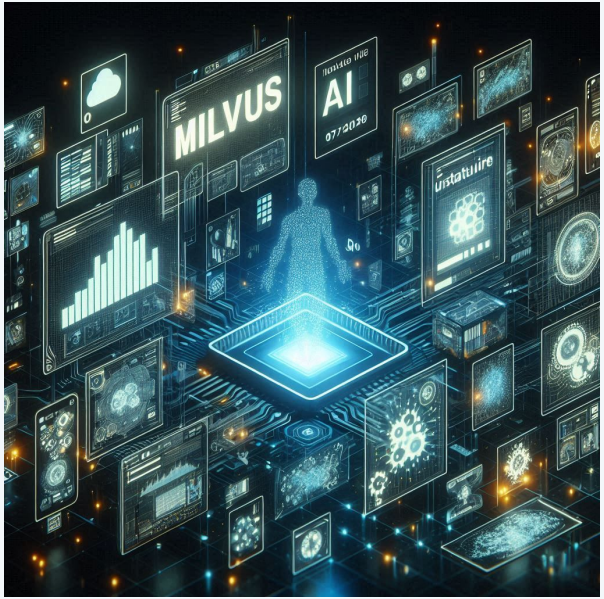
It's in the Air Tonight. Sensor Data in RAG

Tim Spann @ Zilliz

Slides



X

Overview

I will do a quick overview of the basics of Vector Databases and Milvus and then dive into a practical example of how to use one as part of an application. I will demonstrate how to consume air quality data and ingest it into Milvus as vectors and scalars. We will then use our vector database of Air Quality readings to feed our LLM and get proper answers to Air Quality questions. I will show you how to all the steps to build a RAG application with Milvus, LangChain, Ollama, Python and Air Quality Reports. Finally after demos I will answer questions.

AGENDA

Introduction

Overview of Vector Databases

A Quick Introduction to Milvus

Consume and Ingest Air Quality Data

Building a local RAG application

Q&A

zilliz

01 Introduction

Tim Spann

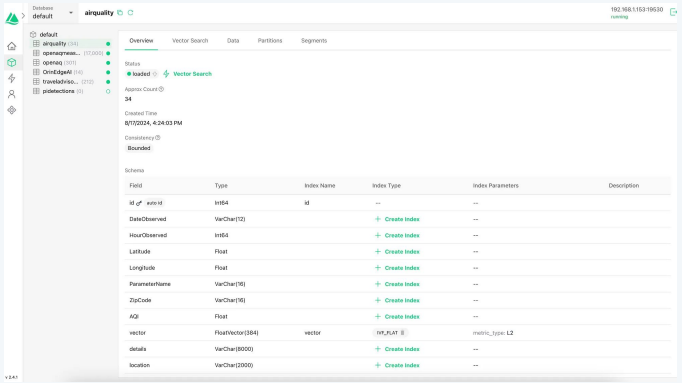
Principal Developer
Advocate, Zilliz

tim.spann@zilliz.com
https://www.linkedin.com/in/timothyspann/
https://x.com/PaaSDev

 |

zilliz

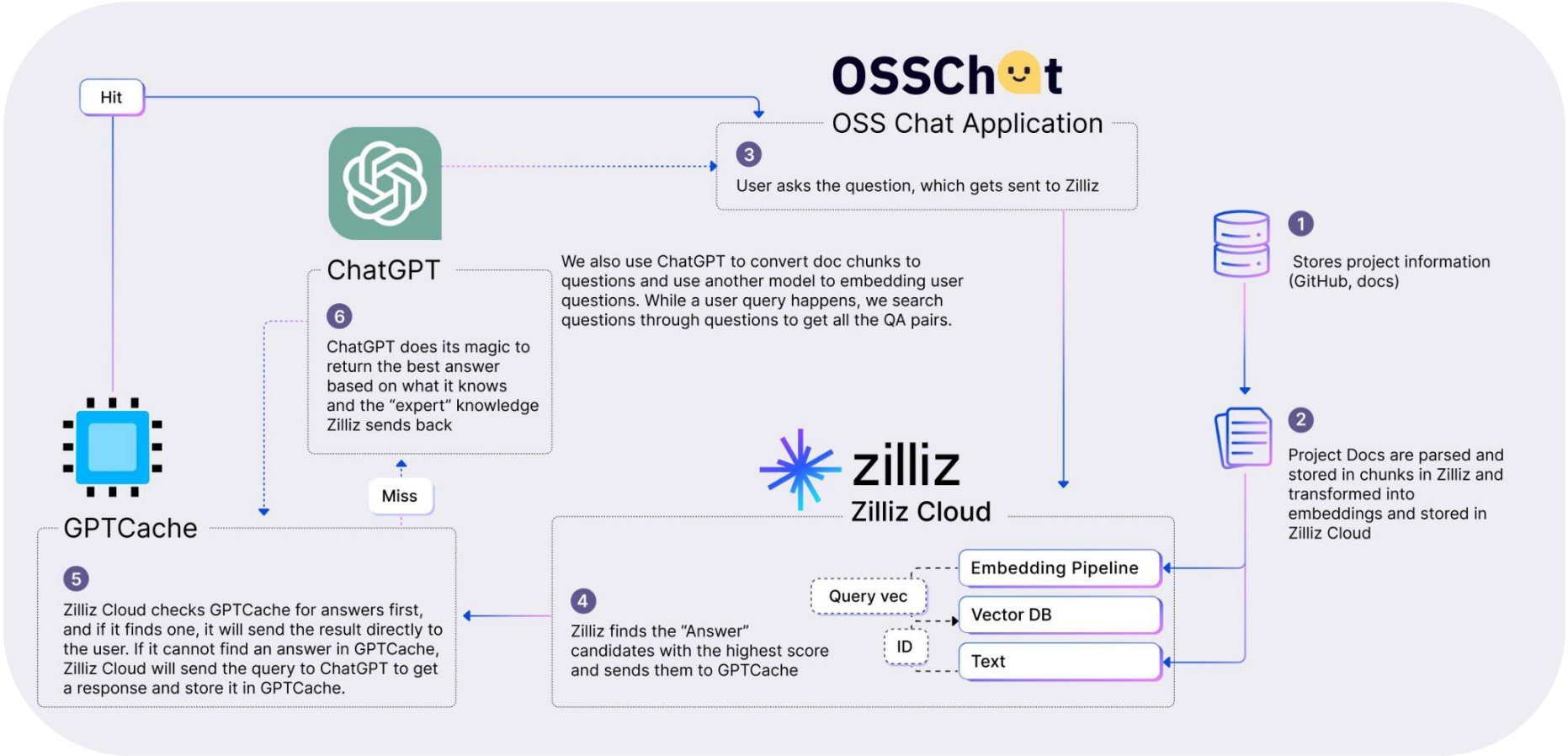
Show Me A Demo



https://multimodal-demo.milvus.io/

https://milvus.io/milvus-demos/reverse-image-search

zilliz



osschat.io

zilliz

Unstructured Data is Everywhere

Unstructured data is any data that does not conform to a predefined data model.

