

# 向量数据库： 大模型的长期记忆体

李晨

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

# 讲师介绍



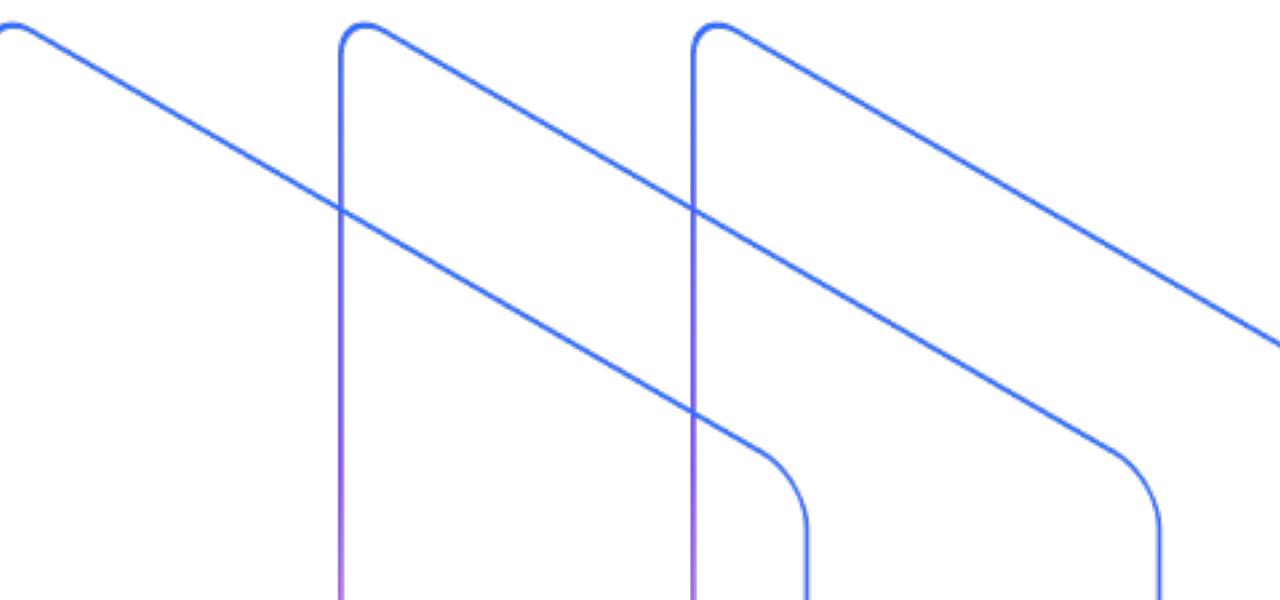
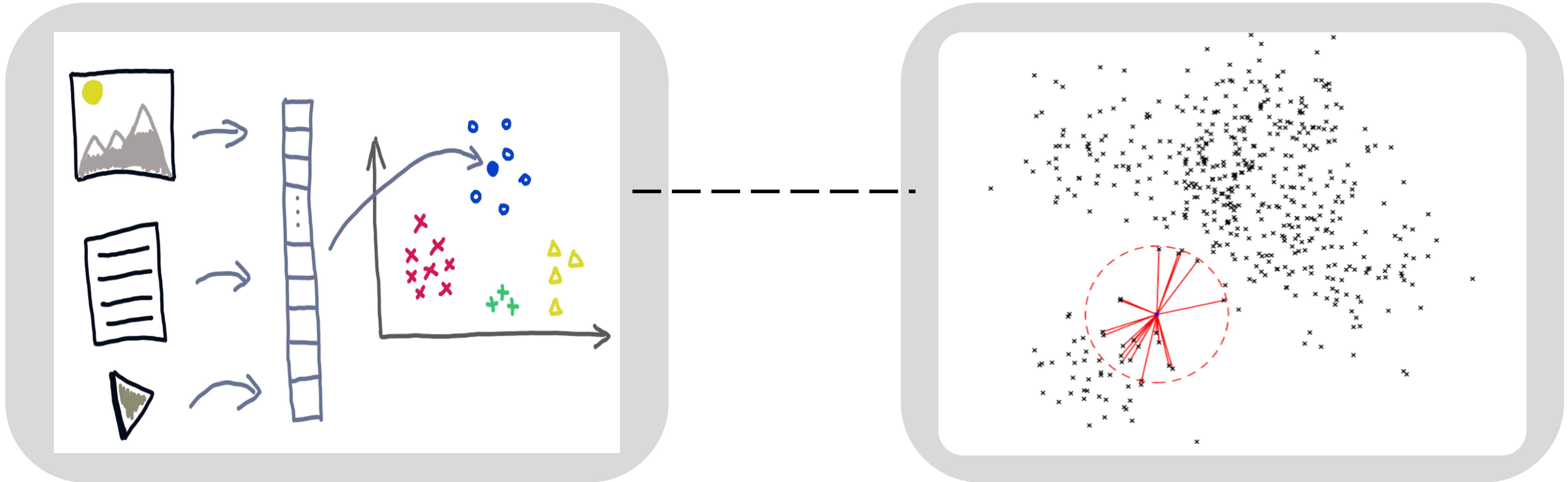
李晨

Zilliz 开发者关系及市场运营负责人，客户成功与生态发展负责人。

Linux Foundation APAC Evangelist。曾任开源中国社区负责人，Linux Foundation 开源软件学园运营负责人。

多年来从事开源项目和社区的组织、管理工作，有着丰富的市场和运营经验，策划及参与执行多个国内外顶级基金会在内的落地和生态拓展工作。

# Embedding and Vector Search



# LLM Limitations

RE

What are the limitations of ChatGPT?



As of my knowledge cutoff date in September 2021, ChatGPT, which typically refers to versions of the GPT (Generative Pre-trained Transformer) language model used in conversational contexts, has several limitations:

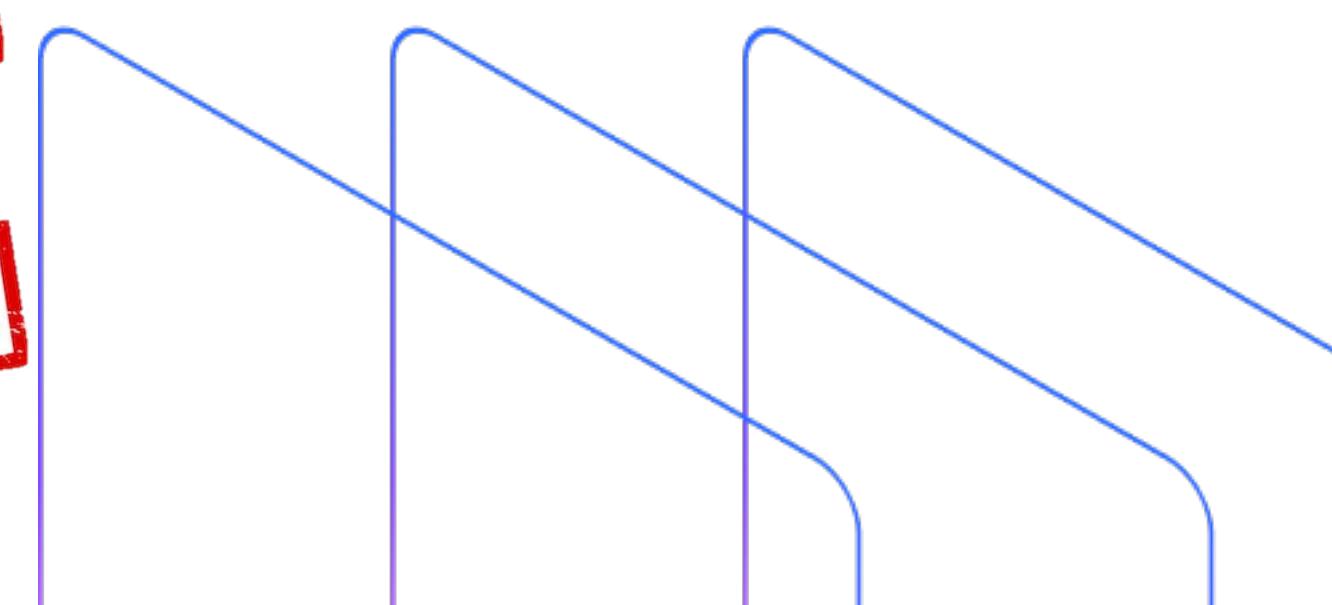
1. Lack of Ground Truth: ChatGPT does not have access to real-world knowledge beyond its training data, and it cannot verify the truthfulness or accuracy of its responses. It may generate plausible-sounding but incorrect or misleading answers.

