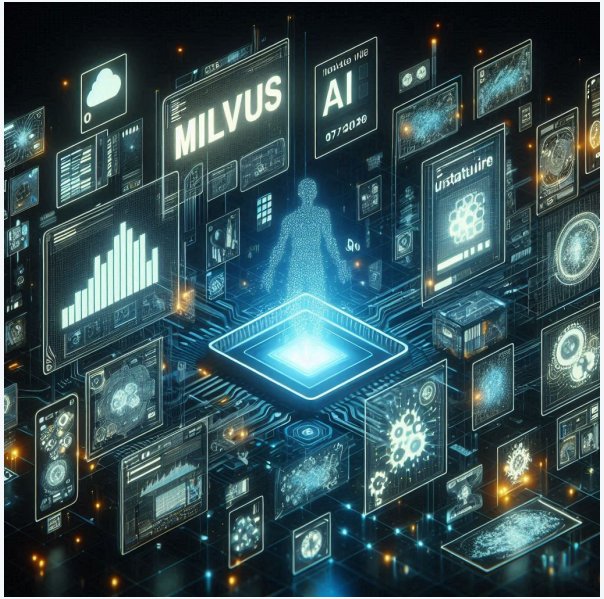
Hybrid Search

Tim Spann @ Zilliz

zilliz

milvus

Slides


MILVUS
AI


AICamp | NEW YORK
IN PERSON MEETUP
AI, LLMs, ML and
Data Meetup

X

Tim Spann

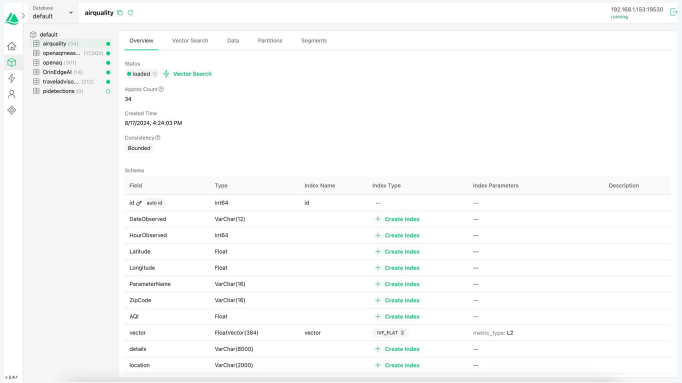
Principal Developer
Advocate, Zilliz

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

 | © Copyright 2024 Zilliz

zilliz

Show Me A Demo



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

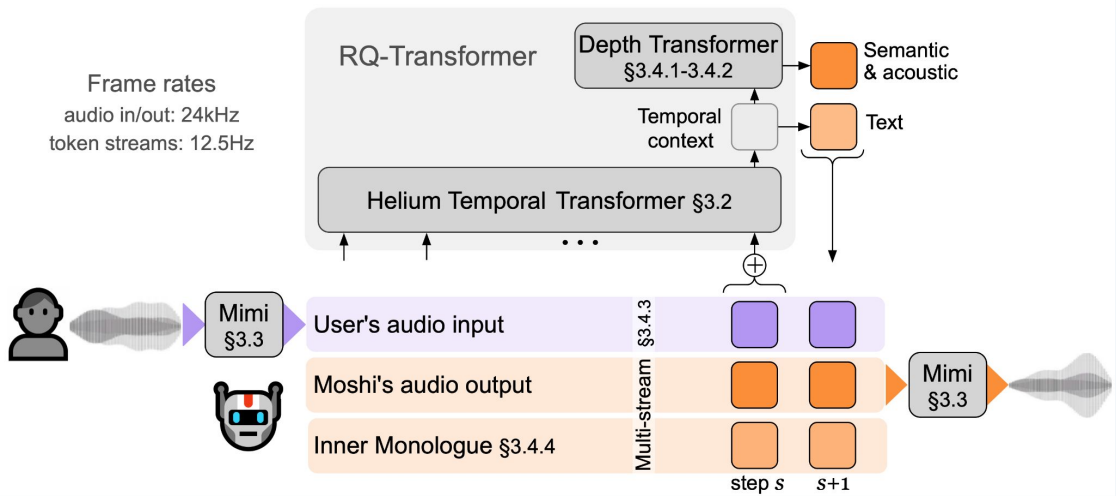
https://milvus.io/milvus-demos/reverse-image-search

 | © Copyright Zilliz

zilliz

What's New?

- https://huggingface.co/mistralai/Mistral-Small-Instruct-2409
- https://github.com/kyutai-labs/moshi



