

# Build AI-powered Search Applications with Milvus Vector Database

Jiang Chen @ Zilliz





# Jiang Chen

Head of Ecosystem and  
Developer Relations



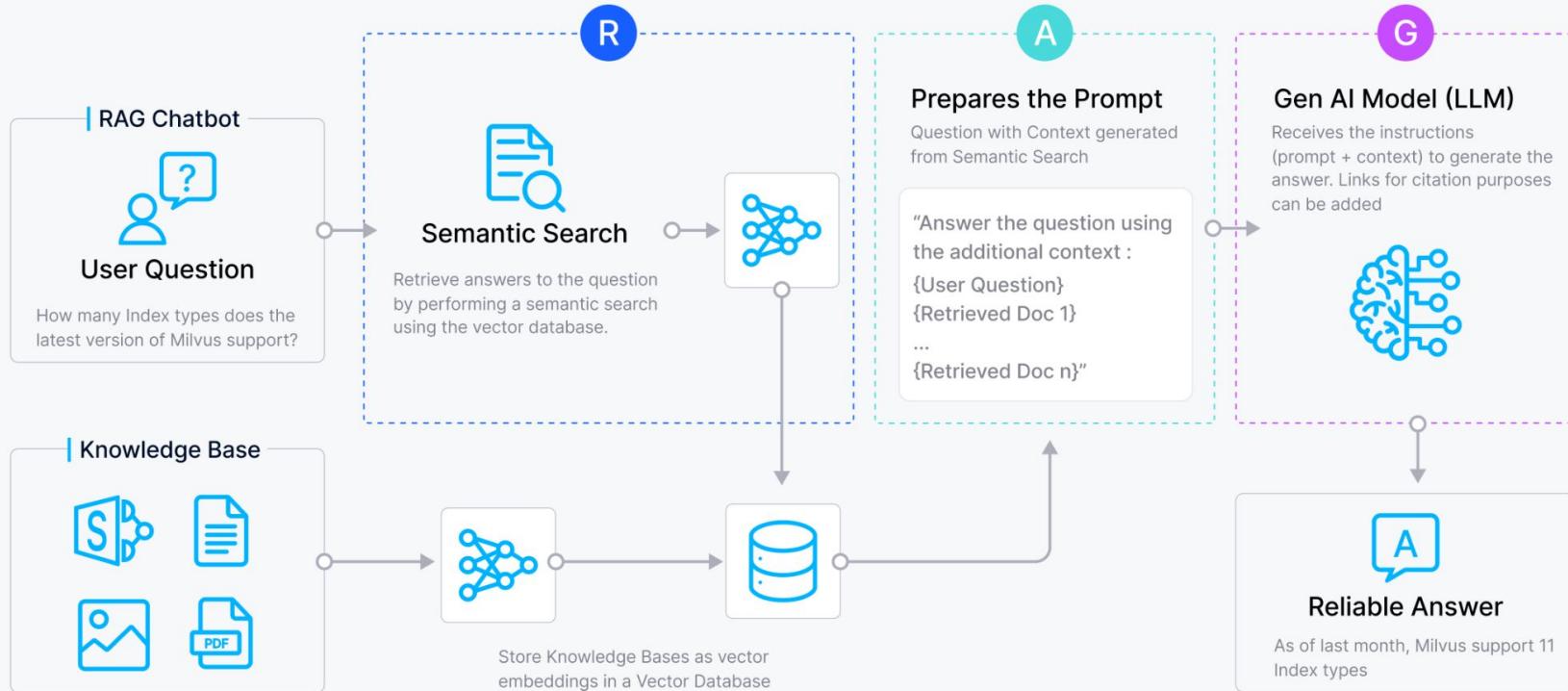
[jiang.chen@zilliz.com](mailto:jiang.chen@zilliz.com)

<https://www.linkedin.com/in/jiangc1010/>

<https://x.com/jiangc1010>

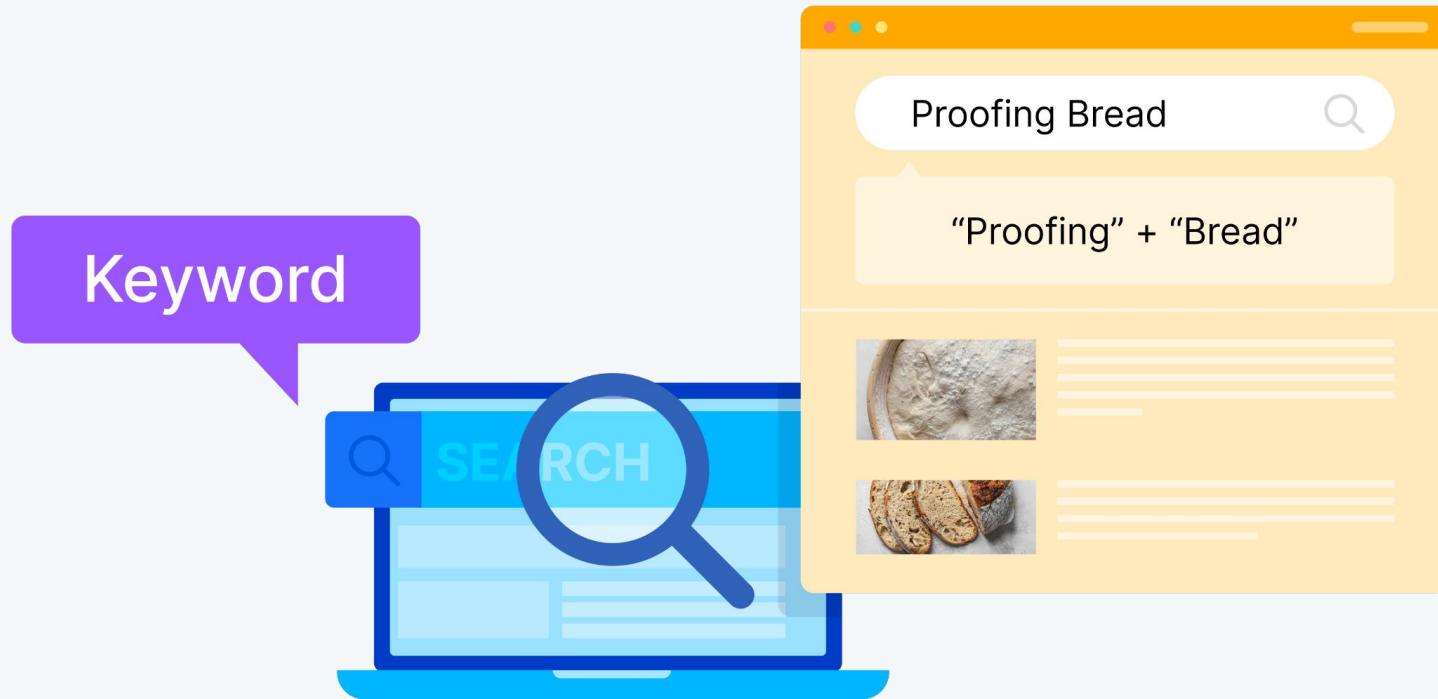


# Retrieval-Augmented Generation



# Shifted Search and Data Paradigm

# Traditional database was built upon exact search



# ...which misses context, semantic meaning, and user intent

Q | Apple



VS.



Q | Rising dough

Rising Dough ✓

VS.