Domain-specific knowledge

**WRONG!**

Real-time information

**WRONG!**



# Knowledge Retrieval for LLMs



We're also hosting two plugins ourselves, a [web browser](#) and [code interpreter](#). We've also open-sourced the code for [a knowledge base retrieval plugin](#), to be self-hosted by any developer with information with which they'd like to augment ChatGPT.

## Retrieval

The open-source retrieval plugin enables ChatGPT to access personal or organizational information sources (with permission). [It allows users to obtain the most relevant document snippets from their data sources](#), such as files, notes, emails or public documentation, by asking questions or expressing needs in natural language.

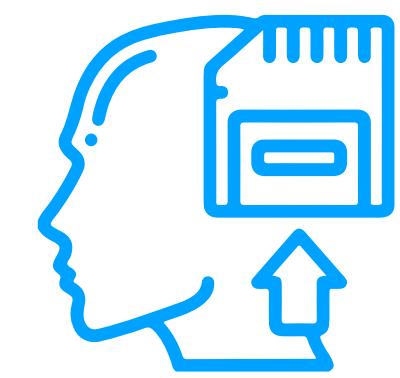
As an open-source and self-hosted solution, developers can deploy their own version of the plugin and register it with ChatGPT. The plugin leverages [OpenAI embeddings](#) and allows developers to choose a [vector database \(Milvus, Pinecone, Qdrant, Redis, Weaviate or Zilliz\)](#) for indexing and searching documents. Information sources can be synchronized with the database using webhooks.

