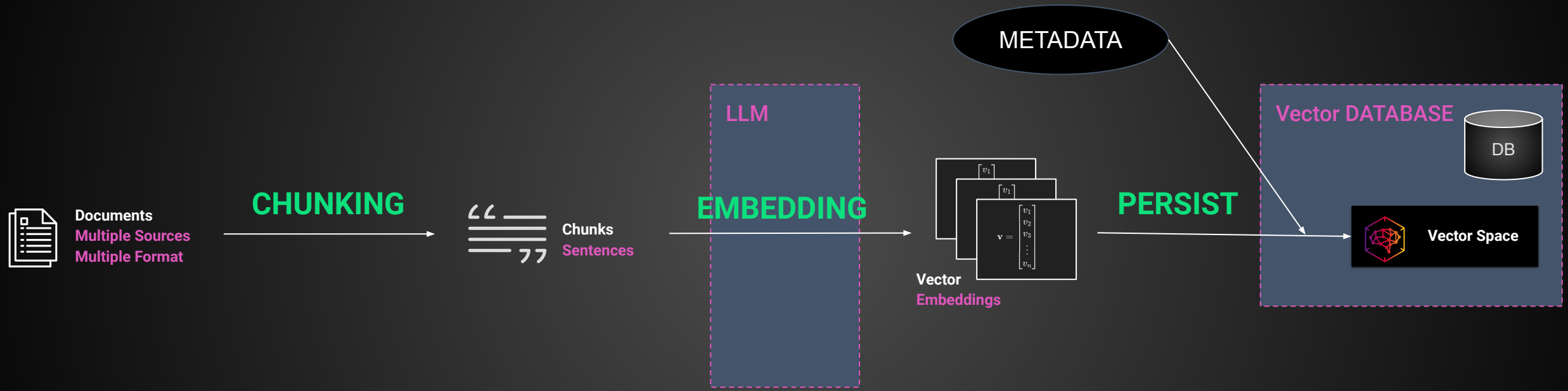
AstraDB Vector Store



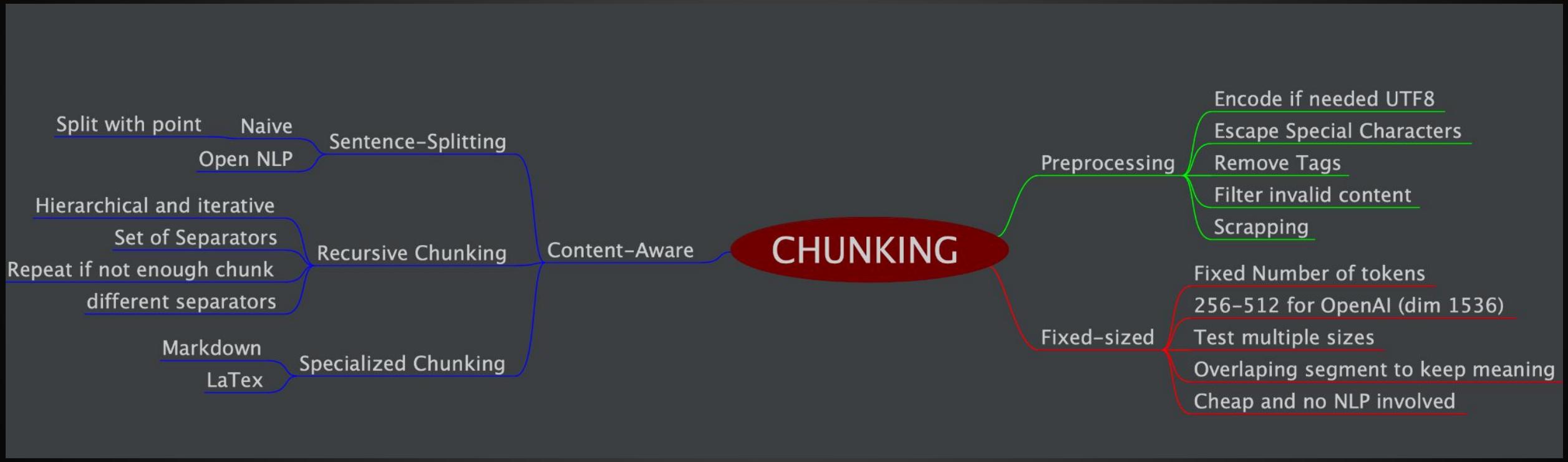
6.
Retrieval Augmented Generation

- Naive Rag
  - Ingestion
  - Retrieval
- Advanced RAG
  - Overview
  -
  -

(Naive) Retrieval-Augmented Generation
Ingestion

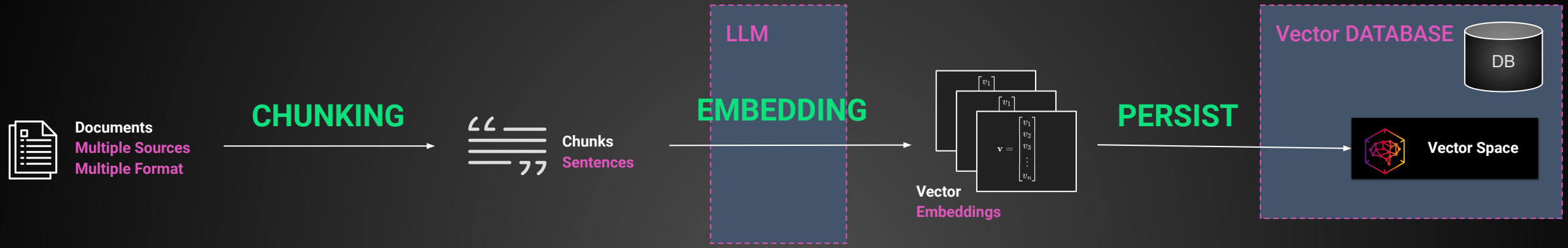


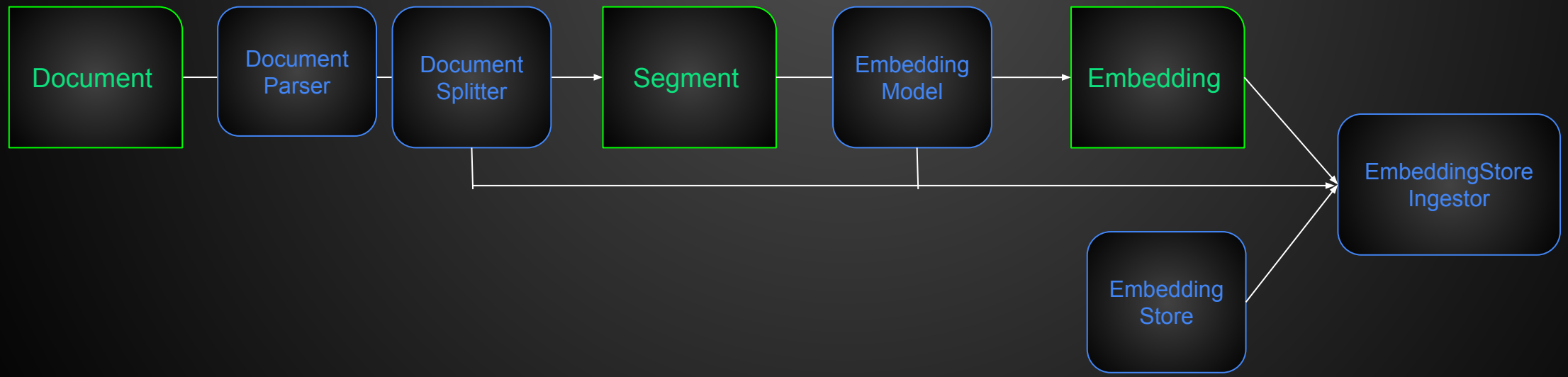
Chunking


CHUNKING
Content-Aware
Sentence-Splitting
Naive
Split with point
Open NLP
Recursive Chunking
Hierarchical and iterative
Set of Separators
Repeat if not enough chunk
different separators
Specialized Chunking
Markdown
LaTex
Preprocessing
Encode if needed UTF8
Escape Special Characters
Remove Tags
Filter invalid content
Scrapping
Fixed-sized
Fixed Number of tokens
256-512 for OpenAI (dim 1536)
Test multiple sizes
Overlaping segment to keep meaning
Cheap and no NLP involved