Proofing Bread ✗

Q | Change car tire

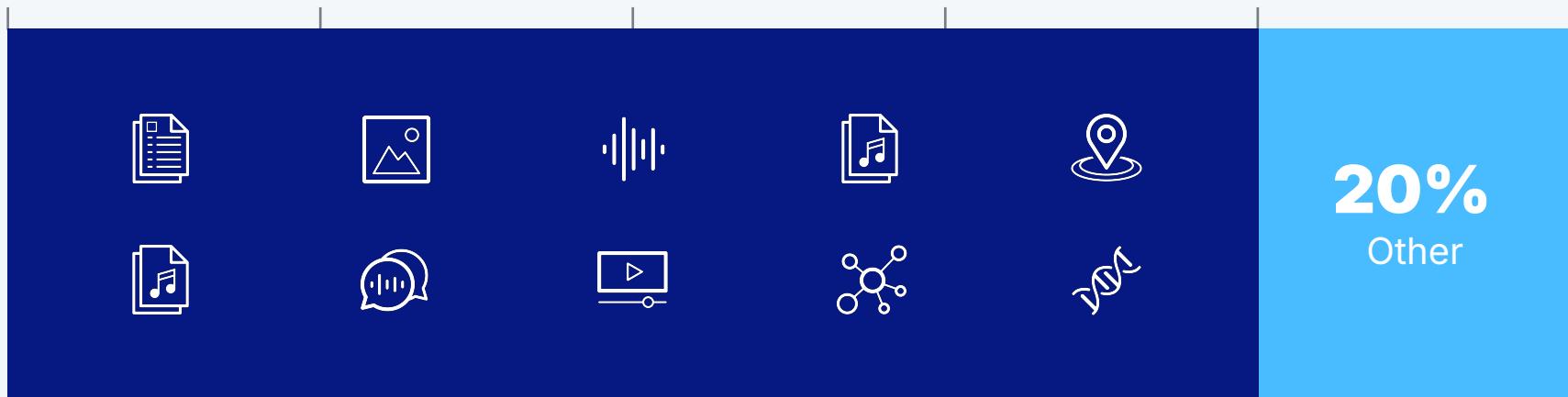


VS.



...and cannot process increasingly growing unstructured data

**< 80%** newly generated data in 2025  
will be unstructured data



# The evolution of AI made the semantic search of unstructured data possible



## Search by Probability

Statistical analyses of common datasets established the foundation for processing unstructured data, e.g. NLP, and image classification



## AI Model Breakthrough

The advancements in BERT, ViT, CBT etc. have revolutionized semantic analysis across unstructured data



## Vectorization

Word2Vec, CNNs, Deep Speech pioneered unstructured data embeddings, mapping the words, images, videos into high-dimensional vectors

# This new AI breakthrough requires new databases to fully unleash its potential



## Support multiple use case types

Accommodate diverse data requirements, enhancing flexibility and effectiveness in varied operational contexts



## Scale as needed

Enable robust handling of expanding data volumes and search demands



## Highly performant

Ensures swift and accurate query responses, crucial for optimal user experience



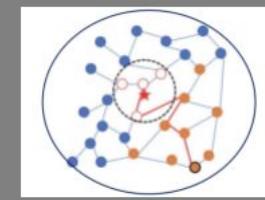
Unstructured Data



Embedding Model

$$\begin{bmatrix} \mathbf{V}_{1,1} & \mathbf{V}_{1,2} & \cdots & \mathbf{V}_{1,n} \\ \mathbf{V}_{2,1} & \mathbf{V}_{2,2} & \cdots & \mathbf{V}_{2,n} \\ \mathbf{V}_{3,1} & \mathbf{V}_{3,2} & \cdots & \mathbf{V}_{3,n} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{V}_{n,1} & \mathbf{V}_{n,2} & \cdots & \mathbf{V}_{n,n} \end{bmatrix}$$

Vectors



Similarity Search

# Milvus: The most widely-adopted vector database

Milvus is an **Open-Source Vector Database** to **store, index, manage, and use** the massive number of **embedding vectors** generated by deep neural networks and LLMs.



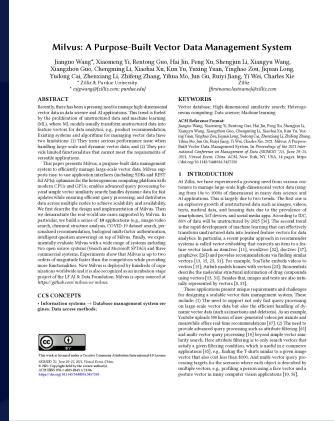
400+  
contributors

29K+  
stars

66M+  
docker pulls

2.7K  
+  
forks

# Built by database & AI experts



Milvus was built by a top-tier team of **algorithm and database engineers** with a strong pedigree in developing **high-performance, scalable, and highly available** distributed systems, uniquely tailored for **vector search**.

# Milvus Users



AT&T



BOSCH

Chegg



CISION®

COMPASS

Deloitte.

ebay

FARFETCH

Grab



Inflection

intuit

Microsoft

new relic

NVIDIA®

OMERS

OII Otter.ai

PayPal

paloalto  
NETWORKS

POSHMARK

RABLOX

salesforce

Shell

shutterstock

T

TREND  
MICRO

Walmart

ZipRecruiter

zomato

# Multi-modal Search

Image

Drag and drop file here  
Limit 200MB per file

Browse files



Text

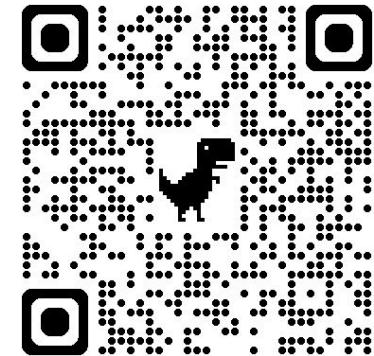
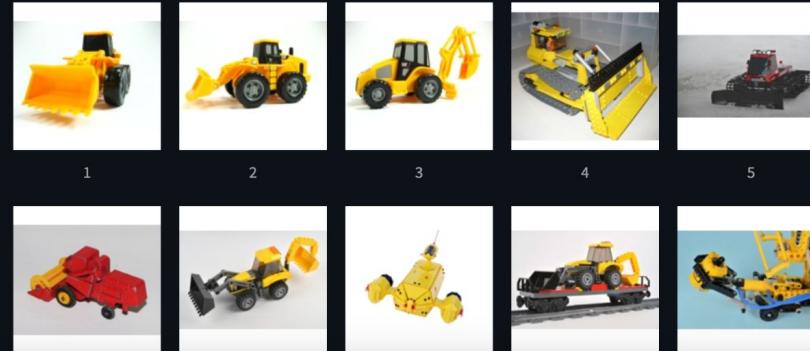
toy of this.