# CVP framework: LLM + VectorDB + Prompt



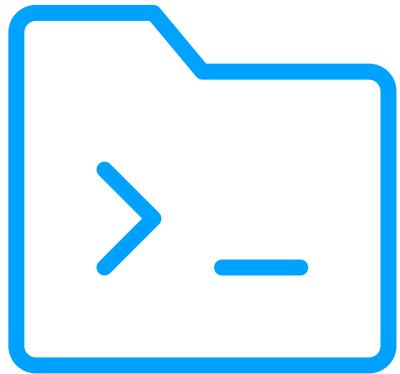
## LLM

Analysis and reasoning unit  
& natural language adapter



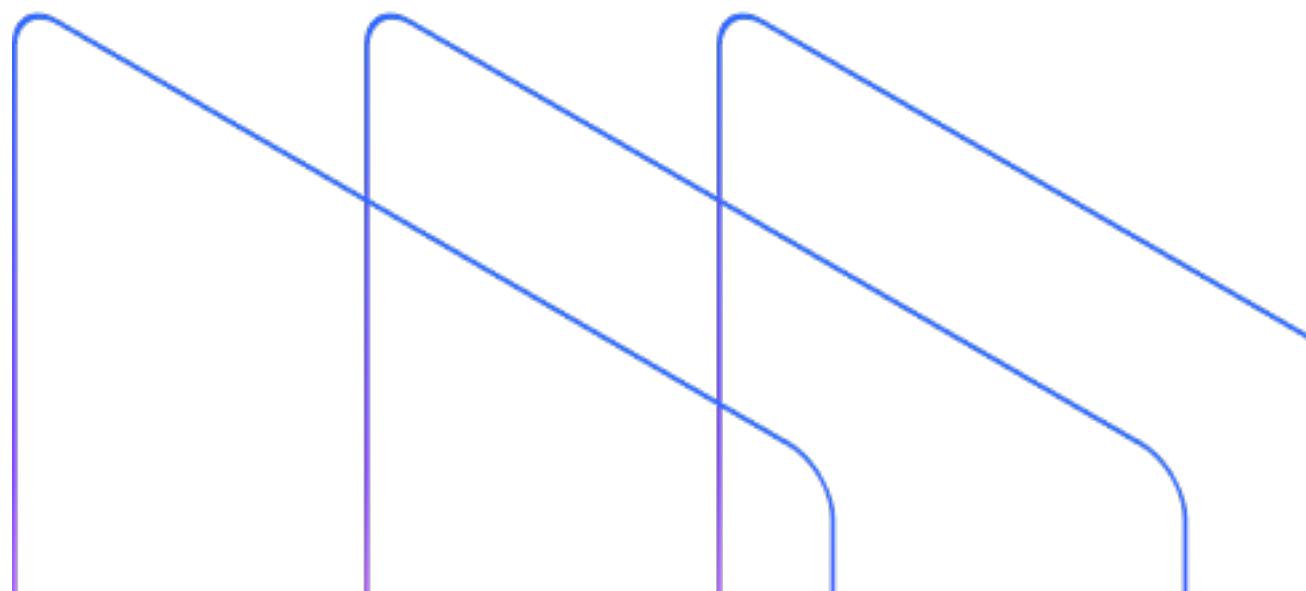
## VectorDB

Long-term memory  
& knowledgebase



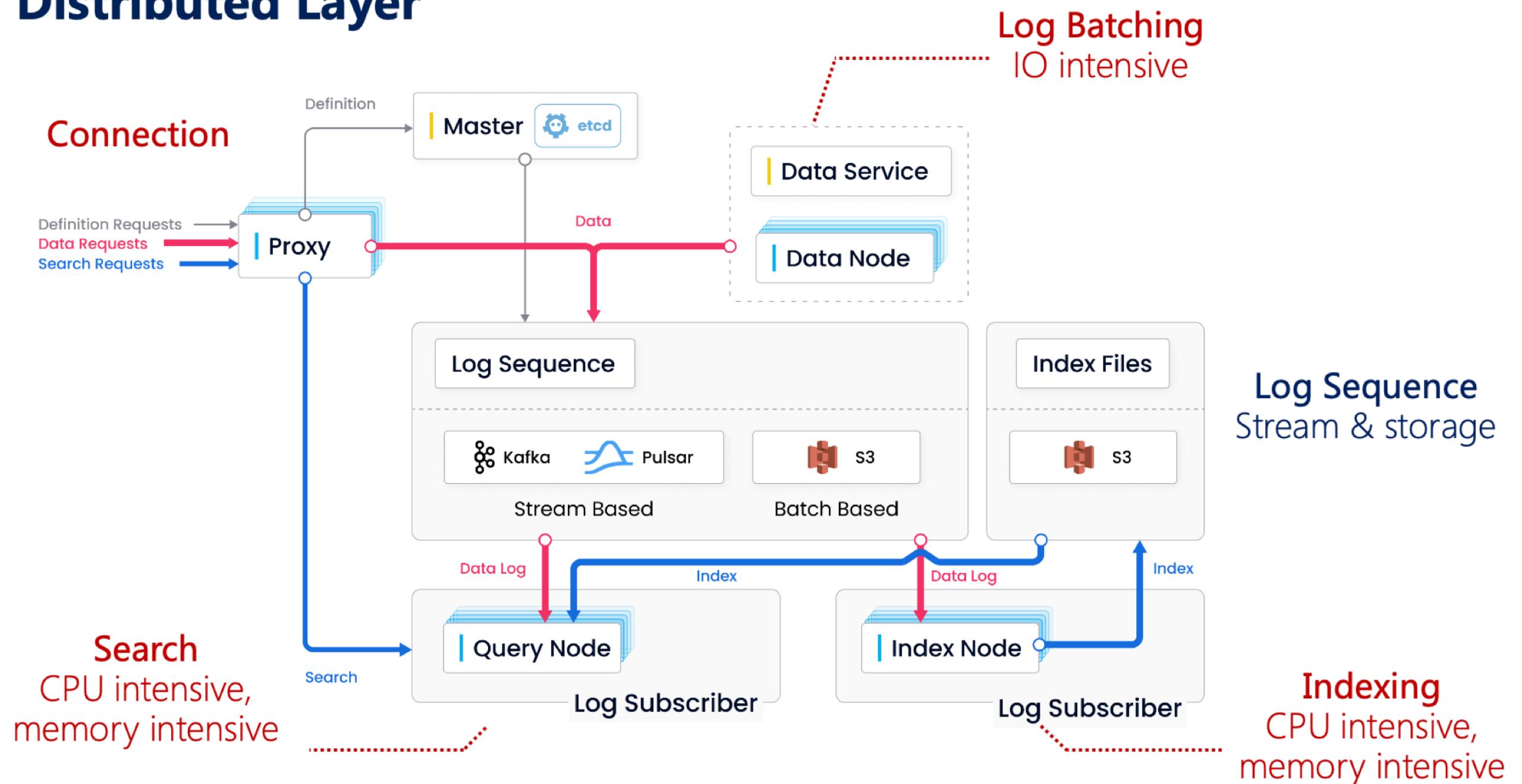
## Prompt

Business logic  
modeling



# Milvus: 为云而生的向量数据库

## Distributed Layer



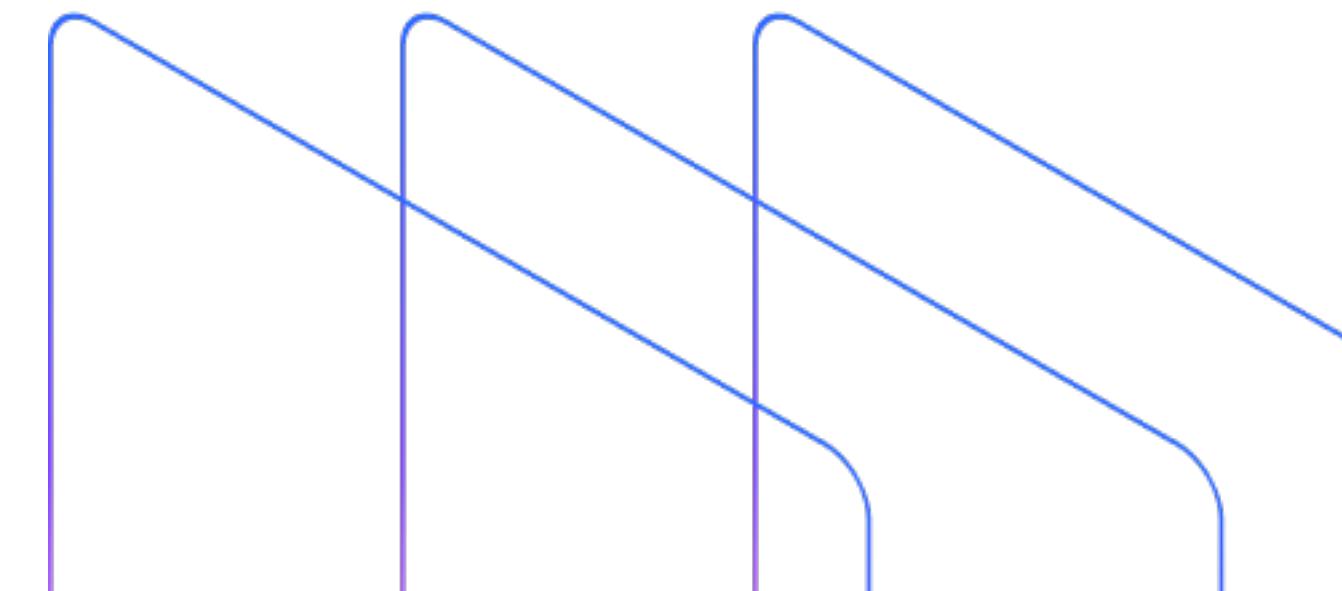
- 分布式云原生，基于 K8s 进行微服务化设计
- 存储计算分离，弹性扩缩容
- 高可用，故障分钟级恢复
- 百亿级向量的扩展能力
- 基于消息队列实现数据的实时增删
- 集成 OpenAI, Langchain, Huggingface, Pytorch 等 AI 生态
- 强大的生态工具 – GUI, CLI, 监控, 备份