Currently, 90% of unstructured data is never analyzed.

Text

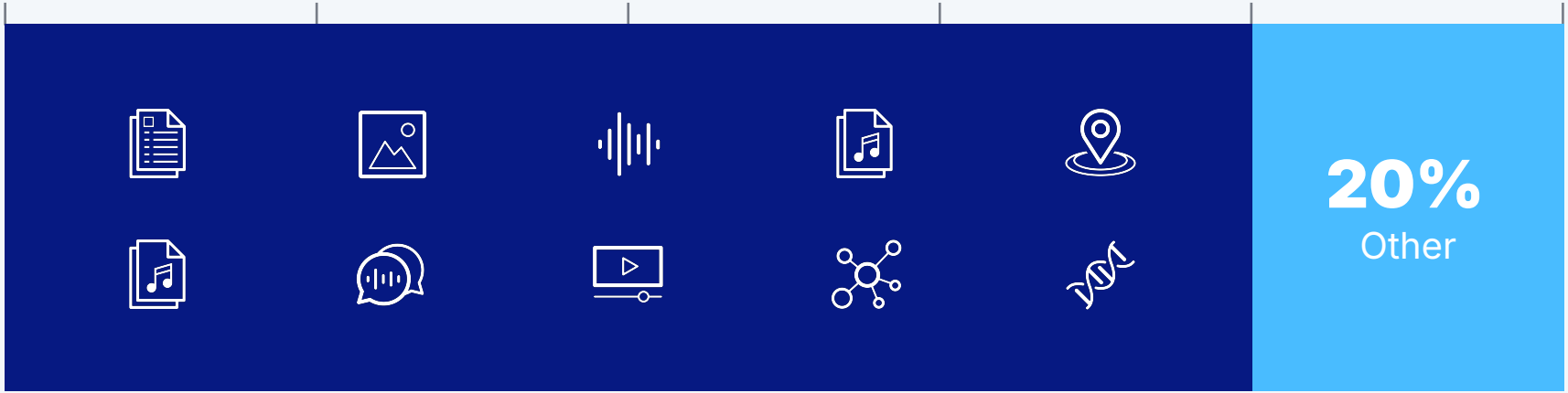
Images

Videos

and more!

…and cannot process increasingly growing unstructured data

80% newly generated data in 2025 will be unstructured data



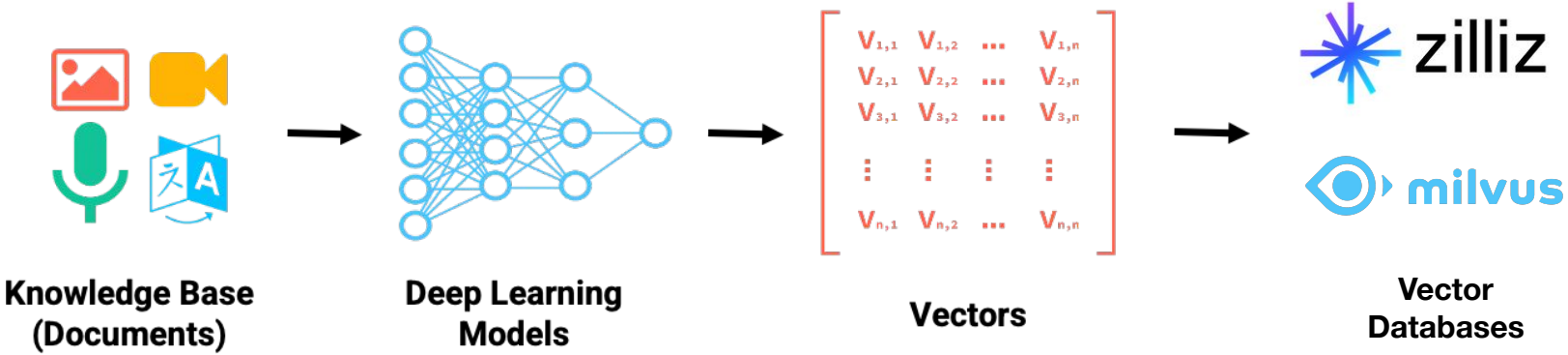
*Data Source: The Digitization of the World by *IDC*

 |

zilliz

The challenge of unstructured data

- **Problem:** Unstructured data comes in lots of forms, no easy way to interact with it all
- **Solution:** Vector embeddings
- **How:** Neural networks e.g. embedding models



02

Overview of Vector Databases

zilliz

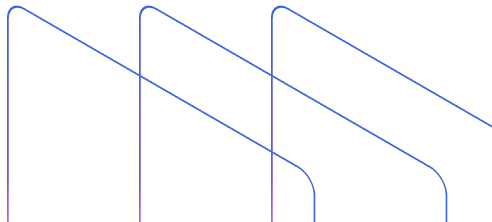
Why a Vector Database?

Purpose-built to store, index and query vector embeddings from unstructured data.