## Multimodal Image Search

Powered by  milvus

To learn more, check out our [tutorial here!](#)

### Search Results



[multimodal-demo.milvus.io](https://multimodal-demo.milvus.io)

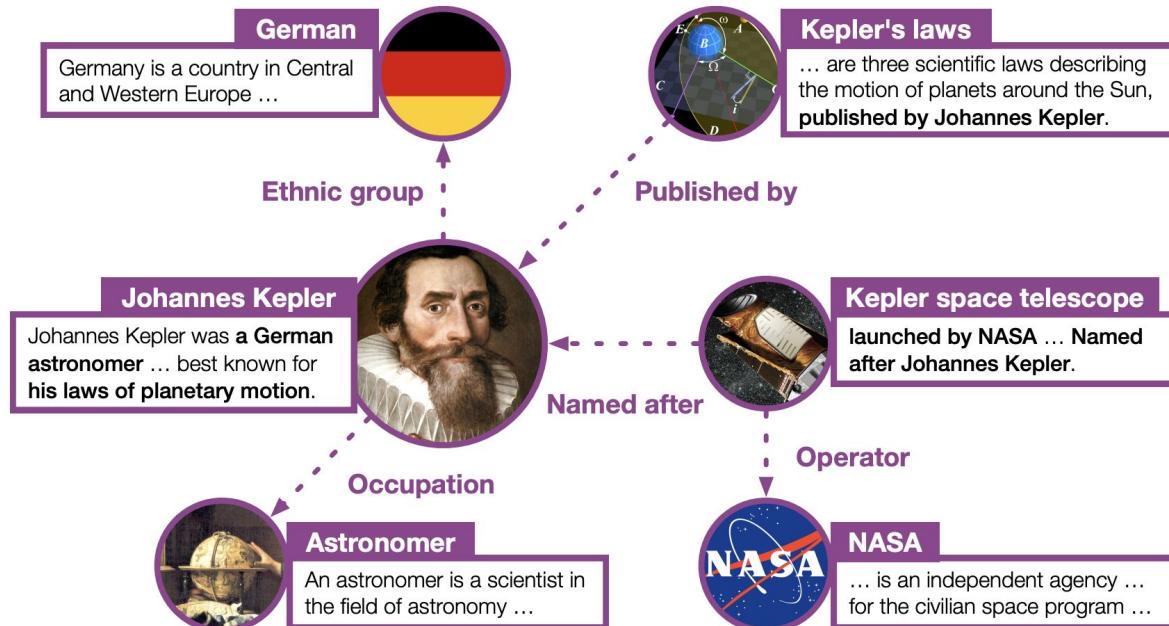
# MagicLens

Query Image	Instruction	MagicLens	Prior SOTA
	<i>Find the identical image</i>		
	<i>Compare its height to the world's tallest building</i>		
	<i>Outside view from the inside of it</i>		
	<i>Find other attractions in this country</i>		

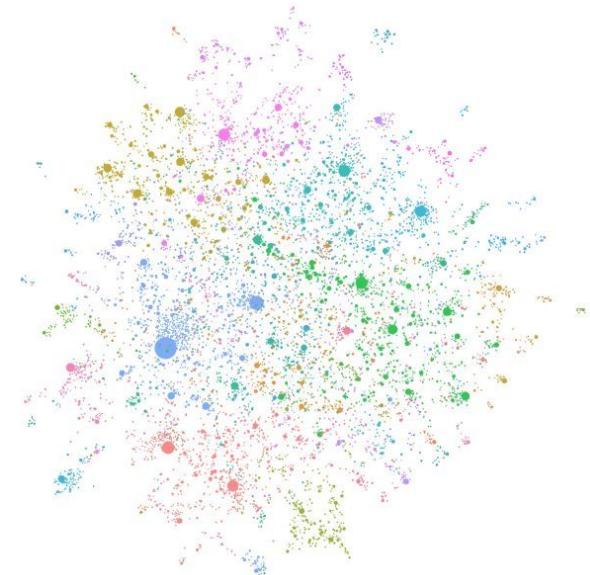
(query image, target image) -> PaLI -> PaLM2 -> instruction



# Knowledge Graph and Data Mining

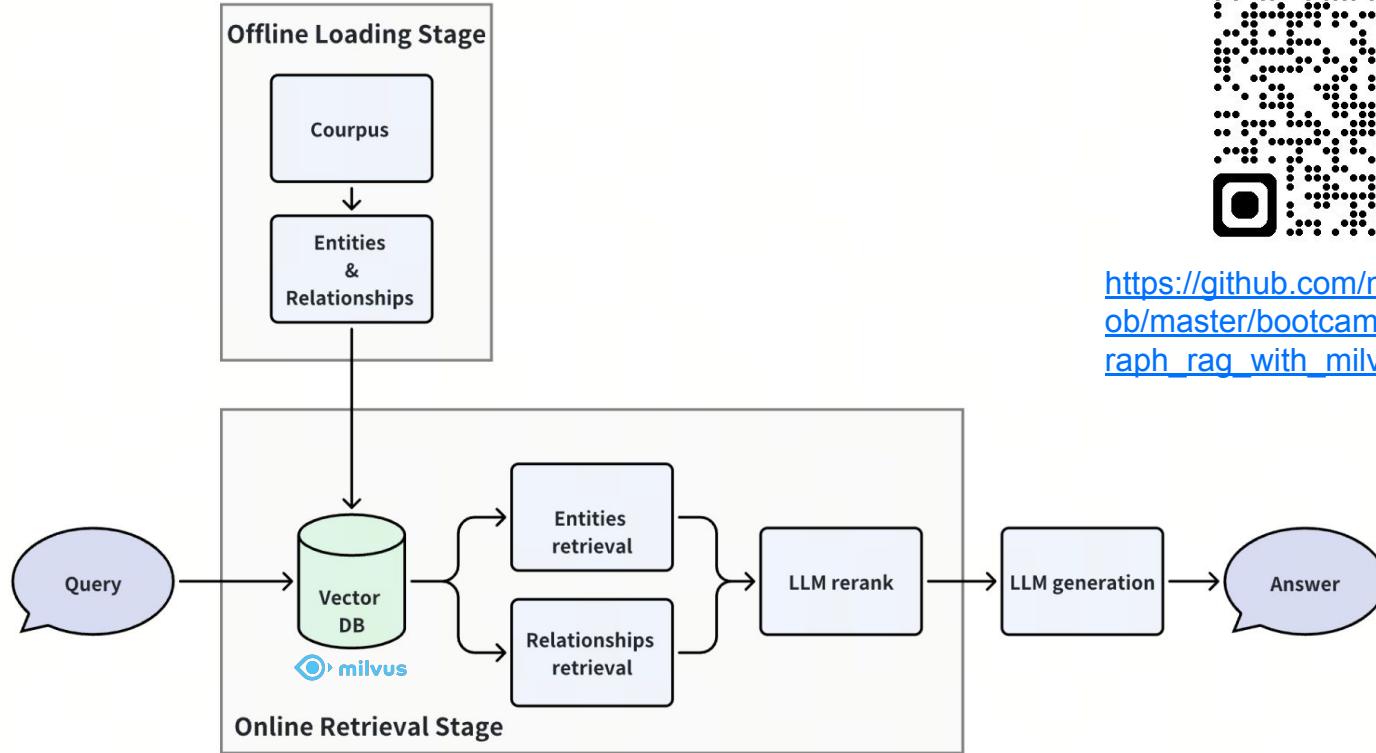
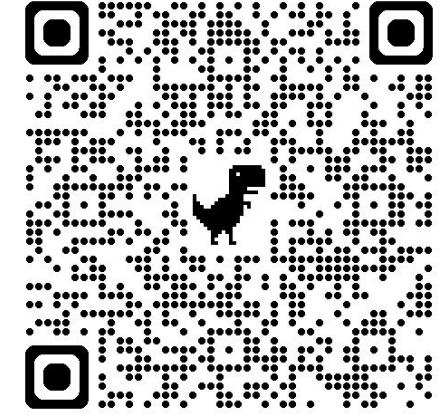