<https://towhee.io/>

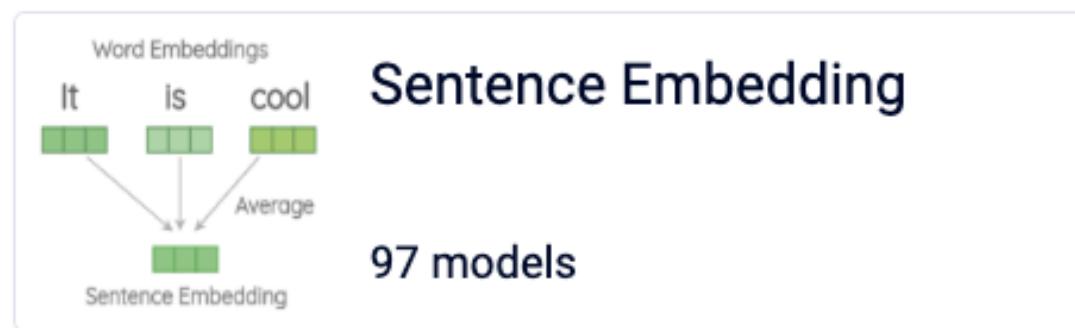
**Pipeline & Ecosystem Integration**

**LLM + VectorDB + Prompt chained together**



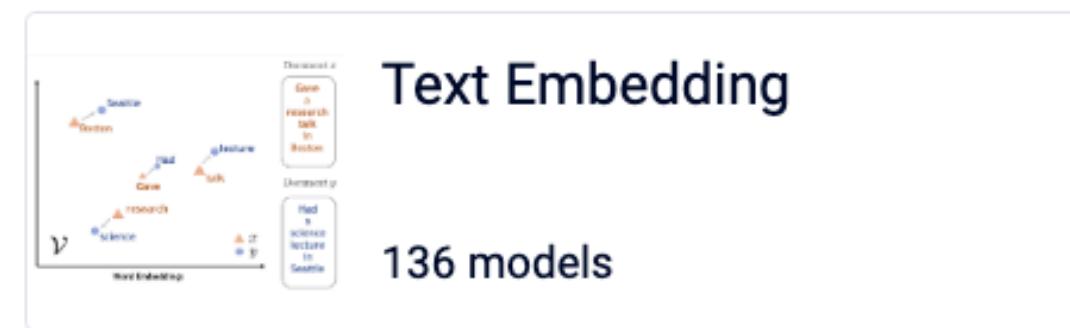
# Towhee Operators & Pipelines

## Natural Language Processing



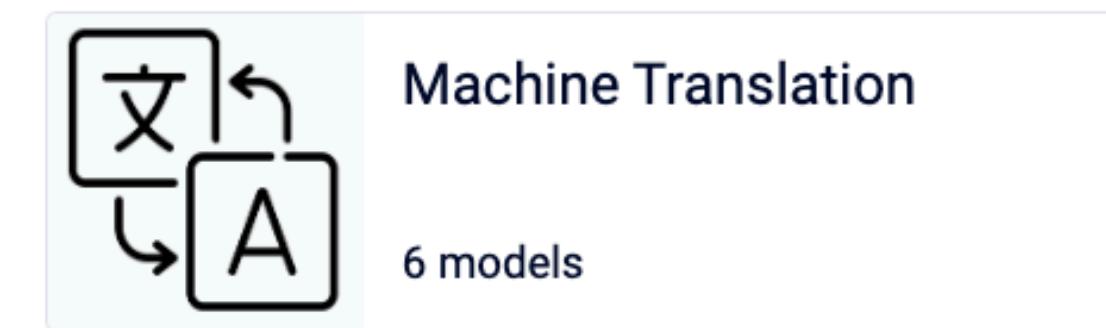
Sentence Embedding

97 models



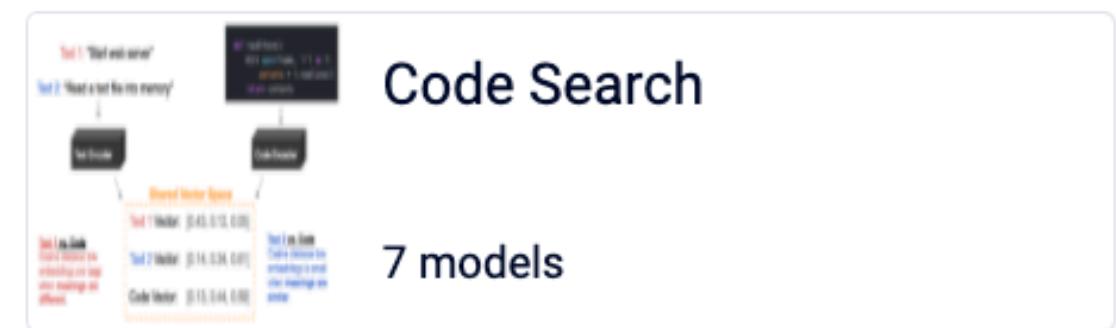
Text Embedding

136 models



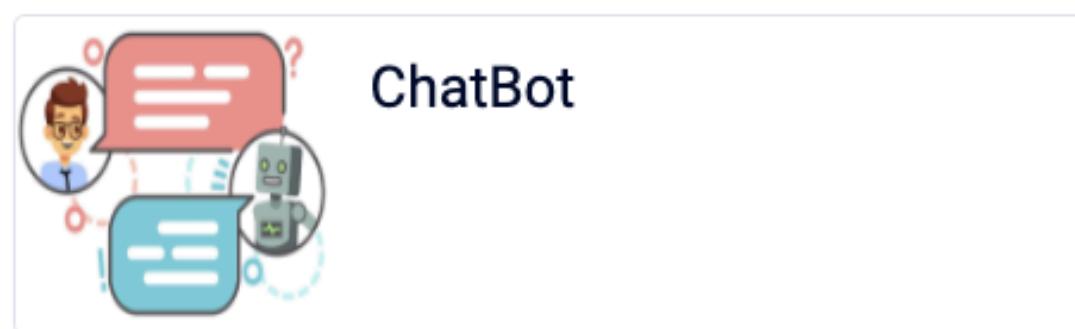
Machine Translation

6 models



Code Search

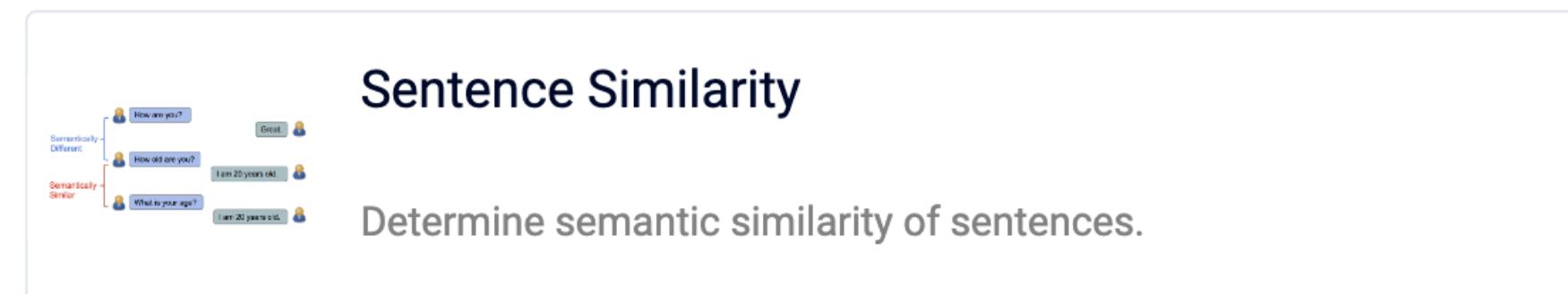
7 models



ChatBot

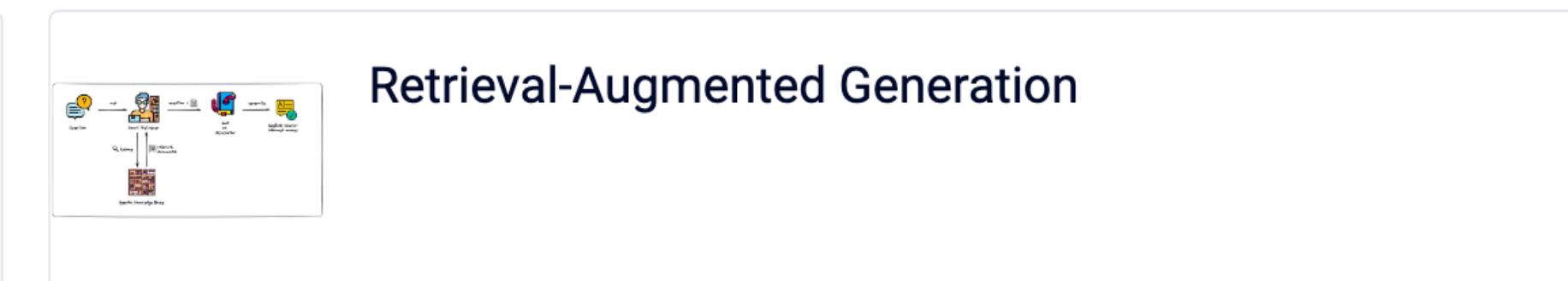
## Natural Language Processing

## Operators



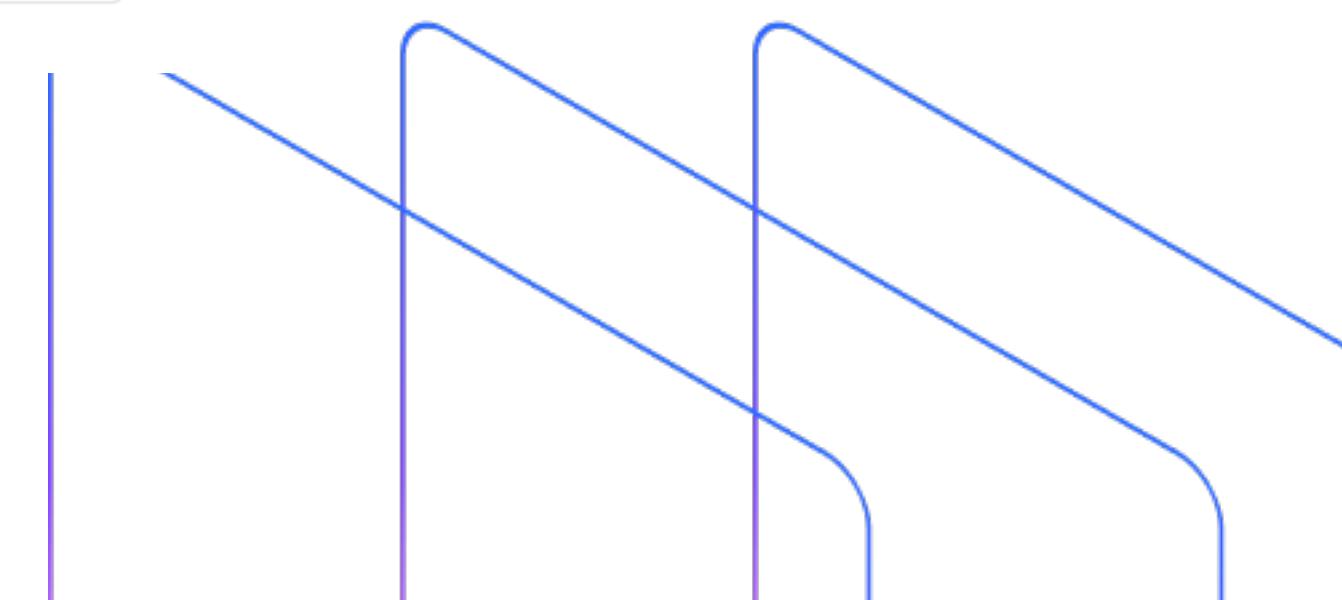
Sentence Similarity

Determine semantic similarity of sentences.



Retrieval-Augmented Generation

## Pipelines



# Zilliz – 构建开源+云的非结构化数据处理方案



## Milvus

Milvus is a vector database used to store, index, and manage massive embedding vectors generated by deep neural networks and other machine learning (ML) models.



## Towhee

Towhee makes it easy to build neural data processing pipelines for AI applications. With hundreds of models, algorithms, and transformations, Towhee helps you encode your unstructured data into embeddings.



## Attu

Attu is an open-source management tool for Milvus with an intuitive GUI, allowing you to interact easily with your databases. With just a few clicks, you can visualize your cluster status, manage metadata, perform data queries, and much more.