(Naive) Retrieval-Augmented Generation
Ingestion

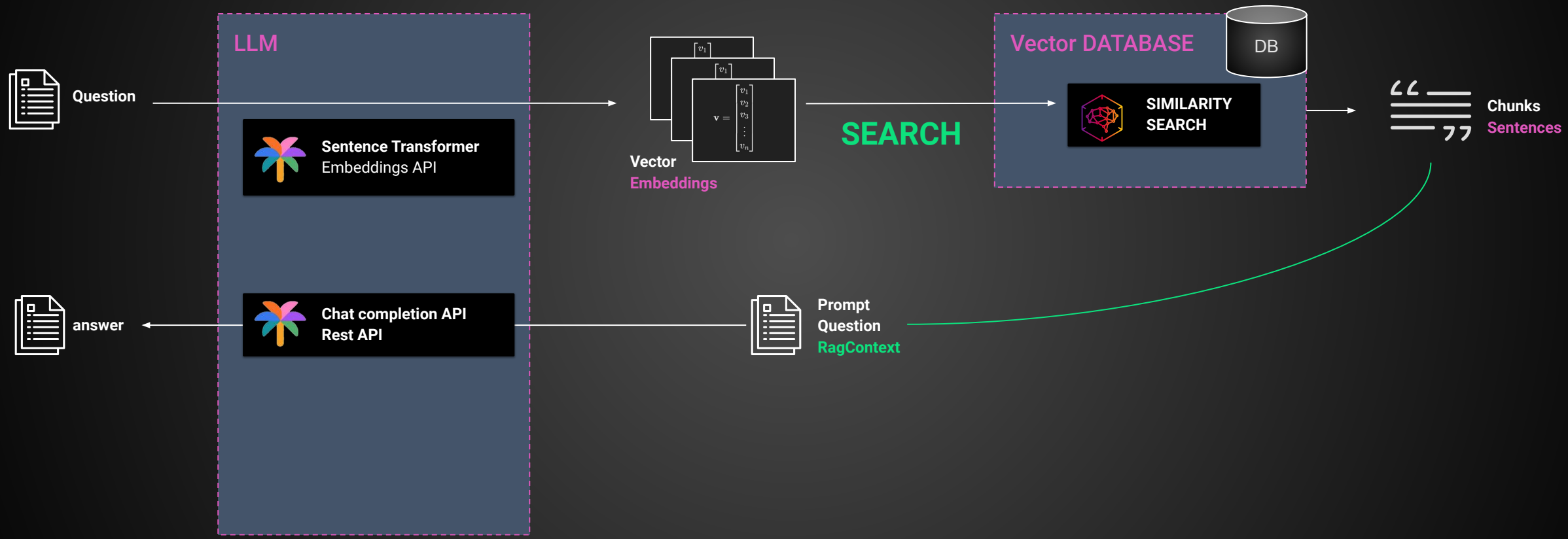
demo