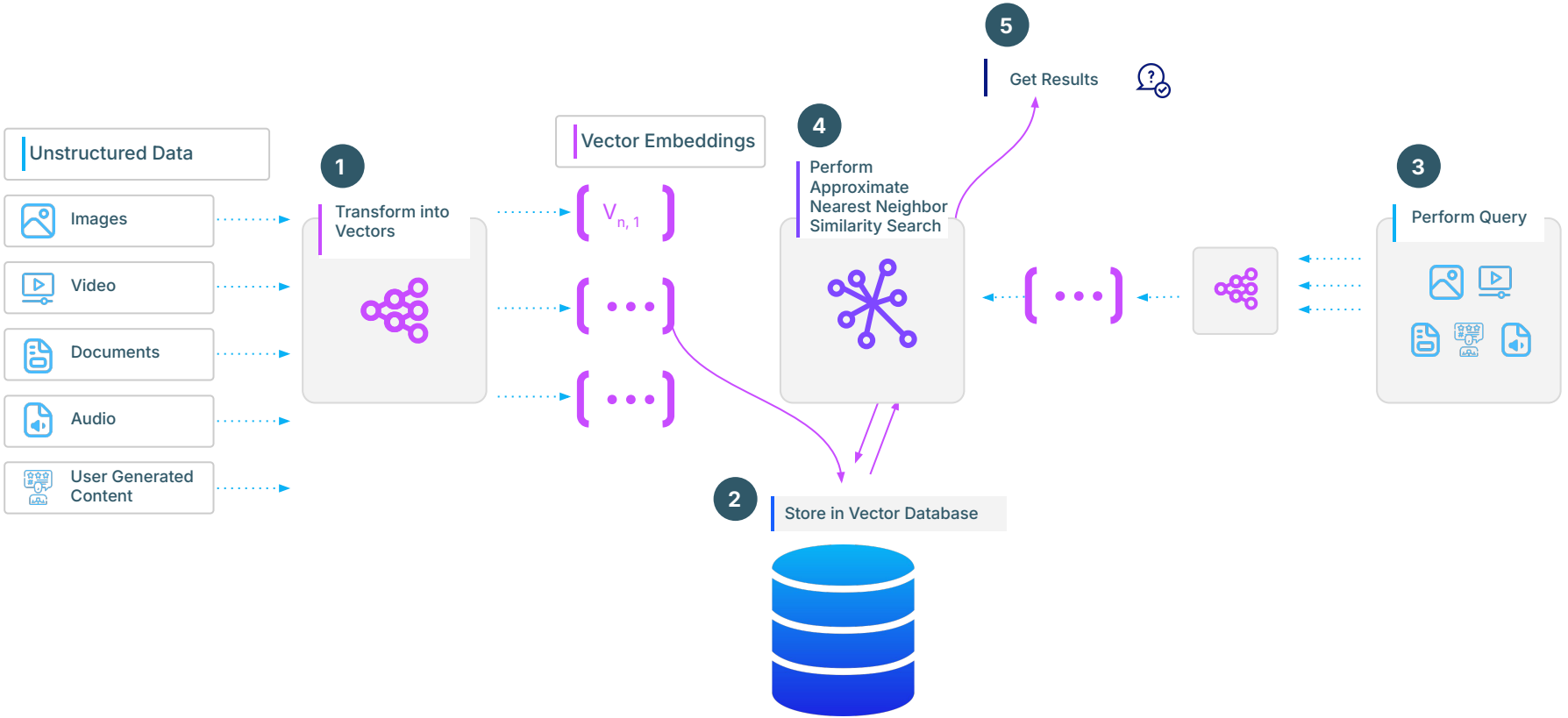
- Vector database
  - Advanced filtering (filtered vector search, chained filters)
  - Hybrid search (e.g. full text + dense vector)
  - Durability (any write in a db is durable, a library typically only supports snapshotting)
  - Replication / High Availability
  - Sharding
  - Aggregations or faceted search
  - Backups
  - Lifecycle management (CRUD, Batch delete, dropping whole indexes, reindexing)
  - Multi-tenancy

- Vector search library
  - High-performance vector search

- How do I support different applications?
  - High query load
  - High insertion/deletion
  - Full precision/recall
  - Accelerator support (GPU, FPGA)
  - Billion-scale storage