围绕 LLM 的开源非结构化数据解决方案

Zilliz Cloud – 开箱即用的全托管向量检索服务

# Zilliz Cloud: 企业级特性

## 维护成本低

low maintenance cost

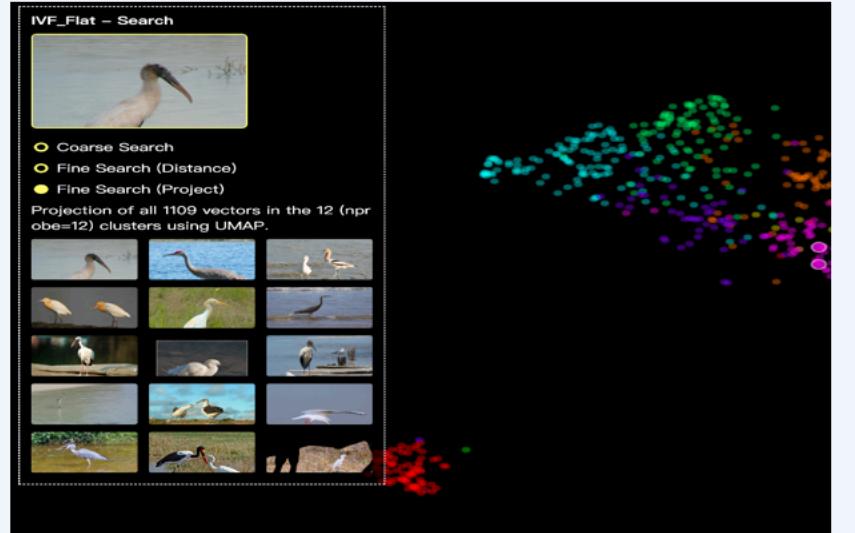
- 一键创建实例资源
- 动态扩缩容
- 完善的监控报警
- 多云支持



## 使用门槛低

Low threshold for use

- 免费实例
- 可视化界面
- 多语言SDK
- 丰富的生态支持
- 数据迁移



## 丰富的企业级特性

Enterprise-level features

- 7 \* 24服务支持
- 99.9 SLA保障
- 数据备份, 订阅
- 组织架构管理
- Dedicated Cloud



## 安全放心

Data Security

- RBAC权限管理
- TLS, 白名单
- PrivateLink
- 审计日志
- SOC2合规认证



# Our Partners: Zilliz x AI x Cloud



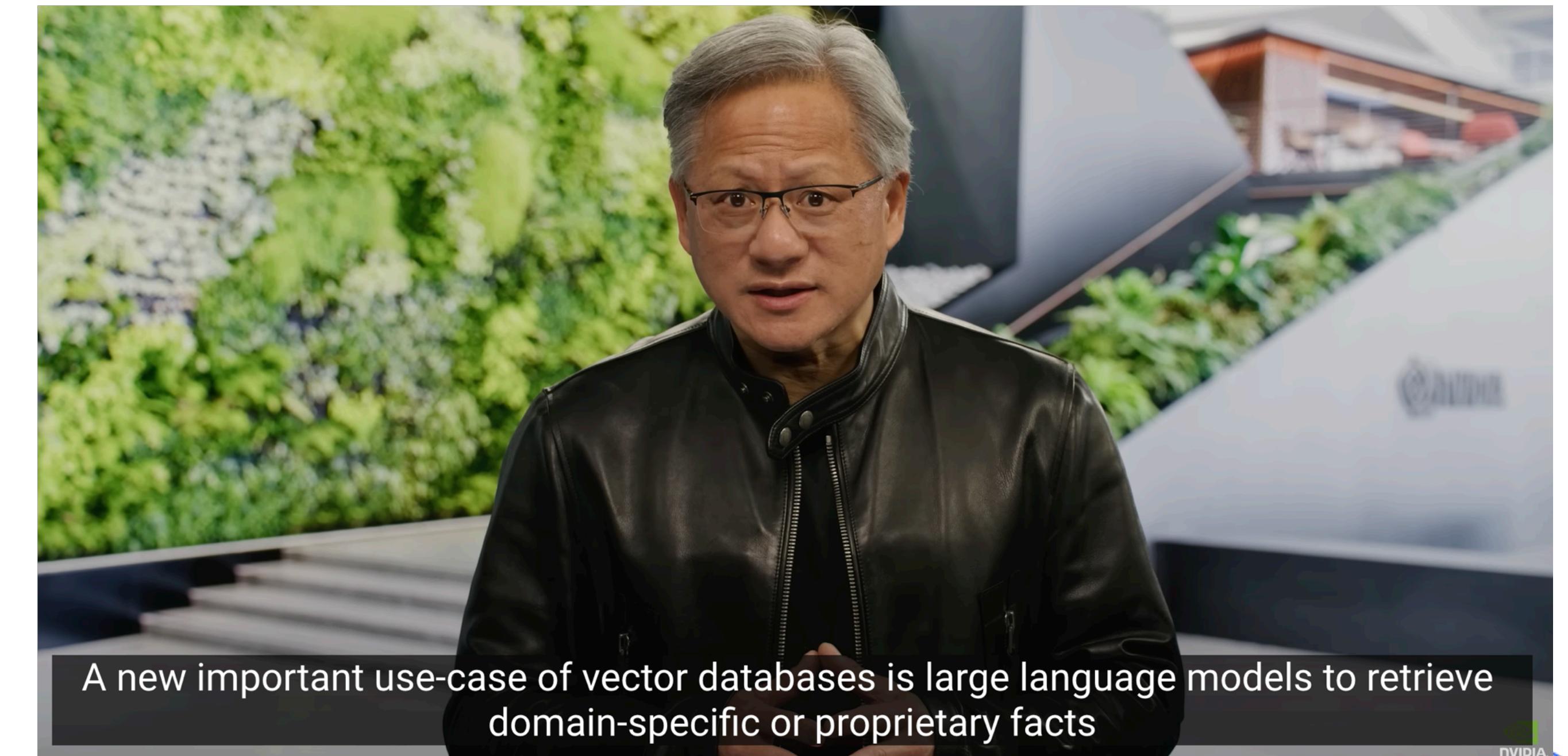
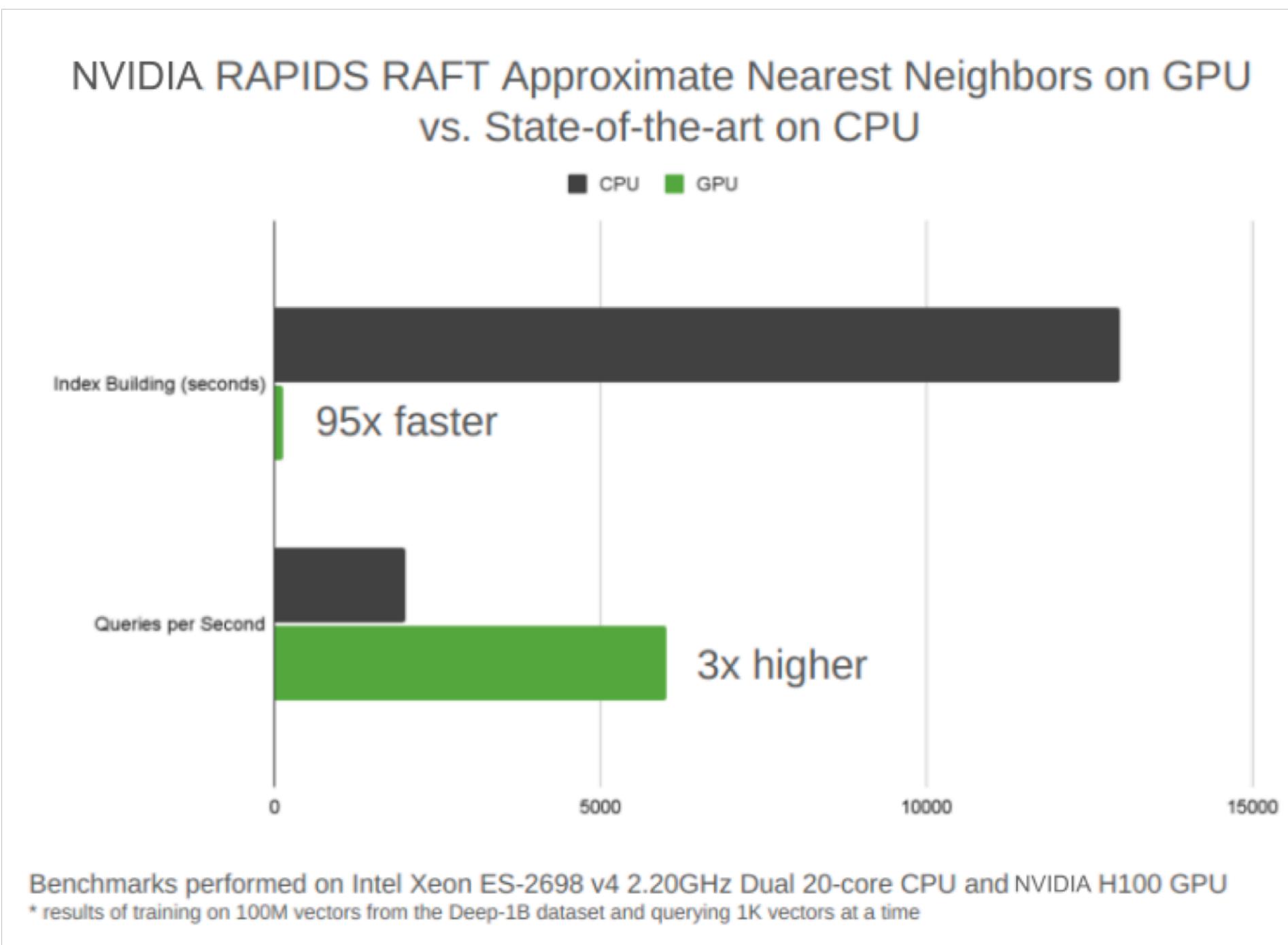
Milvus 被全球超过 **1000家** 企业用户所信赖，超过 **350万次** 下载和安装

Milvus Github Star 数目超过 **2万**，贡献者人数超过 **300**

Milvus DB-Engine引擎排名 **223**，并且在 SIGMOD 和 VLDB 等数据库顶会上发表了论文，奠定了向量数据库的基础

# VectorDB x 异构并行计算

- Milvus 与 Nvidia Rapids 合作，成为了全球第一个支持 GPU 的向量数据库，相比于 CPU 实现了3倍查询性能提升
- Milvus 团队与 AWS Graviton 合作，利用 ARM Neon 指令集加速，大幅提升了向量检索在 ARM CPU 上执行的性价比



# Answer from ChatGPT

RE

What is TTL in Milvus and how do I set it



As of my knowledge cutoff date in September 2021, Time to Live (TTL) is a feature in Milvus that allows users to automatically expire and delete data from a collection after a specified period of time. TTL can be useful in scenarios where you want to manage the size of a collection, such as when you want to keep only the most recent documents or remove older data.

As of Milvus v1.x, the TTL feature was supported and could be configured as a parameter in the collection configuration. However, in Milvus v2.x, the feature is not yet supported. The Milvus team was aware of the demand for this feature and planned to reintroduce it in a future version of Milvus v2.x, but as of my knowledge cutoff date, it has not been implemented.