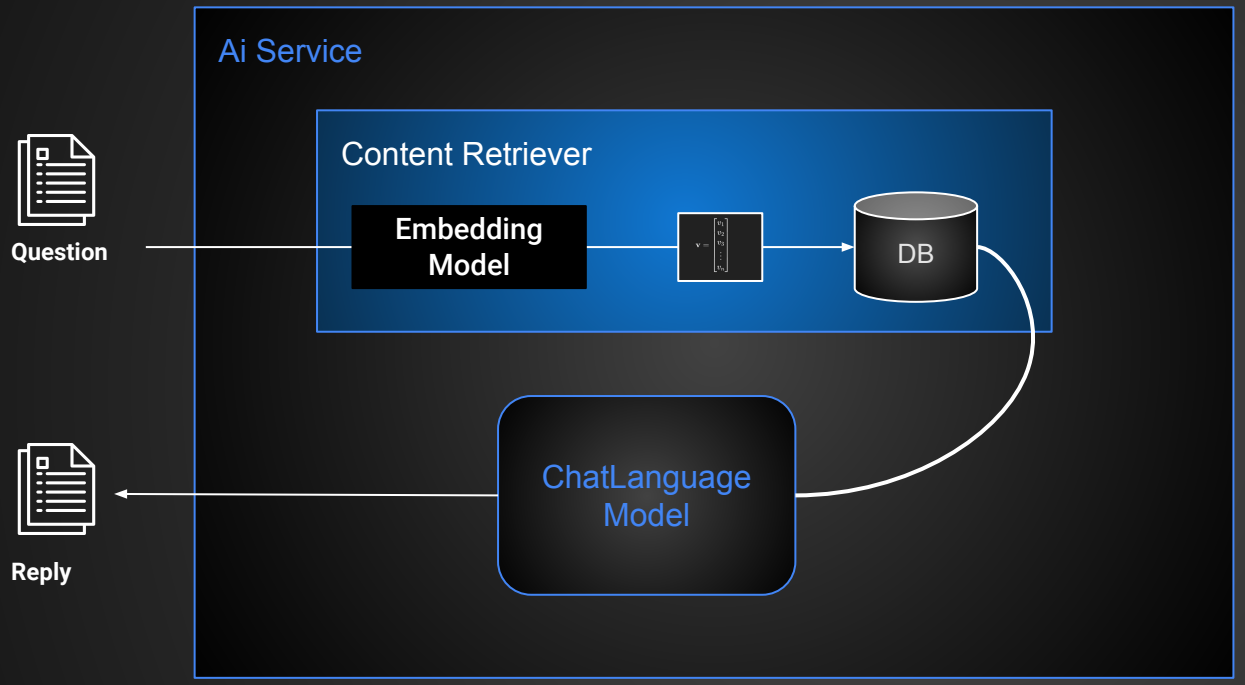


(Naive) Retrieval-Augmented Generation
Retriever

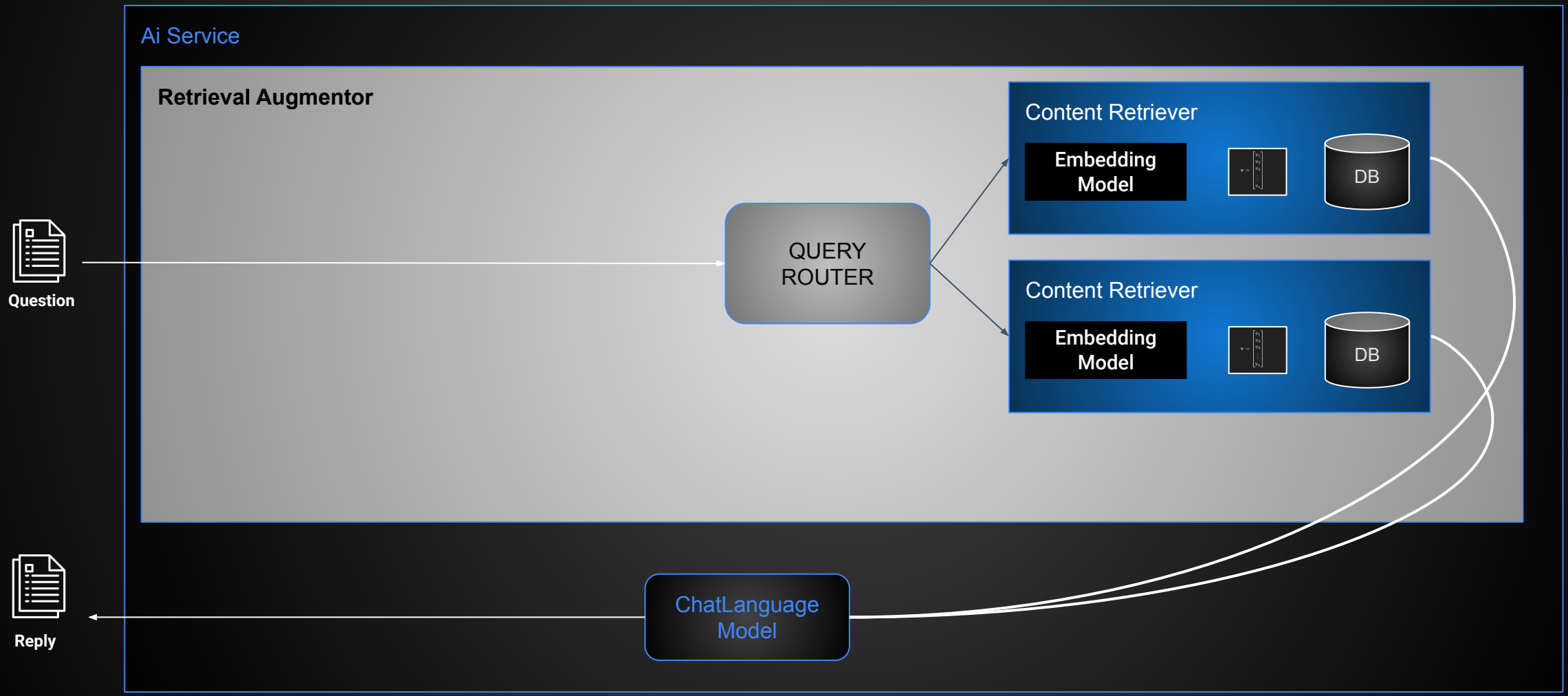