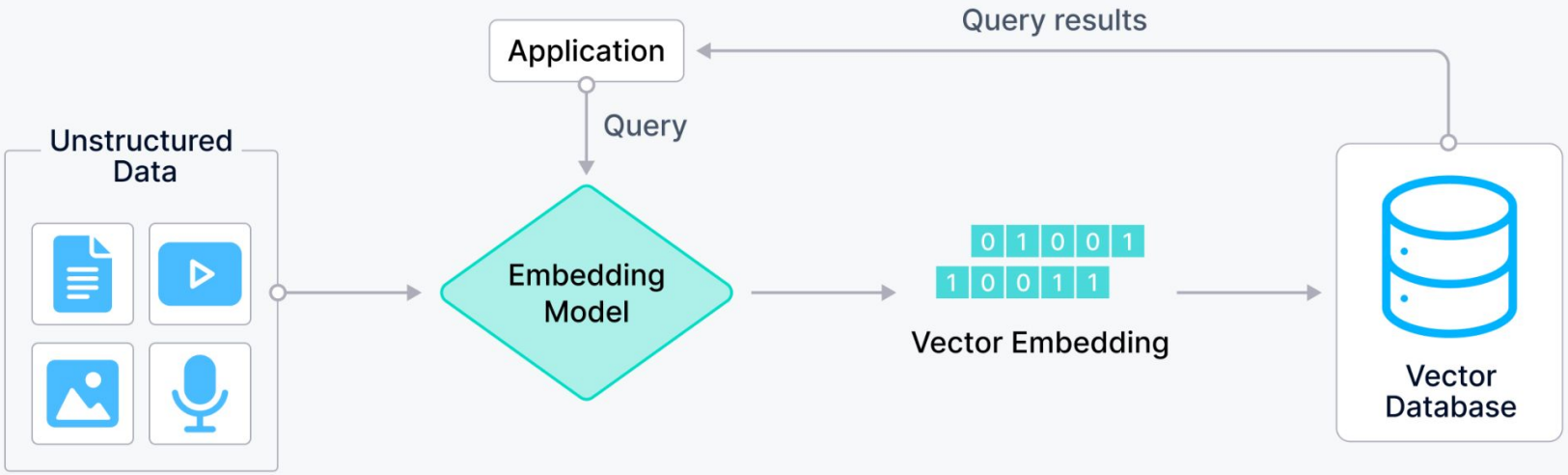


How Similarity Search Works



Vector Database: Making Sense of Unstructured Data

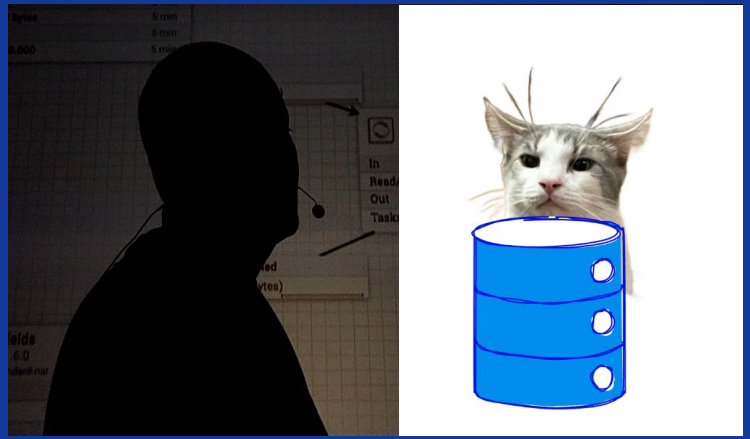
A vector database stores embedding vectors and allows for semantic retrieval of various types of unstructured data.



03

A Quick Introduction to Milvus

Milvus Features



Scalable and Elastic Architecture

Diverse Index Support

Versatile Search Capabilities

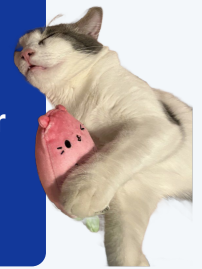
Tunable Consistency

Multi-Tenancy

Hardware-Accelerated Compute Support

Python, Java, Golang, NodeJS

Milvus Lite, K8, Zilliz Cloud, Docker



 |

Technologies for various types of Use cases

Index Types

Offer a wide range of **15 indexes** support, including popular ones like Hierarchical Navigable Small Worlds (HNSW), PQ, Binary, Sparse, DiskANN and GPU index

Empower developers with tailored search optimizations, catering to performance, accuracy and cost needs

Search Types

Support multiple types such as **top-K ANN, Range ANN, sparse & dense, multi-vector, grouping,** and metadata **filtering**

Enable query flexibility and accuracy, allowing developers to tailor their information retrieval needs

Multi-tenancy

Enable **multi-tenancy** through collection and partition management

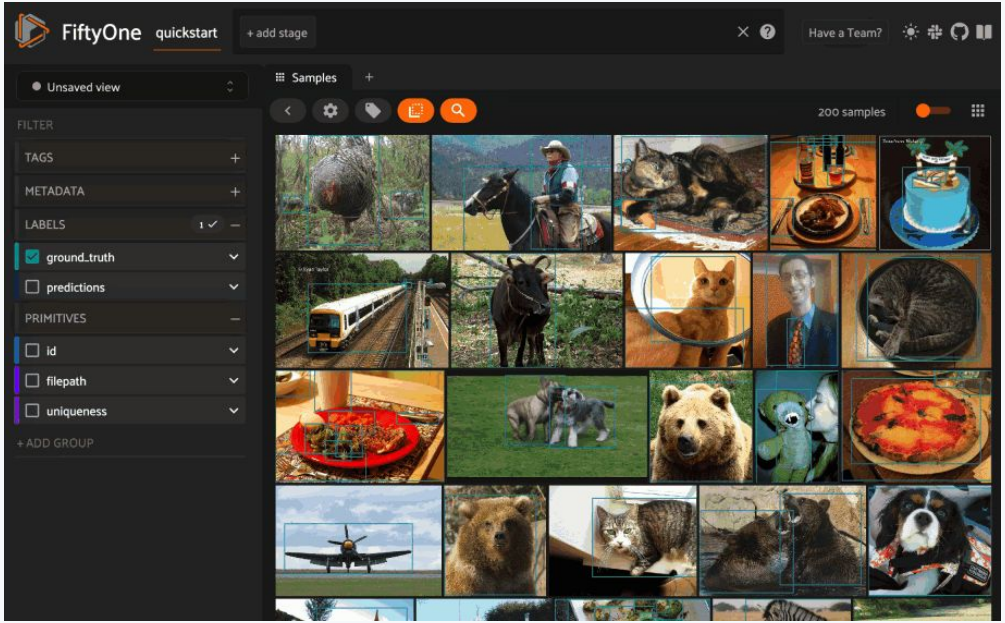
Allow for efficient resource utilization and customizable data segregation, ensuring secure and isolated data handling for each tenant

Compute Types

Designed for various compute powers, such as **AVX512, Neon for SIMD, quantization cache-aware optimization** and **GPU**

Leverage strengths of each hardware type, ensuring high-speed processing and cost-effective scalability for different application needs