If you are using Milvus v1.x, you can set the TTL for a collection by specifying the `'ttl'` parameter in the collection's configuration. Here is an example of how to create a collection using the Milvus Python SDK:

python

```
from milvus import Milvus, DataType

# Initialize Milvus client
client = Milvus(host='localhost', port='19530')

# Define collection schema
collection_name = 'example_collection'
dimension = 128
collection_param = {
    'collection_name': collection_name,
    'dimension': dimension,
    'index_file_size': 1024, # optional
    'metric_type': 'L2', # optional
    'ttl': 3600 # TTL in seconds (e.g., 1 hour)
}

# Create collection with TTL
status = client.create_collection(collection_param)
```

Copy code

# Answer from OSSChat

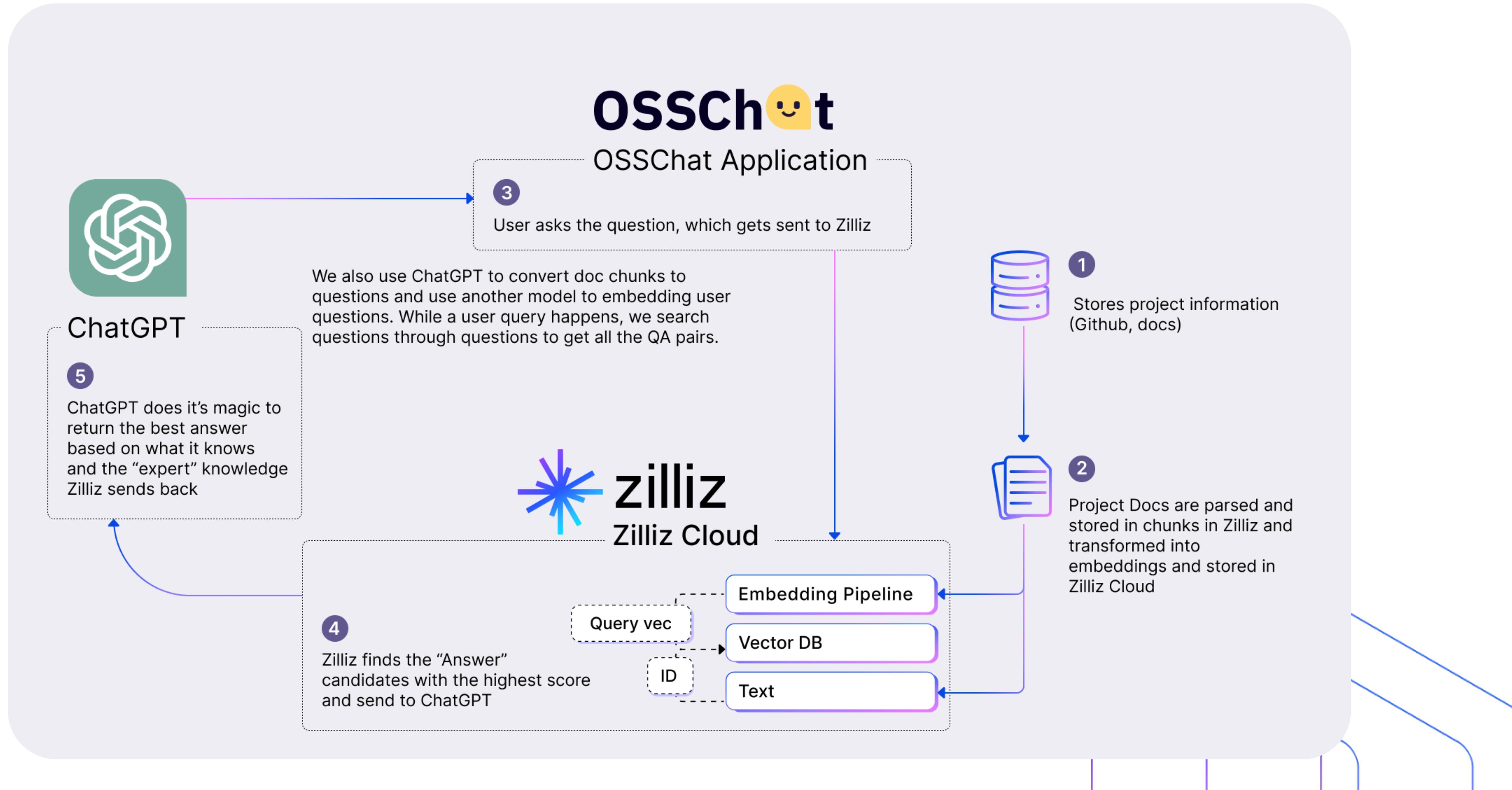


What is TTL in Milvus and how do I set it

To set TTL in Milvus, you can modify the properties of a collection using the `collection.set_properties` method in Python. The parameter for TTL is `collection.ttl.seconds`, which specifies the expiration time of data in a collection. Expired data in the collection will be cleaned up and will not be involved in searches or queries. The value should be 0 or greater, and the default value is 0, which means TTL is disabled. For more information on modifying collection properties, you can refer to the Milvus documentation.