Picture Credit: KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation



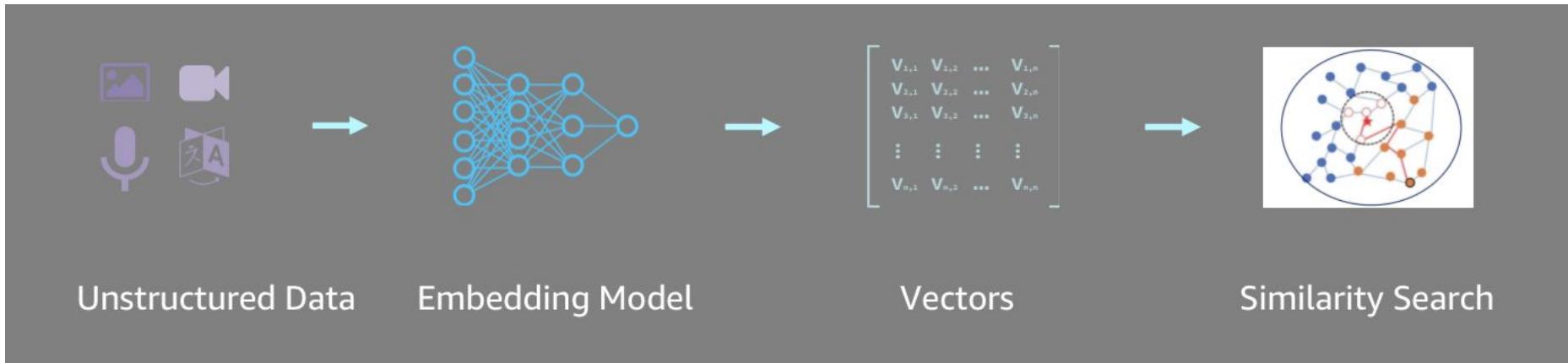
Picture Credit: <https://microsoft.github.io/graphrag/>

# Knowledge Embedding + Vector Search



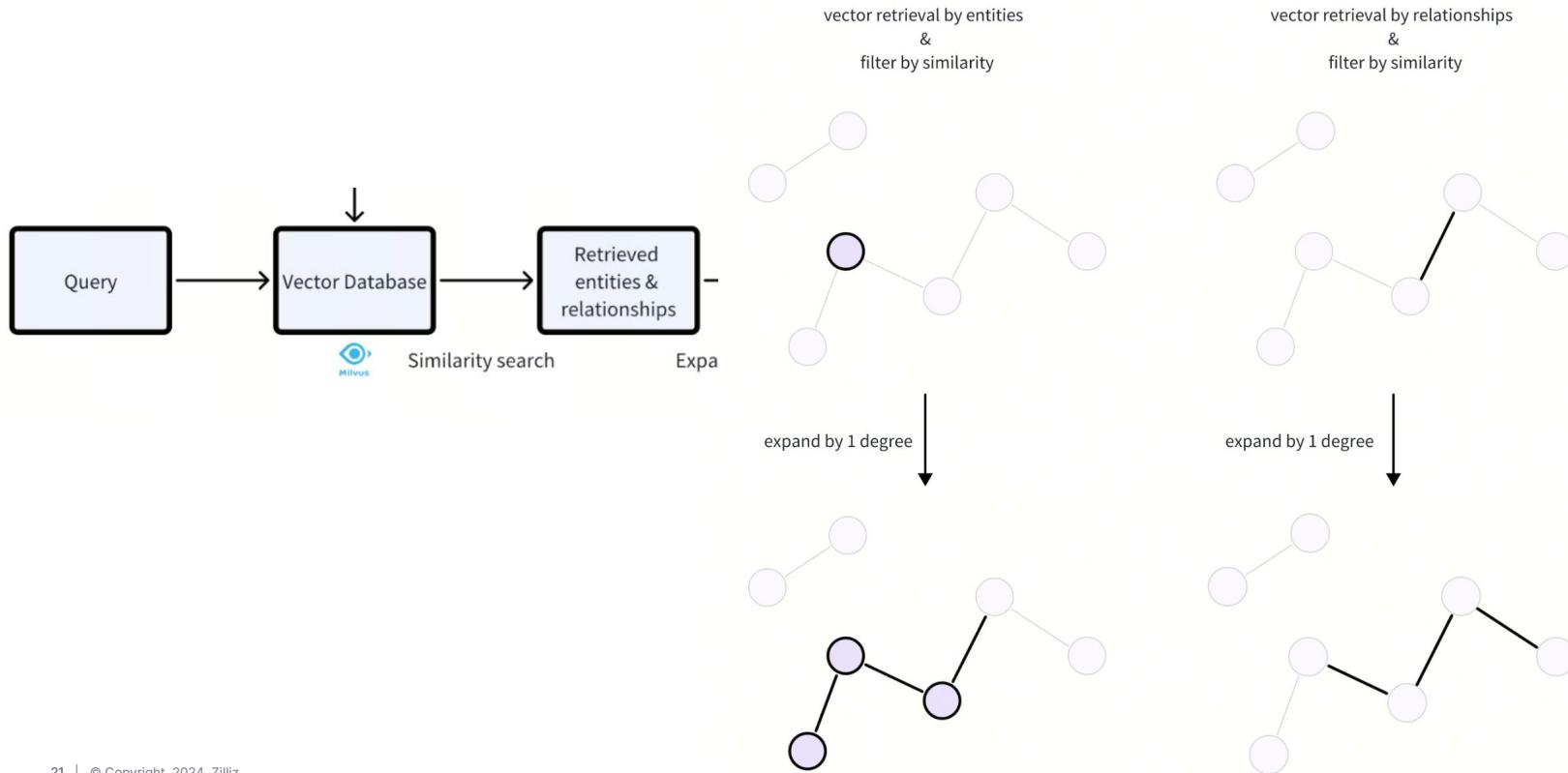
[https://github.com/milvus-io/bootcamp/blob/master/bootcamp/tutorials/quickstart/graph\\_rag\\_with\\_milvus.ipynb](https://github.com/milvus-io/bootcamp/blob/master/bootcamp/tutorials/quickstart/graph_rag_with_milvus.ipynb)