demo



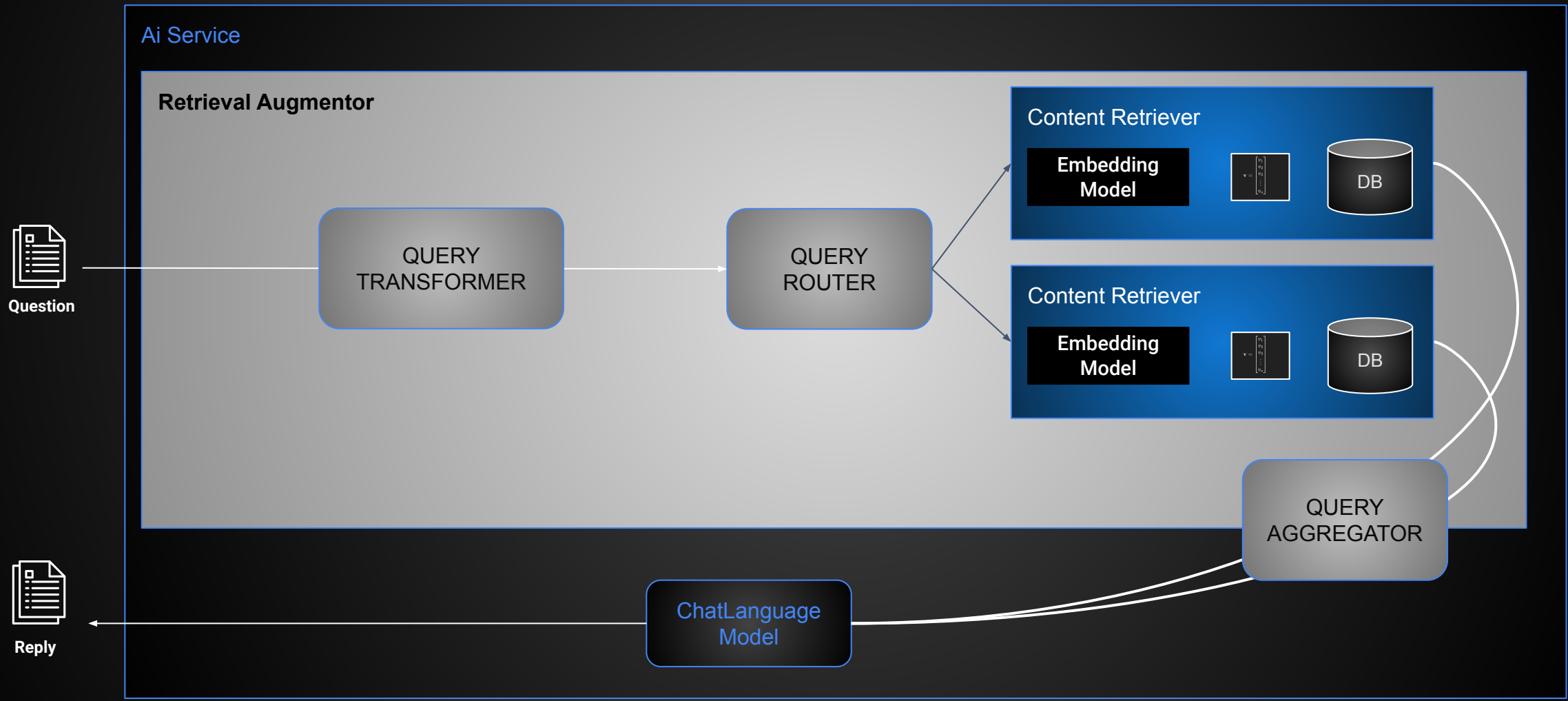
(Naive) Retrieval-Augmented Generation
Retriever