FiveOne + Milvus



https://docs.voxel51.com/integrations/milvus.html

https://milvus.io/docs/integrate_with_voxel51.md

04

Consume and Ingest Air Quality Data

zilliz

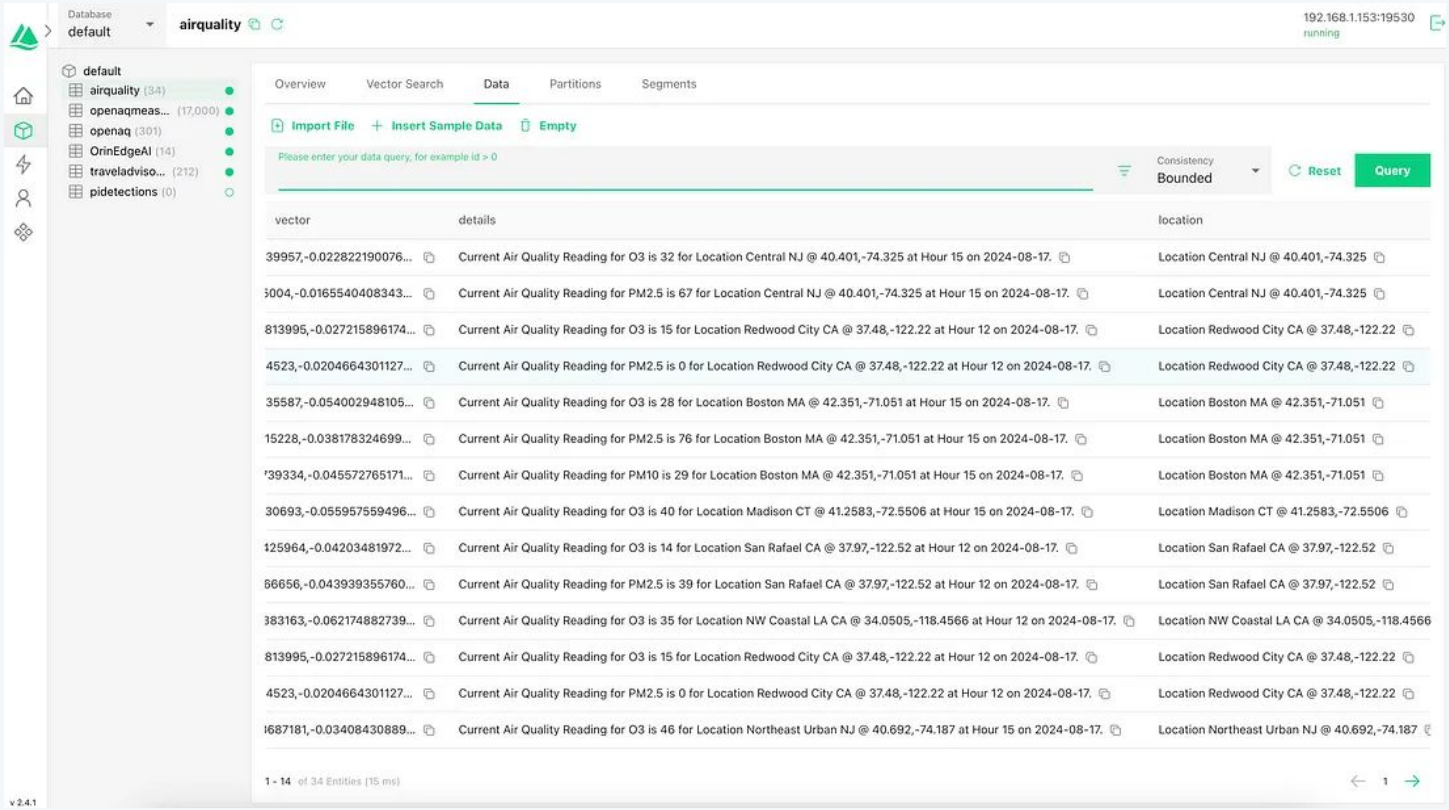
DATA!!!!

zilliz

REST JSON Ingest

```
    url =  'https://api.openaq.org/v2/measurements?country=US&date_from={0}&date_to={1}&limit=1000&page={2}&offset=0&sort=desc&radius=1000&order_by=datetime'.format(
        str(YESTERDAYS_DATE), str(TODAYS_DATE), str(aqpage) )
    headers = {"accept": "application/json", "x-api-key": str(API_KEY)}

    response = requests.get(url, headers=headers)

    if ( len(response.text) > 0 ):
        openaq2 = json.loads(response.text)
    else:
        openaq2['results'] = {}

    try:
        for jsonitems in openaq2['results']:
            count = count + 1
            fulllocation = 'Location {0}: {1}, {2}, {3} @ {4},{5}'.format(jsonitems.get('locationId'),
                                                                          jsonitems.get('location'),jsonitems.get('city'),jsonitems.get('country'),
                                                                          jsonitems.get('coordinates')['latitude'],jsonitems.get('coordinates')['longitude'] )
            details = 'Current Air Quality Reading for {0} is {1} {2} for {3} at {4}. Is Mobile: {5} Is Analysis: {6} Entity: {7} Sensor Type: {8}'.format(jsonitems.get('parameter'), jsonitems.get('value'),
jsonitems.get('unit'),fulllocation, jsonitems.get('date')['local'],
            jsonitems.get('isMobile'), jsonitems.get('isAnalysis'), jsonitems.get('entity'), jsonitems.get('sensorType'))

            data.append({ "locationId": int(jsonitems.get('locationId')),
                          "location": str(jsonitems.get('location','')),
                          "parameter": str(jsonitems.get('parameter','')),
                          "value": float(jsonitems.get('value')),
                          "datelocal": str(jsonitems.get('date')['local']),
                          "unit": str(jsonitems.get('unit','')),
                          "latitude": float(jsonitems.get('coordinates')['latitude']),
                          "longitude": float(jsonitems.get('coordinates')['longitude']),
                          "country": str(jsonitems.get('country','')),
                          "city": str(jsonitems.get('city','')),
                          "isMobile": str(jsonitems.get('isMobile','')),
                          "isAnalysis": str(jsonitems.get('isAnalysis','')),
                          "entity": str(jsonitems.get('entity','')),
                          "sensorType": str(jsonitems.get('sensorType','')),
                          "vector": model(details), "details": str(details), "fulllocation": str(fulllocation)})

        res = milvus_client.insert(collection_name=OAQ3_COLLECTION_NAME, data=data)
        print(count)
    except Exception as ex:
```

Scalars and Vectors in Milvus



zilliz

05

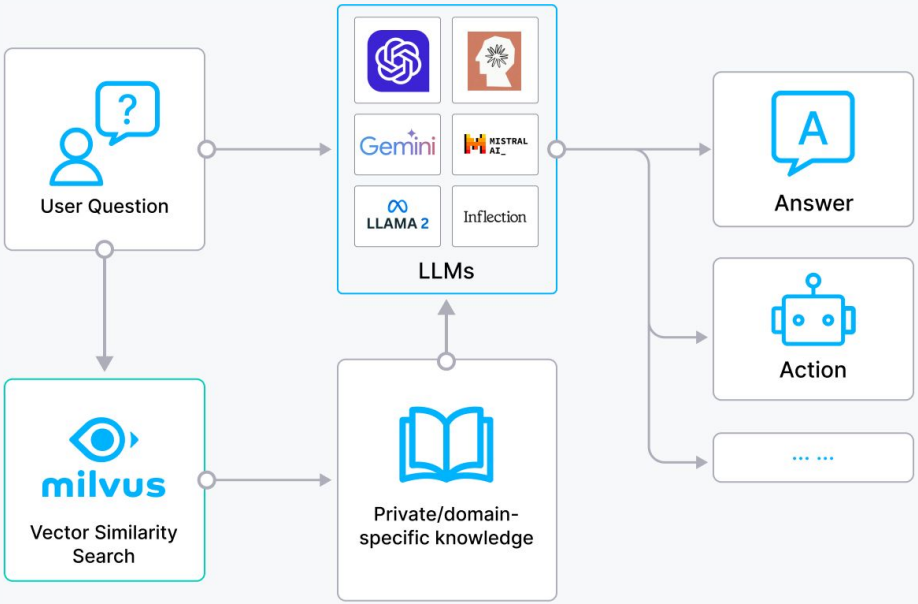
Building a local RAG application

zilliz

Retrieval-Augmented Generation (RAG)