# How to Embed Knowledge Entities and Triples

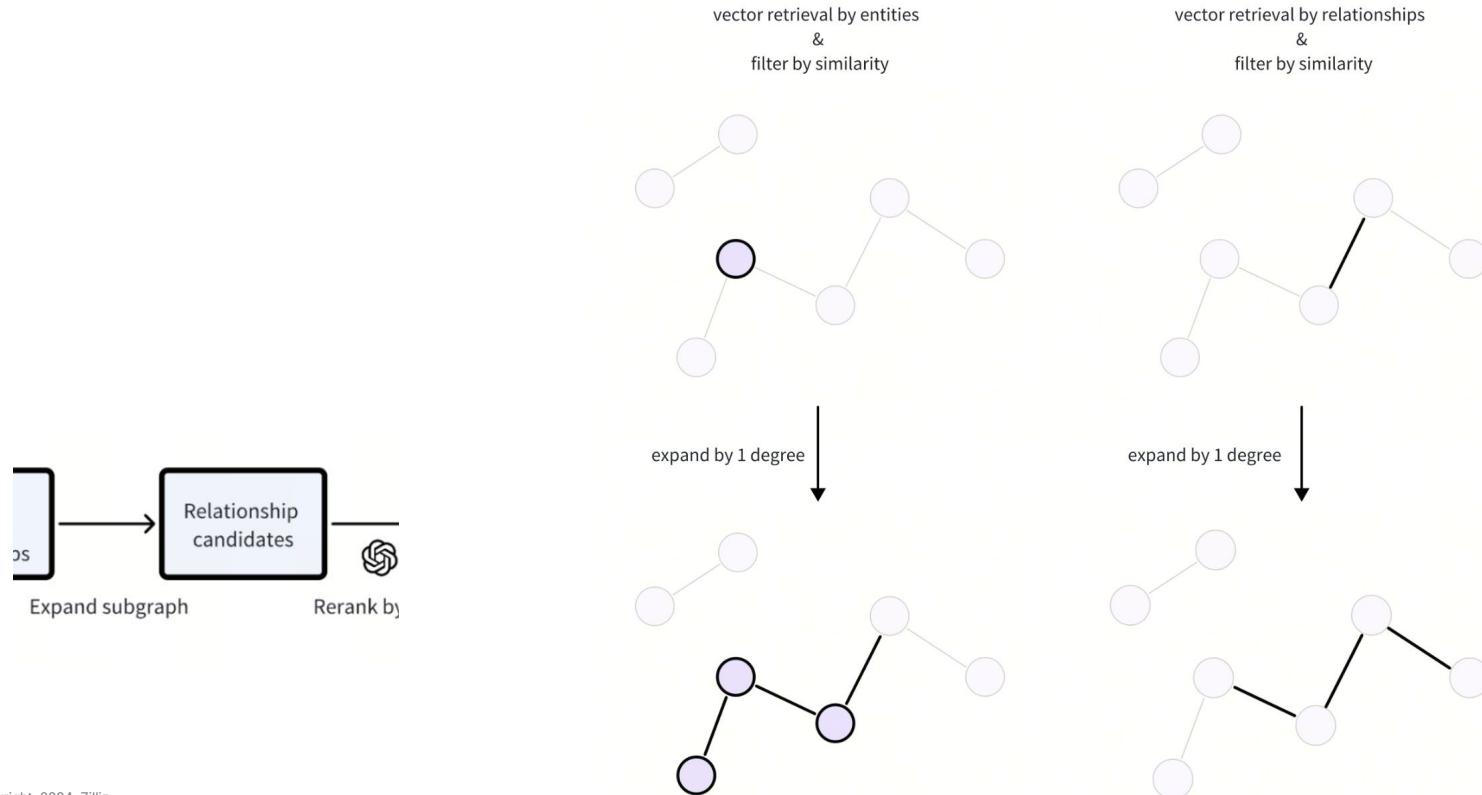


$(\text{Alex}, \text{child of}, \text{Brian}) \rightarrow \text{"Alex child of Brian"}$   
 $(\text{Cole}, \text{married to}, \text{Brian}) \rightarrow \text{"Cole married to Brian"}$

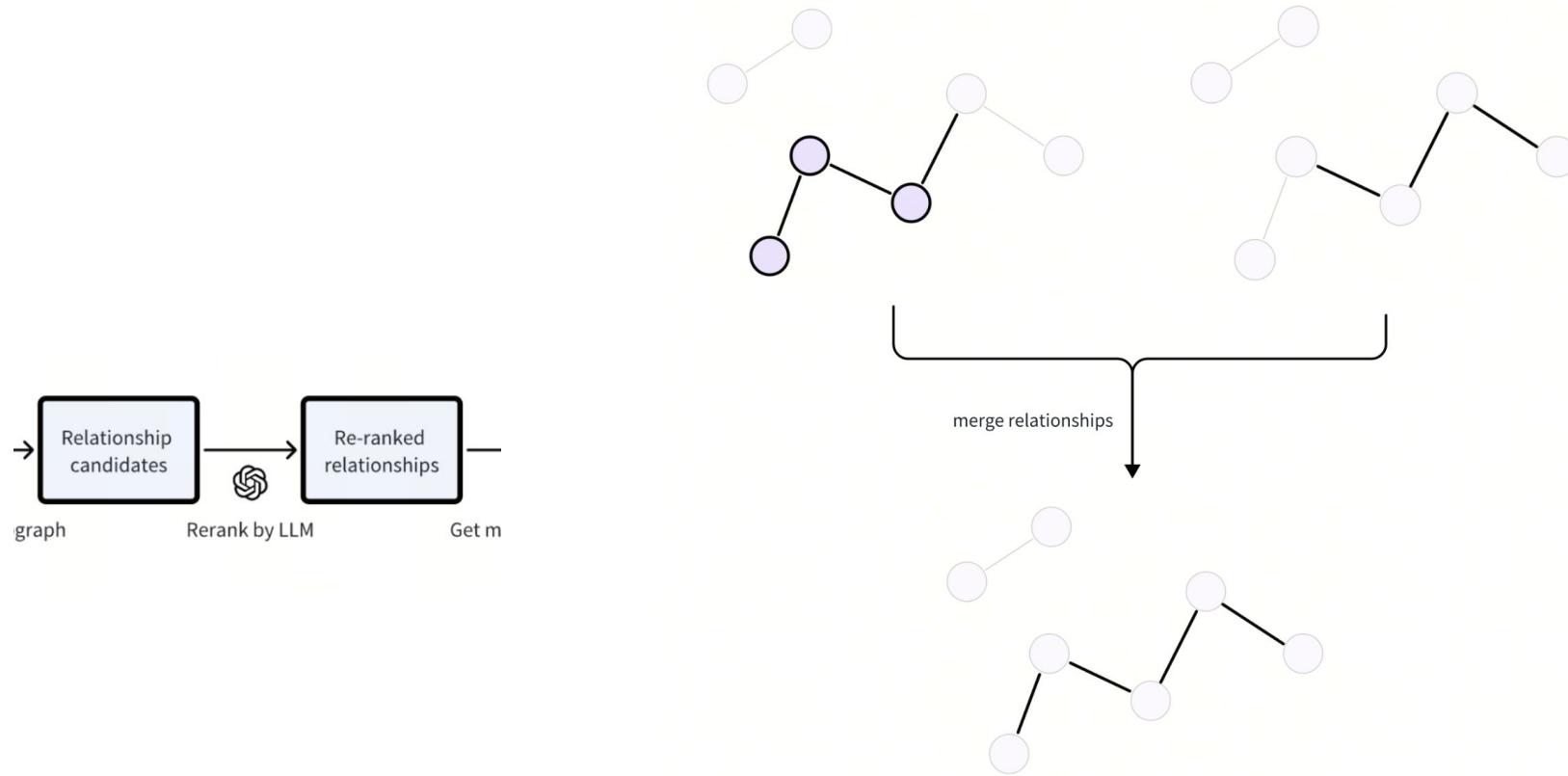
# Knowledge Retrieval with Vector Embedding