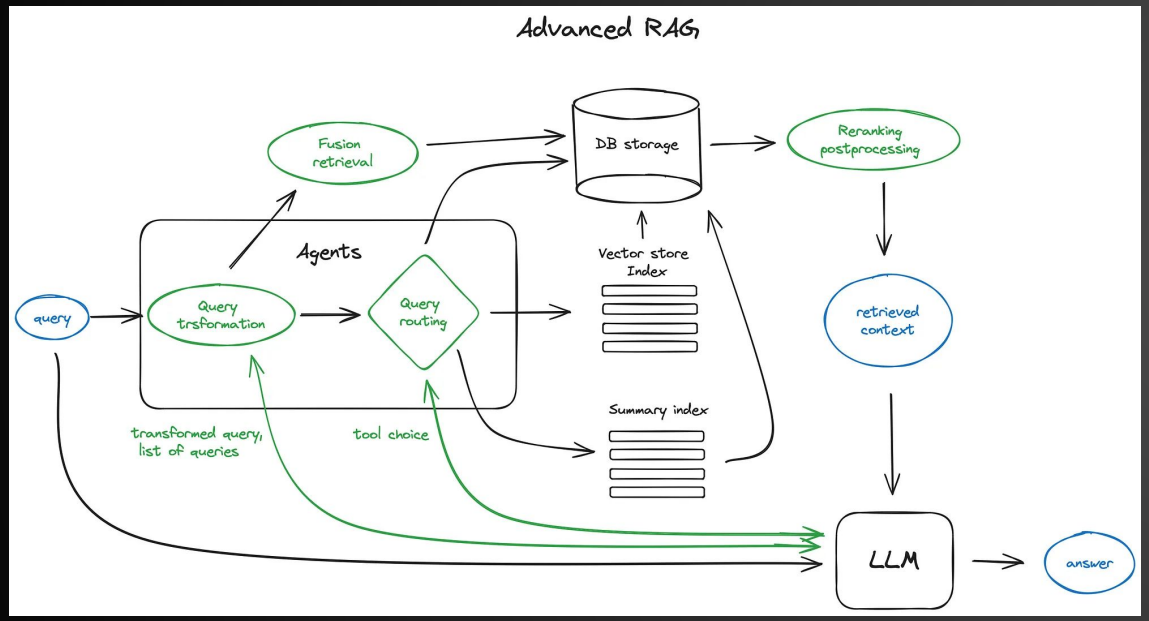
Advanced Retrieval-Augmented Generation
Query Routing



Advanced Retrieval-Augmented Generation
Re-Ranking



Advanced RAG



Retriever — retrieves passages of text from a knowledge source that are relevant to the context