A technique that combines the strength of retrieval-based and generative models:

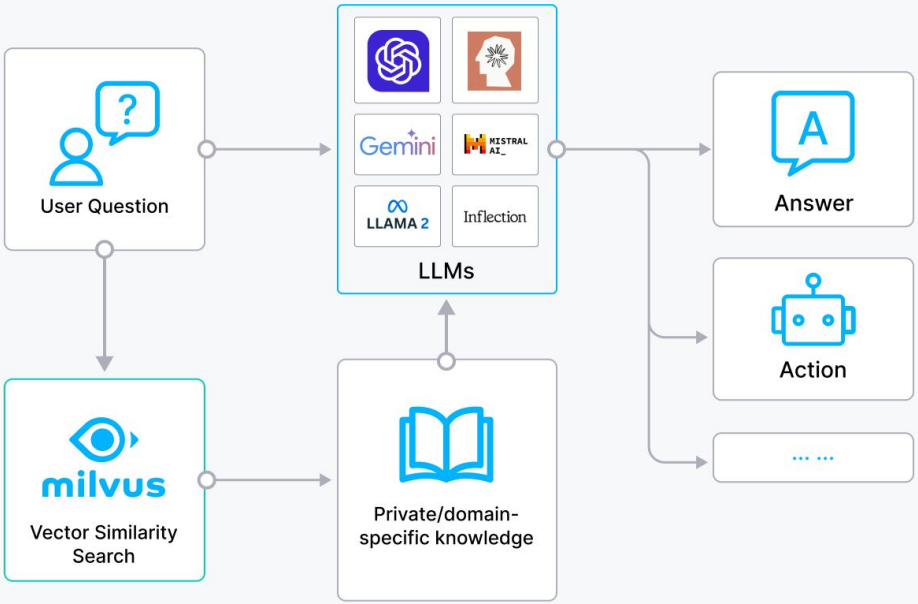
- Improve accuracy and relevance
- Eliminate hallucination
- Provide domain-specific knowledge



RAG : an economic perspective

A business model that bridges public data and private data

- Data sovereignty
- You can't and shouldn't give your private data to others



Ollama + Llama 3.1 + Milvus + LangChain = RAG

```
# ---------------------------------------------------------------------------
# Connect to Milvus
# https://zilliz.com/blog/a-beginners-guide-to-using-llama-3-with-ollama-milvus-langchain
# https://github.com/stephen37/ollama_local_rag/blob/main/rag_berlin_parliament.py
# https://python.langchain.com/v0.2/docs/integrations/vectorstores/milvus/
# https://api.python.langchain.com/en/latest/vectorstores/langchain_milvus.vectorstores.milvus.Milvus.html

vector_store = Milvus(
    embedding_function=embeddings,
    collection_name=OAQ3_COLLECTION_NAME,
    primary_field = "id",
    text_field="details",
    connection_args={"uri": MILVUS_URL},
)
```

```
def run_query() -> None:
    llm = Ollama(
        model="llama3.1",
        callback_manager=CallbackManager([StreamingStdOutCallbackHandler()]),
        stop=["<|eot_id|>"],
    )

    query = input("\nQuery: ")
    qa_chain = RetrievalQA.from_chain_type(
        llm, retriever=vector_store.as_retriever(collection =  OAQ3_COLLECTION_NAME, expr="parameter == 'o3'"))

    result = qa_chain.invoke({"query": query})

    resultforslack = str(result["result"])
    print(resultforslack)

    try:
        response = client.chat_postMessage(mrkdwn=True, channel="C06NE1FU6SE", text="",
                                           blocks=[{"type": "section","text": {"type": "mrkdwn","text": "*" + str(query) + "*  \n\n" + str(resultforslack) +"\n" }}])


    except SlackApiError as e:
        # You will get a SlackApiError if "ok" is False
        print("Slack failed")
```

```
    while True:
        run_query()
```

 |

zilliz

Ollama + Llama 3.1 + Milvus + LangChain = RAG

```
Query:  Provide a detailed air quality report for Edison
Based on the provided context, here is a detailed air quality report for Edison:

**Location:** Edison, US (Latitude: 35.345607, Longitude: -118.851825)

**Current Air Quality Reading for O3:**

* **Highest Value:** 0.092 ppm recorded at 2024-08-30T15:00:00-07:00
* **Lowest Value:** 0.052 ppm recorded at 2024-08-29T19:00:00-07:00
* **Current Value:** 0.062 ppm recorded at 2024-08-30T11:00:00-07:00

**Trend Analysis:**

* The current value is slightly higher than the lowest value recorded, indicating a slight increase in ozone levels.
* However, it's still within a relatively safe range and not exceeding any health or safety thresholds.

**Sensor Information:**

* **Sensor Type:** Reference grade
* **Entity:** Governmental Organization

**Mobility:** The sensor is stationary (Mobile: False), indicating that the readings are taken from a fixed location.Based on the provided context, here is a detailed air quality report for Edison:

**Location:** Edison, US (Latitude: 35.345607, Longitude: -118.851825)

**Current Air Quality Reading for O3:**

* **Highest Value:** 0.092 ppm recorded at 2024-08-30T15:00:00-07:00
* **Lowest Value:** 0.052 ppm recorded at 2024-08-29T19:00:00-07:00
* **Current Value:** 0.062 ppm recorded at 2024-08-30T11:00:00-07:00

**Trend Analysis:**

* The current value is slightly higher than the lowest value recorded, indicating a slight increase in ozone levels.
* However, it's still within a relatively safe range and not exceeding any health or safety thresholds.

**Sensor Information:**

* **Sensor Type:** Reference grade
* **Entity:** Governmental Organization

**Mobility:** The sensor is stationary (Mobile: False), indicating that the readings are taken from a fixed location.
```


zilliz

Embeddings Models

zilliz

06

Q & A

zilliz

https://medium.com/@tspann/whats-in-the-air-tonight-mr-milvus-fbd42f06e482

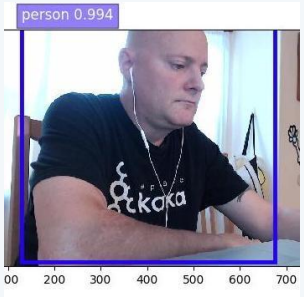
RESOURCES

 |

zilliz

Vector Database Resources

Give Milvus a Star!