# Knowledge Retrieval with Vector Embedding

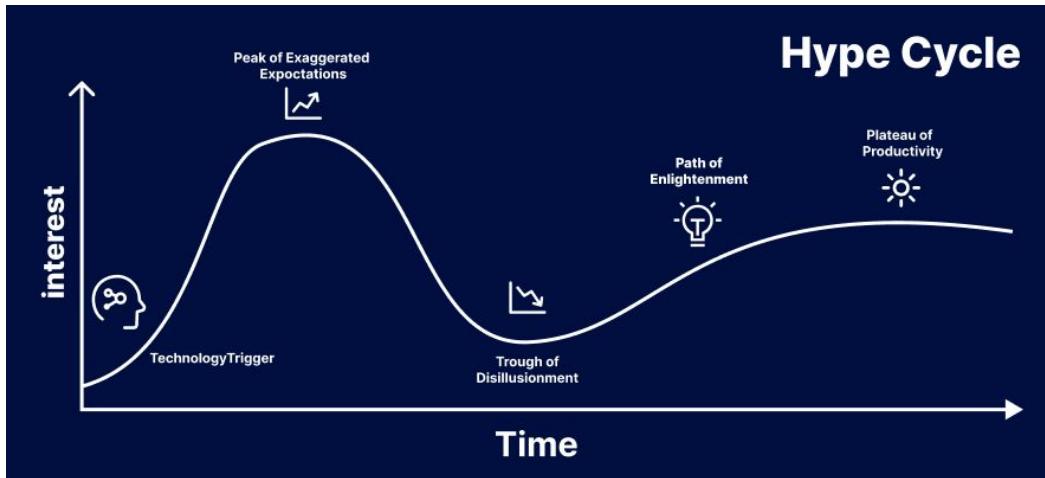


# Knowledge Retrieval with Vector Embedding



# Why is Milvus special

# AI Hype?



What do these companies that navigated the "trough of disillusionment" have in common?

Data Volumes.