Reranker (optional) — rescores and reranks retrieved passages

Generator — integrates context with retrieved passages to generate output text

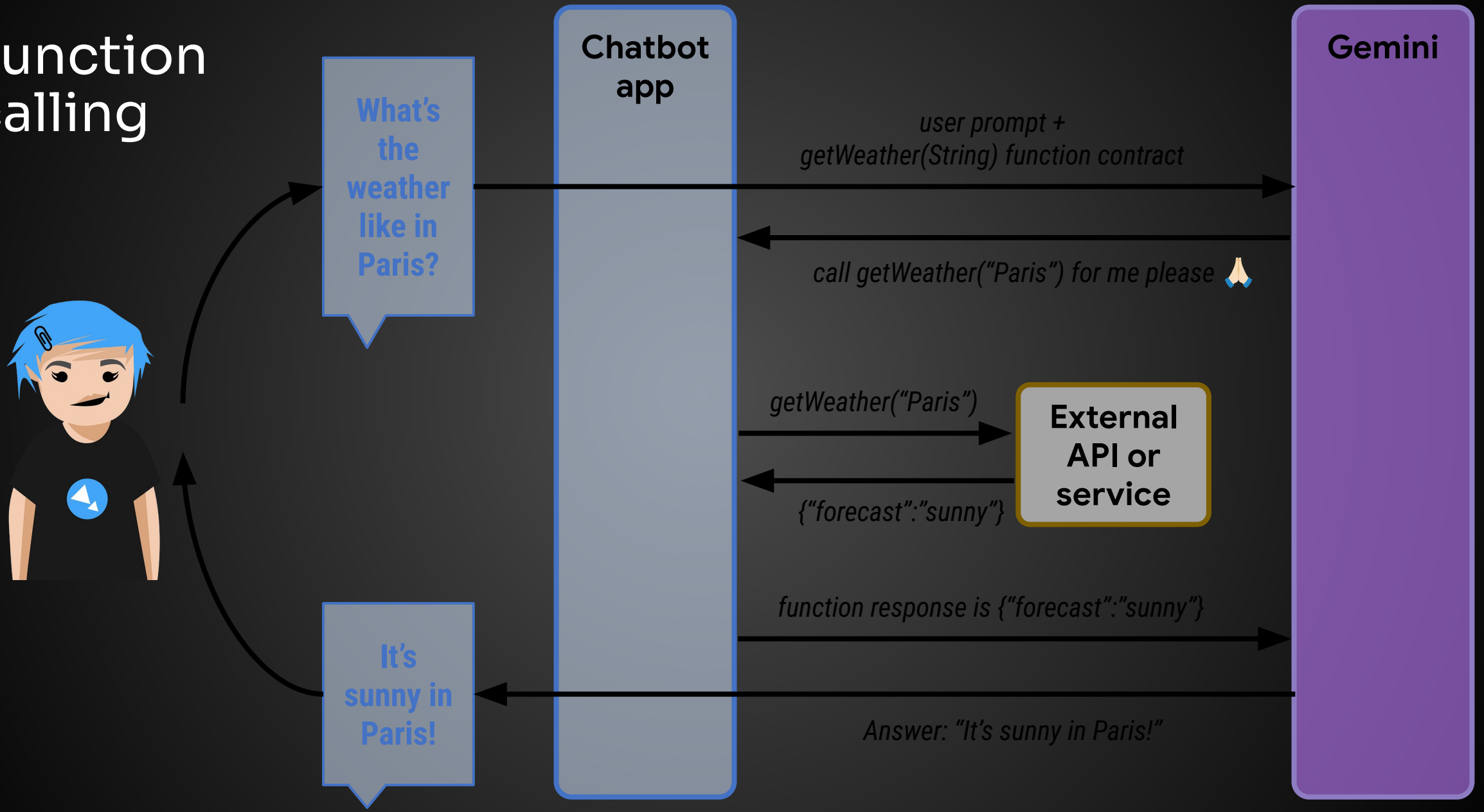
More Techniques Advanced RAG Techniques

- Embeddings
  - Hierarchical index retrieval (index summary then chunk)
  - colBERT (full retrieval)
- Retriever
  - Parent Retriever : Find the chunk, include the paragraph
  - Hybrid Differential Evolution (HyDe)
- LOTR
- Scoring / Re-Ranking
  - Rag Fusion
  - Forward Looking active Retrieval (F.L.A.R.E)
  - Reciprocal Rank Fusion
  - colBERT (vector per token)
  - Cohere
- Working with dense Data
  - Maximal Margin Relevance (M.M.R)
  - colBERT

8.

Functions Calling
Semantic Search

Function calling



Conclusion and more Generative AI & RAG @Devoxx

Apache Lucene : de l'indexation textuelle à l'intelligence artificielle

TOOLS-IN-ACTION (INTERMEDIATE LEVEL)

Friday from 17:00 – 17:30

Paris 242AB

Comprendre l'IA: construisez votre propre ChatGPT d'entreprise avec LangChain4J

2H HANDS-ON LAB (BEGINNER LEVEL)

Thursday from 10:30 – 12:30

Paris 243

Construire son Assistant Intelligent avec Hugging Face et Elasticsearch

DEEP DIVE (INTERMEDIATE LEVEL)

Thursday from 13:30 – 16:30

Neuilly 251

Generative AI in Practice: A Hands-on Codelab

3H HANDS-ON LAB (BEGINNER LEVEL)

Thursday from 13:30 – 16:30

Paris 142

Java rencontre l'IA : Comment intégrer les LLMs dans vos applications avec LangChain4j

CONFERENCE (INTERMEDIATE LEVEL)

Thursday from 13:30 – 14:15

Amphi bleu

La recherche à l'ère de l'IA

CONFERENCE (INTERMEDIATE LEVEL)

Thursday from 11:35 – 12:20

Paris 141

Comment ça marche l'IA Generative ? LLM, RAG sous le capot.

CONFERENCE (BEGINNER LEVEL)

Friday from 11:35 – 12:20

Paris 141

LangChain4j en Action - Créez des Applications avec LLMs

2H HANDS-ON LAB (BEGINNER LEVEL)

Friday from 10:30 – 12:30

Neuilly 253

RAGtime : Discuter avec vos propres données

3H HANDS-ON LAB (INTERMEDIATE LEVEL)

Friday from 13:30 – 16:30

Paris 243


THANKS
FOR WATCHING
DEVOXX FRANCE 2024