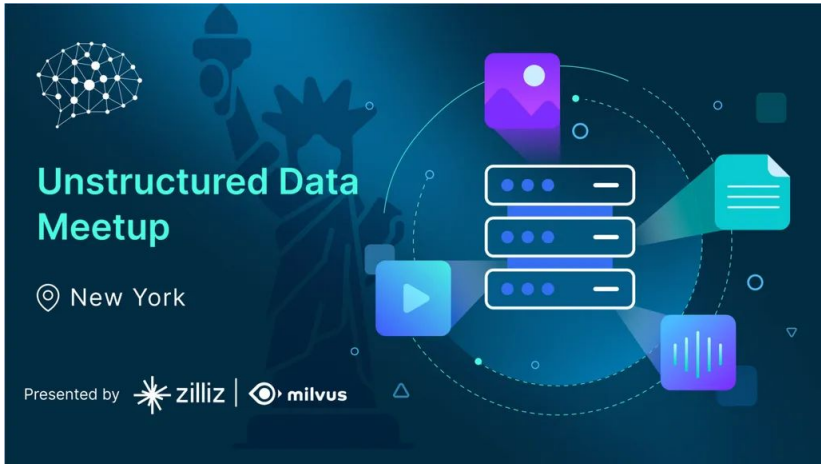


Chat with me on Discord!

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

zilliz

Unstructured Data Meetup



https://www.meetup.com/unstructured-data-meetup-new-york/

This meetup is for people working in unstructured data. Speakers will come present about related topics such as vector databases, LLMs, and managing data at scale. The intended audience of this group includes roles like machine learning engineers, data scientists, data engineers, software engineers, and PMs.

This meetup was formerly Milvus Meetup, and is sponsored by Zilliz maintainers of Milvus.

Generative AI Resource Hub

Tutorials, Code Examples, and Best Practices for Developing and Deploying GenAI Applications.