# Unfortunately, Not All The data are born Equal

**Use Case:** Data Search

**Vectors:** 2 billion

**Req'ts:** 200 ms, Cost

**Index:** DiskANN for cost savings

**Use Case:** Drug Discovery

**Vectors:** 12 billion

**Req'ts:** High recall

**Index:** BIN\_FLAT

**Use Case:** Image Search

**Vectors:** 20 billion

**Req'ts:** High insertion, Cost

**Index:** Disk Based Index

**Use Case:** Product Recommender

**Vectors:** 10 Million

**Req'ts:** 5000 QPS

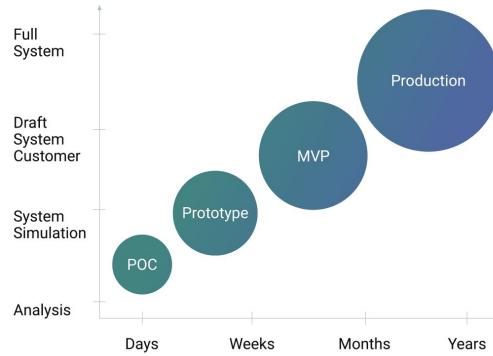
**Index:** HNSW&&Cagra



SmartNews

<https://milvus.io/use-cases>

# Challenges we see for Vector Search



## Challenge 1

Search Quality



## Challenge 2

Computation Intensity



## Challenge 3

Storage Cost



## Challenge 4

Data at Scale



## Challenge 5

Ease of Operation



## Challenge 6

Ecosystems && Ease of use



## Challenge 7

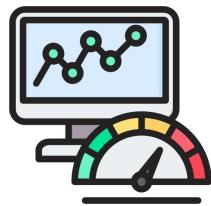
Ensuring Privacy and Security



# That's why we build Milvus

# And it's open sourced under Apache license!

Fast & Cost effective



**3X faster, 3X  
Cheaper**

Pluggable Vector Search Lib  
Tiered Storage

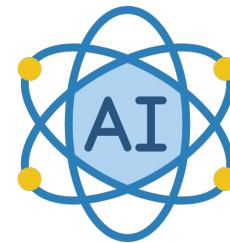
Scalable & Reliable



**Cloud Native,  
K8s Native**

Scale from 1 - 10B  
Storage computation disaggregation

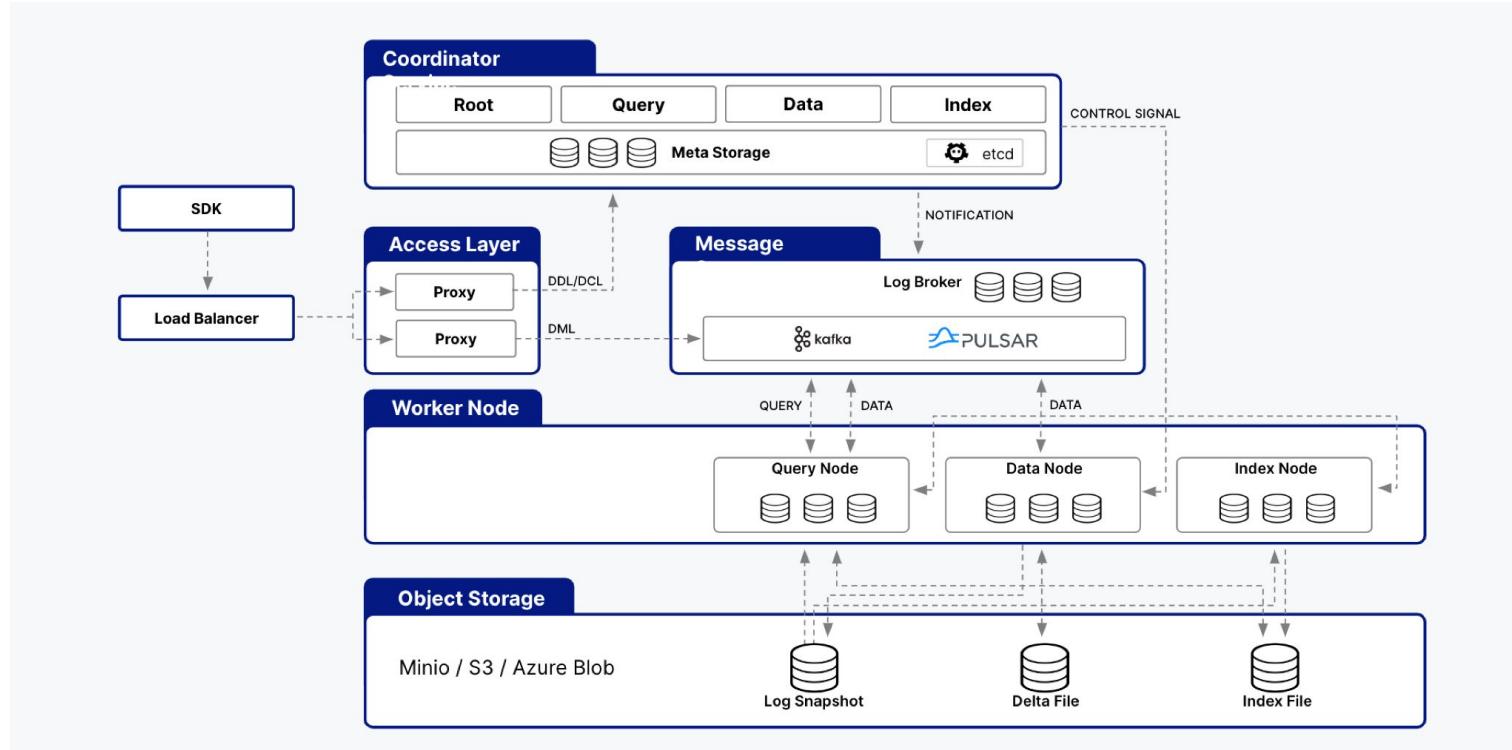
AI Powered



**Vector Native**

Rich functionality for AI  
Born for vector data processing

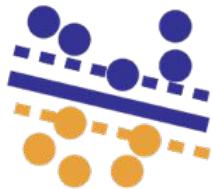
# Milvus Architecture



# Rich functionality



Dynamic Schema



Float, Binary and  
Sparse Vector



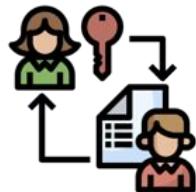
Tag + Vector  
Optimized filtering



Hybrid Search  
Sparse + Dense



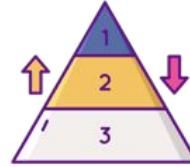
RBAC  
TLS, Encryption



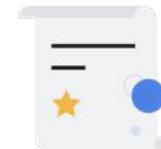
Million level  
Tenants support



Disk based Index



Tiered Storage

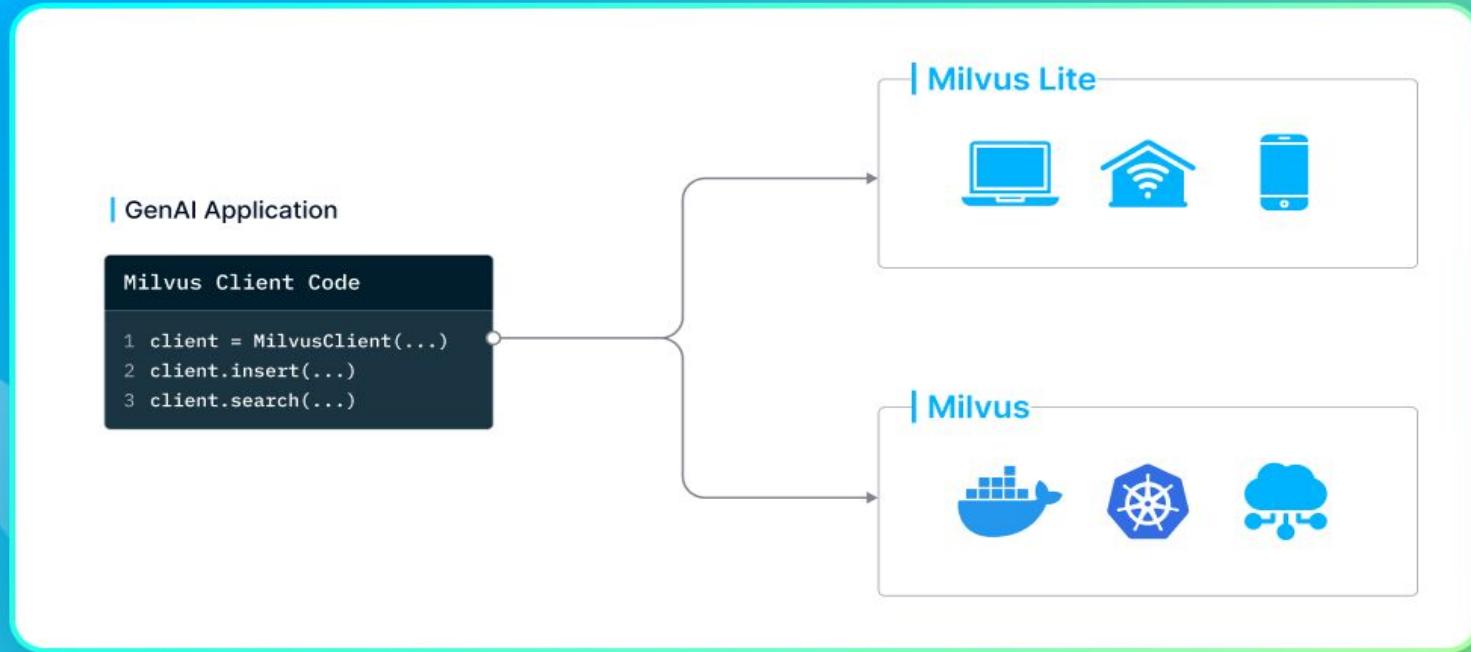


Support bulk import



GPU Support  
Intel + ARM Cpu Support

# Build Once Deploy Anywhere



# Easy to start with, can even run on edge devices!

## Install Milvus

In this guide we use Milvus Lite, a python library included in `pymilvus` that can be embedded into the client application. Milvus also supports deployment on [Docker](#) and [Kubernetes](#) for production use cases.

Before starting, make sure you have Python 3.7+ available in the local environment. Install `pymilvus` which contains both the python client library and Milvus Lite:

```
$ pip install -U pymilvus
```



## Set Up Vector Database

To create a local Milvus vector database, simply instantiate a `MilvusClient` by specifying a file name to store all data, such as "milvus\_demo.db".

```
from pymilvus import MilvusClient  
  
client = MilvusClient("milvus_demo.db")
```



# Scale-up on Docker

## Install Milvus in Docker

Milvus provides an installation script to install it as a docker container. The script is available in the [Milvus repository](#). To install Milvus in Docker, just run

```
# Download the installation script
$ curl -sfL https://raw.githubusercontent.com/milvus-io/milvus/master/scripts/standalone_embed.sh

# Start the Docker container
$ bash standalone_embed.sh start
```

After running the installation script:

- A docker container named milvus has been started at port **19530**.
- An embed etcd is installed along with Milvus in the same container and serves at port **2379**. Its configuration file is mapped to **embedEtc.yaml** in the current folder.
- The Milvus data volume is mapped to **volumes/milvus** in the current folder.

```
client = MilvusClient(
    uri="http://localhost:19530",
    token="root:Milvus"
)
```

# Up to 100 billion vectors with K8s!

You can install Milvus Operator in either of the following ways:

- [With Helm](#)
- [With kubectl](#)

## Install with Helm

Run the following command to install Milvus Operator with Helm.

```
$ helm install milvus-operator \
-n milvus-operator --create-namespace \
--wait --wait-for-jobs \
https://github.com/zilliztech/milvus-operator/releases/download/v0.9.15/milvus-operator-0.9.15.tgz
```

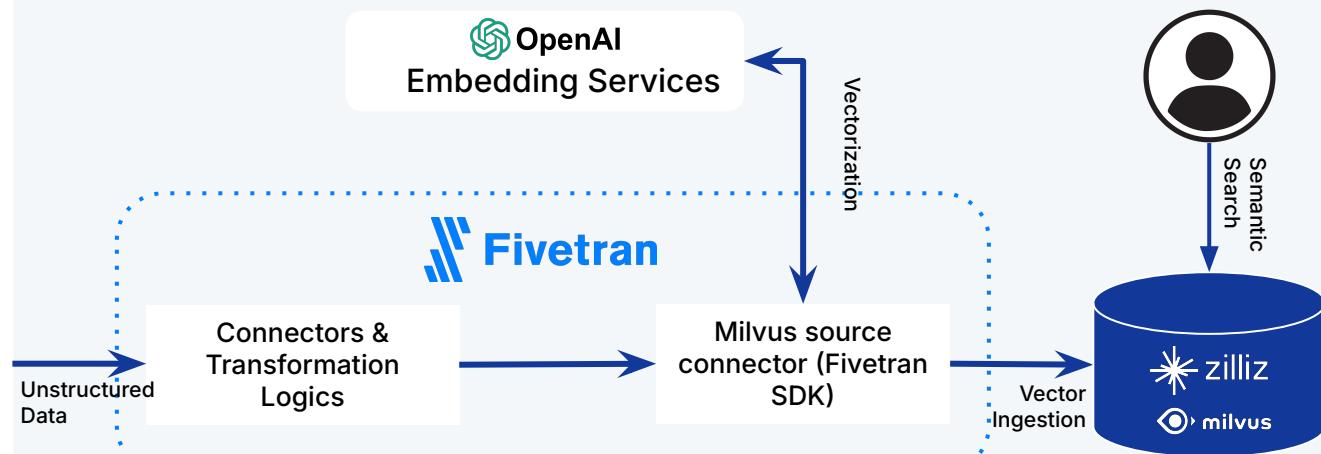
You will see the output similar to the following after the installation process ends.

```
NAME: milvus-operator
LAST DEPLOYED: Thu Jul 7 13:18:40 2022
NAMESPACE: milvus-operator
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Milvus Operator Is Starting, use `kubectl get -n milvus-operator deploy/milvus-operator` to check if
If Operator not started successfully, check the checker's log with `kubectl -n milvus-operator logs
Full Installation doc can be found in https://github.com/zilliztech/milvus-operator/blob/main/docs/
Quick start with `kubectl apply -f https://raw.githubusercontent.com/zilliztech/milvus-operator/main/deploy/helm
More samples can be found in https://github.com/zilliztech/milvus-operator/tree/main/config/samples
CRD Documentation can be found in https://github.com/zilliztech/milvus-operator/tree/main/docs/CRD
```