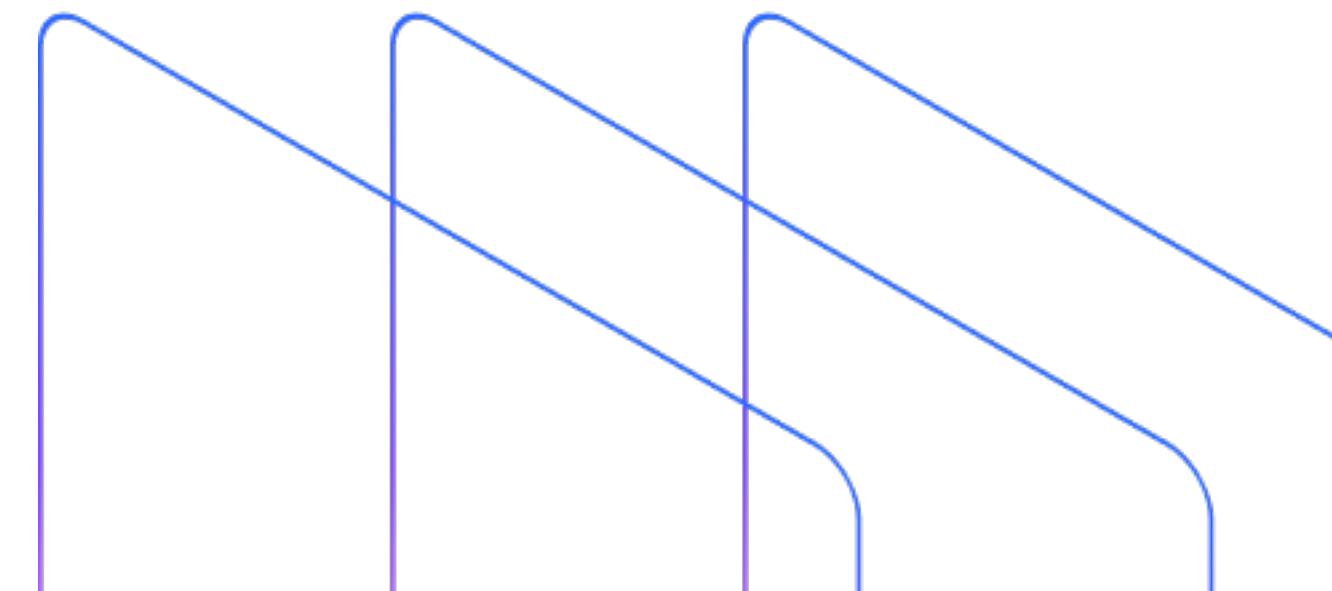
<https://osschat.io/>



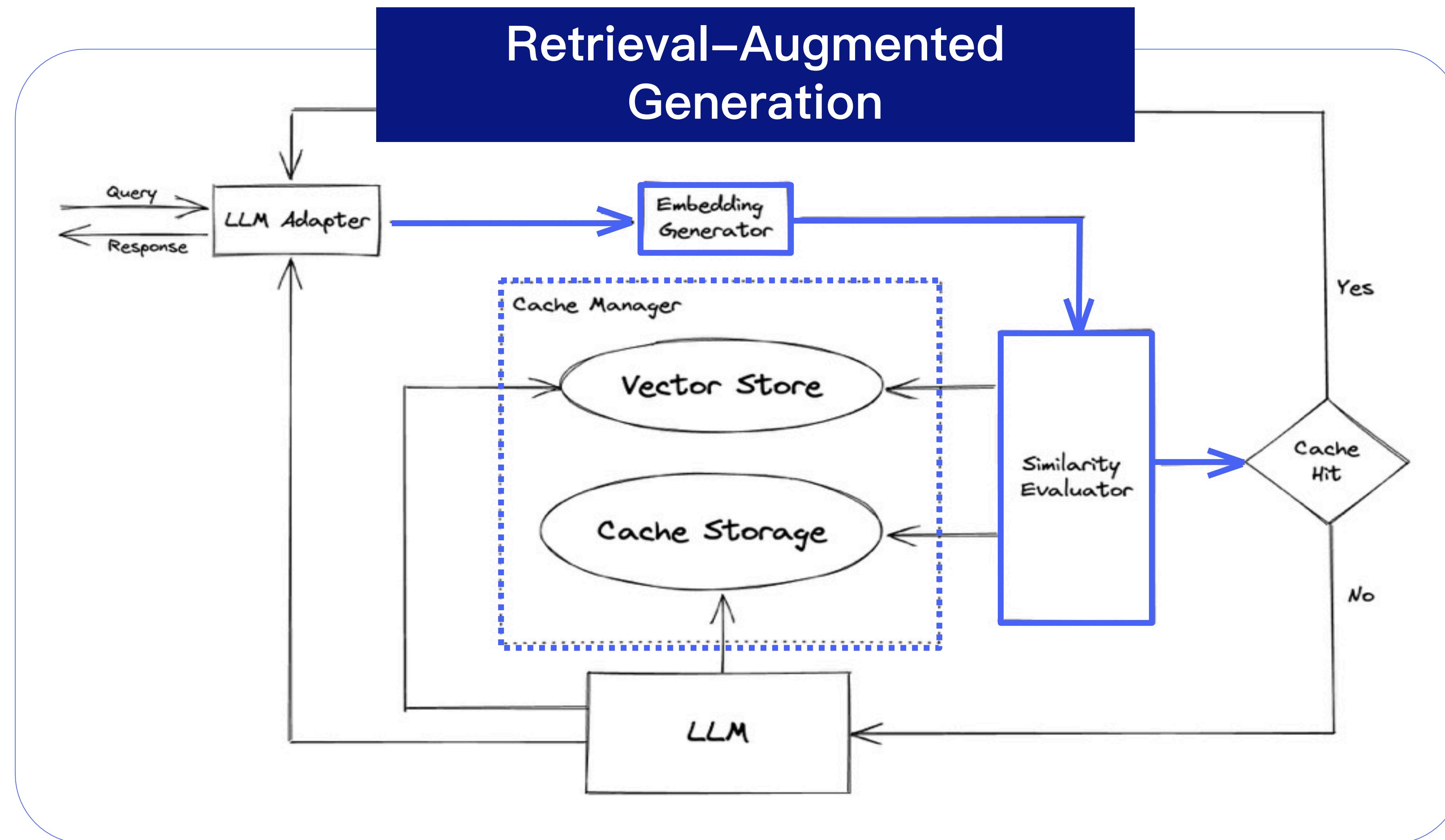
**GPTCache :**

**A Library for Creating Semantic Cache for  
LLM Queries**

**Slash Your LLM API Costs by 10x 💰**  
**Boost Speed by 100x ⚡**



# Architecture



# LLM-powered autonomous agent system

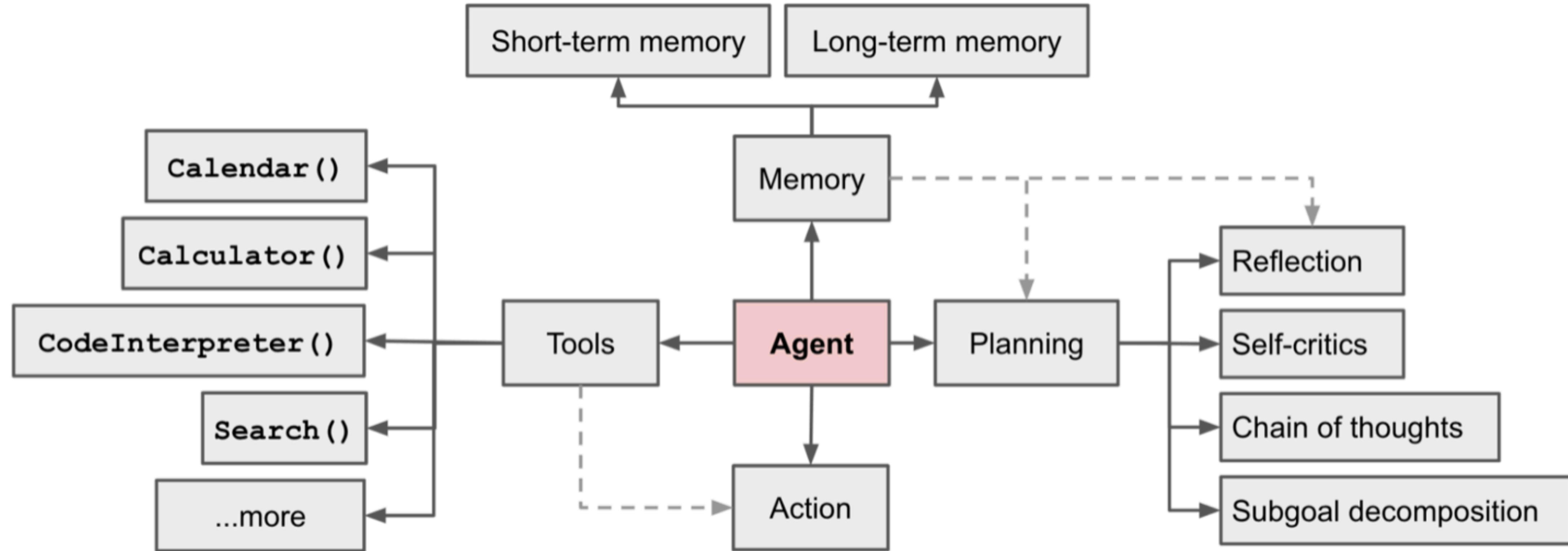


Fig. 1. Overview of a LLM-powered autonomous agent system.

# THANKS FOR WATCHING



Github



公众号



扫码并回复“技术交流”  
加入用户交流群

 <https://github.com/milvus-io/milvus>

 <https://zilliz.com>