https://zilliz.com/learn/generative-ai

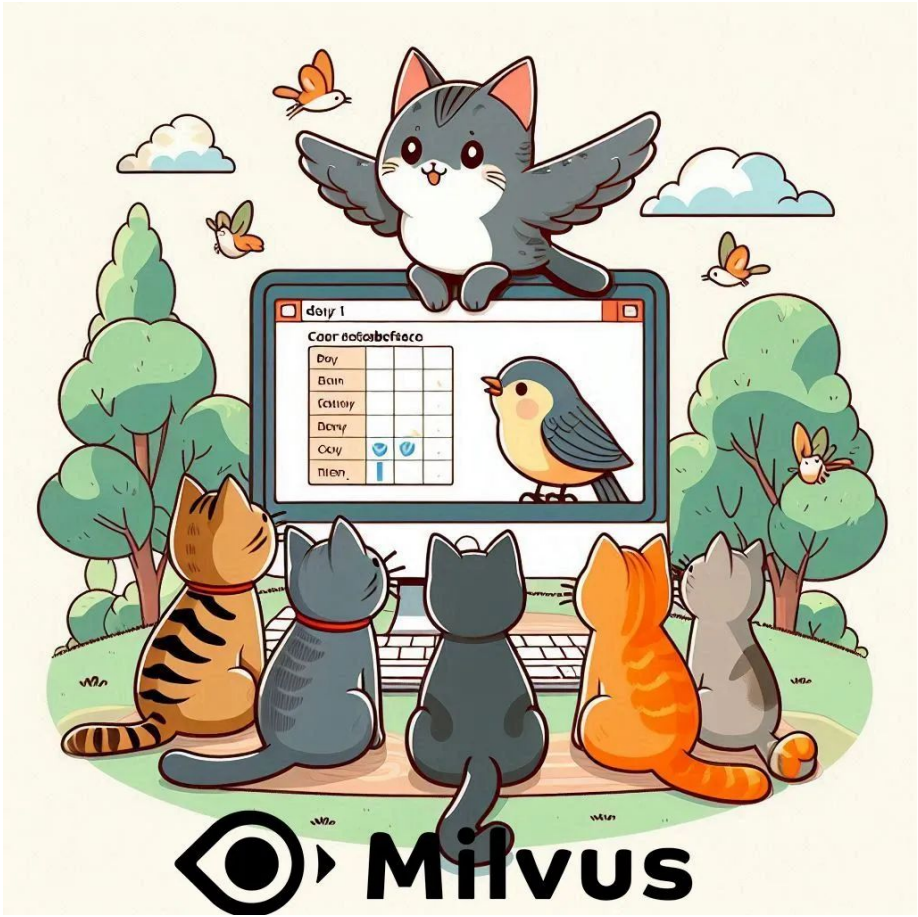
 |

zilliz

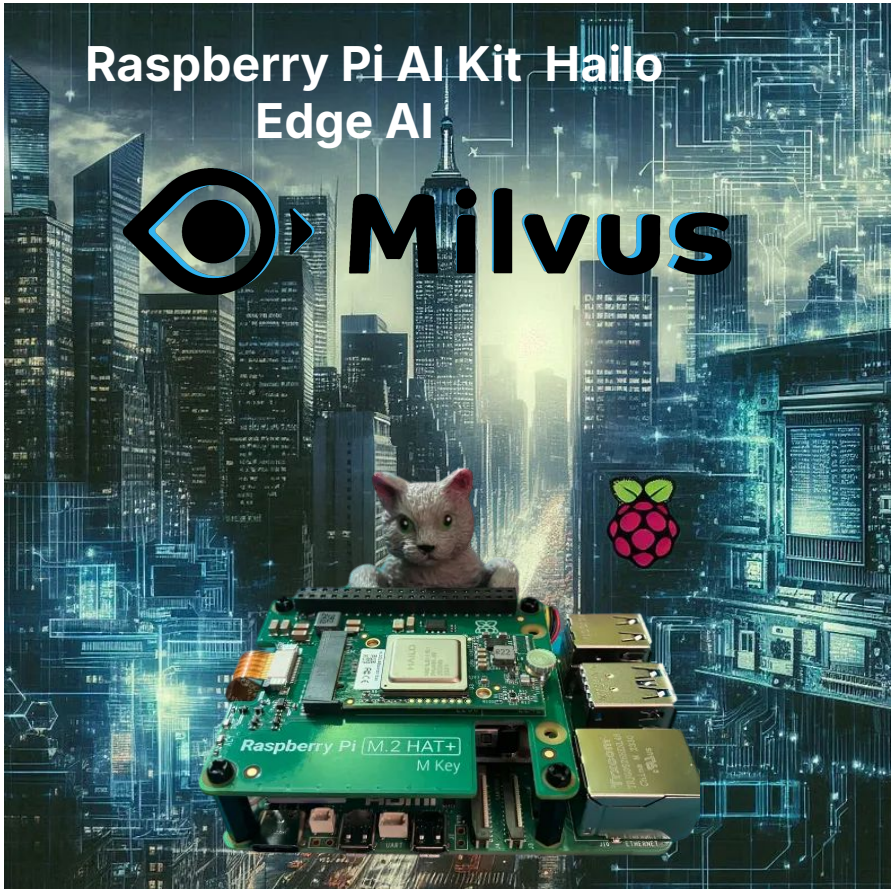

Learning Vector

https://medium.com/@tspann/unstructured-street-data-in-new-york-8d3cde0a1e5b

https://medium.com/@tspann/not-every-field-is-just-text-numbers-or-vectors-976231e90e4d



https://medium.com/@tspann/shining-some-light-on-the-new-milvus-lite-5a0565eb5dd9



https://medium.com/@tspann/unstructured-data-processing-with-a-raspberry-pi-ai-kit-c959dd7fff47

AIM Weekly by Tim Spann

https://bit.ly/32dAJft
https://github.com/milvus-io/milvus

This week in Milvus, Towhee, Attu, GPT Cache, Gen AI, LLM, Apache NiFi, Apache Flink, Apache Kafka, ML, AI, Apache Spark, Apache Iceberg, Python, Java, Vector DB and Open Source friends.

 |

zilliz

Thank you!

milvus.io

github.com/milvus-io/

@milvusio

Connect with me!

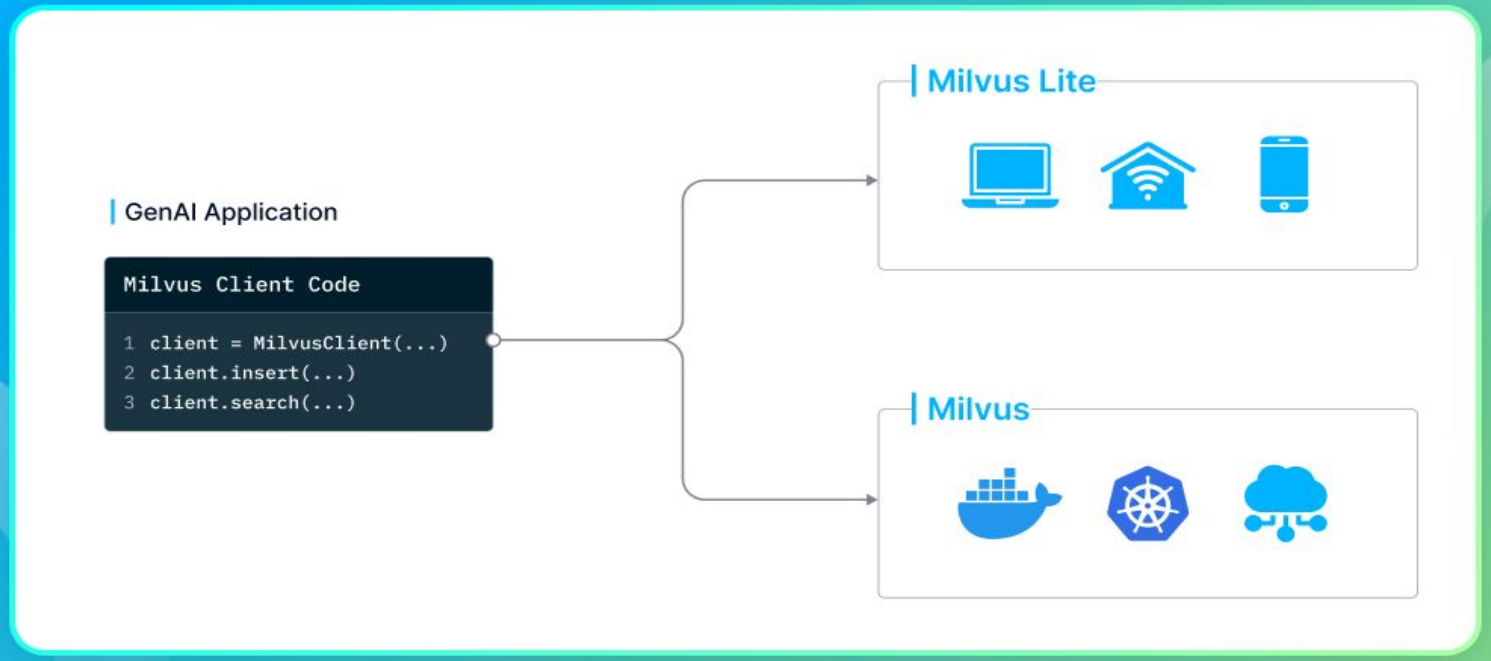
@paasDev

/in/timothyspann

 |

zilliz

Build Once
Deploy Anywhere



milvus

zilliz

BEFORE MILVUS

 |

zilliz

Join us at our next meetup!
meetup.com/unstructured-data-meetup-new-york/

THANK YOU