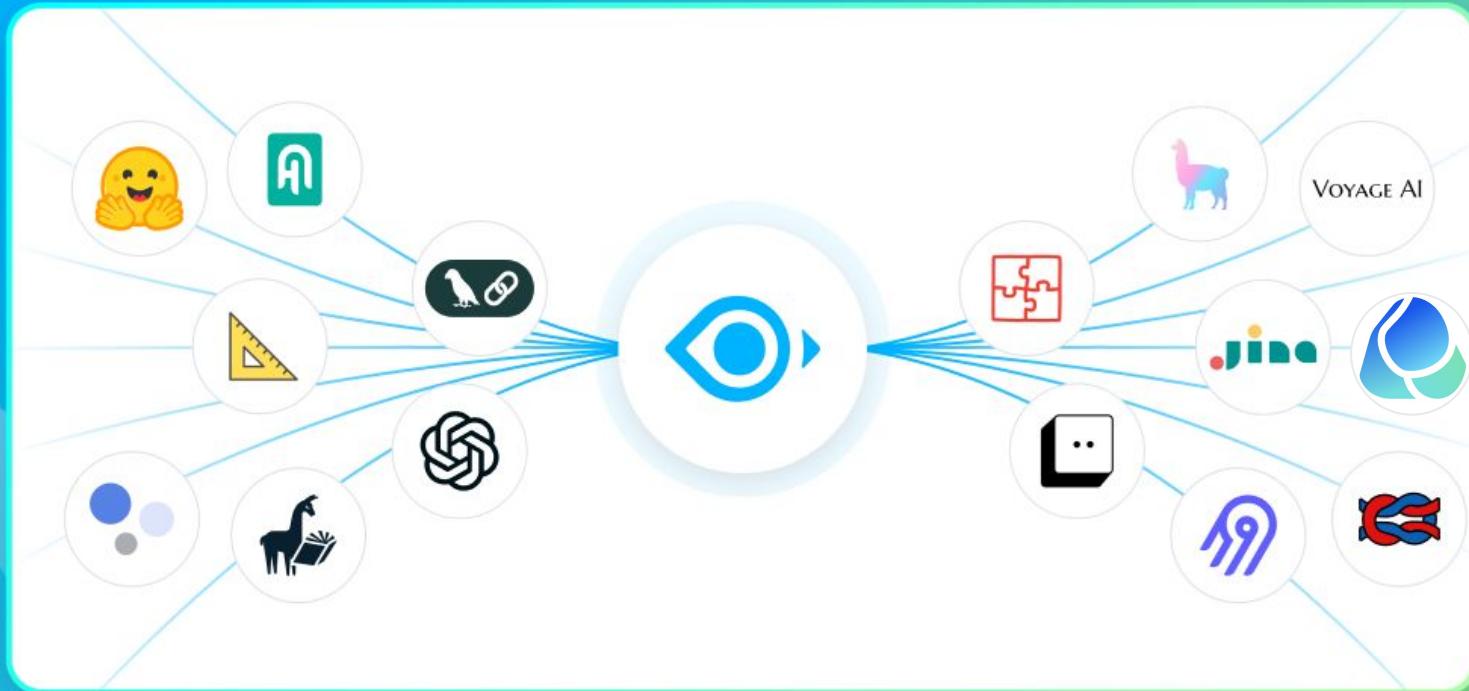
# Fivetran Source Connector

Simplify unstructured data retrievals from 500+ system through Fivetran

500+ data sources



# Seamless integration with all popular AI toolkits



[https://milvus.io/docs/integrations\\_overview.md](https://milvus.io/docs/integrations_overview.md)

Milvus | LF AI

Why Milvus Docs Tutorials Tools Blog Community

Star 29.4K Get Started

Search

Home v2.4.x

About Milvus

Get Started

Concepts

User Guide

Models

Milvus Migration

Administration Guide

Tools

Integrations

Overview

Embedding Models

LLMs

Orchestration

## Integrations Overview

This page provides a list of tutorials for you to interact with Milvus and third-party tools.

Tutorial	Use Case	Partners or Stacks
RAG with Milvus and LlamaIndex	RAG	Milvus, LlamaIndex
RAG with Milvus and LangChain	RAG	Milvus, LangChain
Milvus Hybrid Search Retriever in LangChain	Hybrid Search	Milvus, LangChain
Semantic Search with Milvus and OpenAI	Semantic Search	Milvus, OpenAI
Question Answering Using Milvus and Cohere	Semantic Search	Milvus, Cohere
Question Answering using Milvus and HuggingFace	Question Answering	Milvus, HuggingFace
Image Search using Milvus and Pytorch	Semantic Search	Milvus, Pytorch

English

Table of contents

Integrations Overview

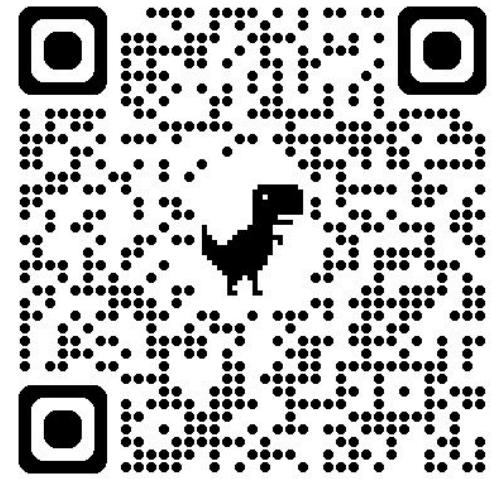
Edit this page

Create an issue

Feedback

Was this page helpful?

Ask AI



# Thank You !

**Milvus**

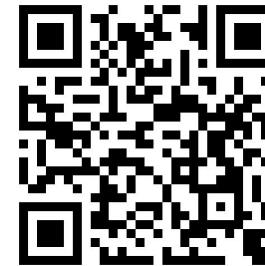
Open Source Self-Managed



[github.com/milvus-io/milvus](https://github.com/milvus-io/milvus)

**Zilliz Cloud**

SaaS Fully-Managed



[zilliz.com/cloud](https://zilliz.com/cloud)



[milvus.io](https://mилvus.io) for tutorials/notebooks