 © Copyright Zilliz

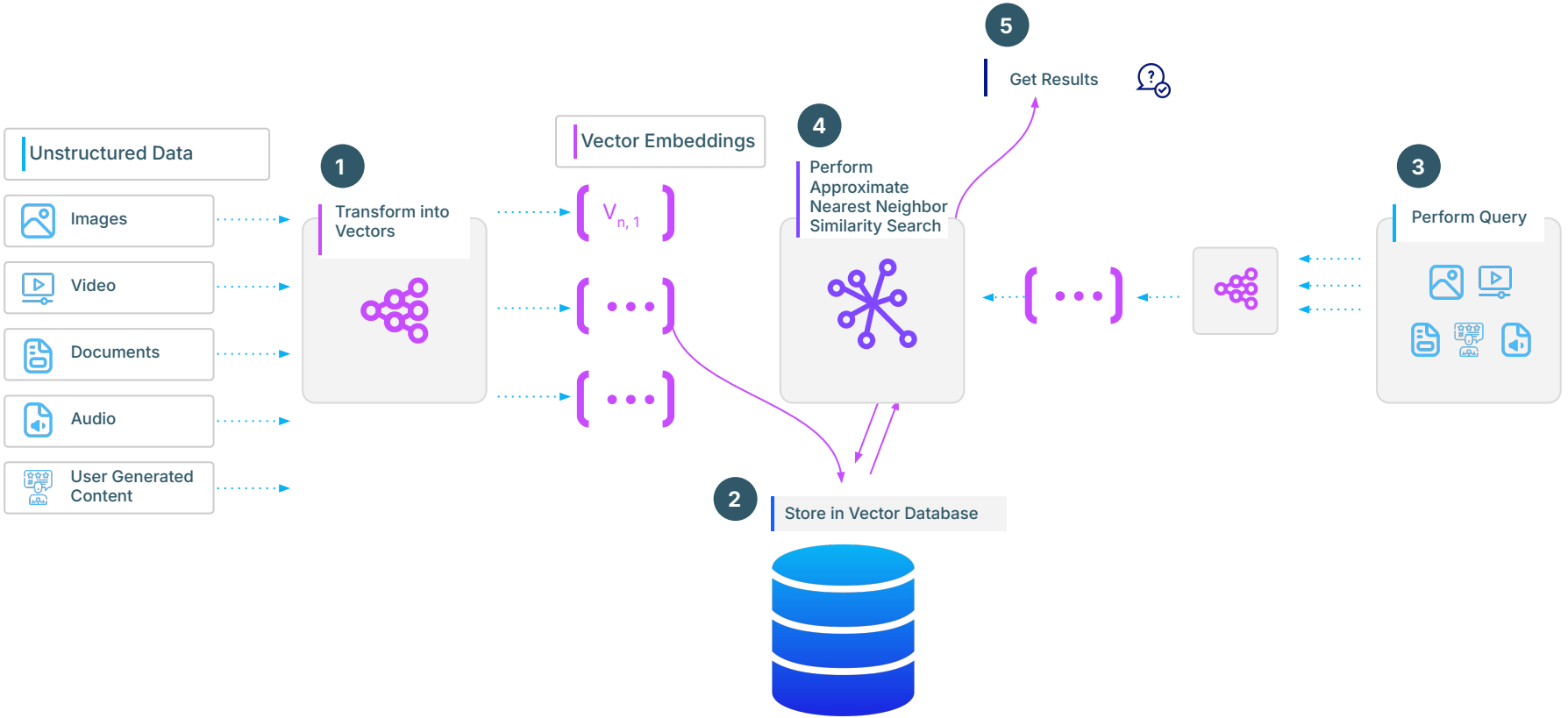
zilliz

02

Overview of Vector Databases

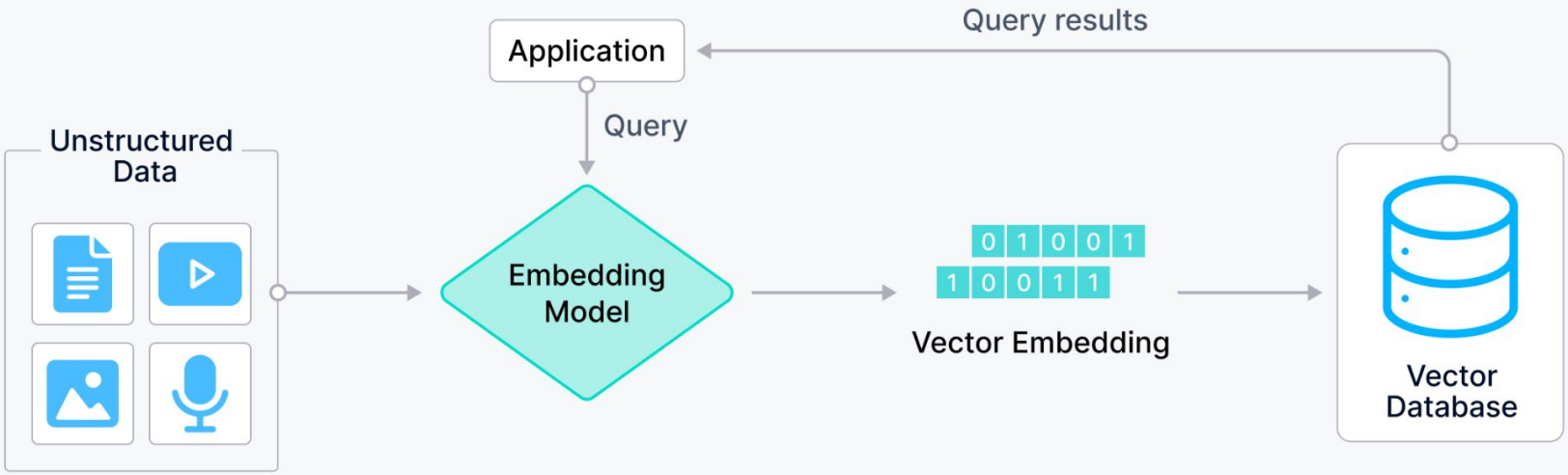
zilliz

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.



zilliz

Do you really need a Vector Database?

ANN Libraries

- FAISS, ANNOY, HNSW
- Supports 1M vectors
- Good for prototyping

Existing Solutions

- 50M~100M vectors
- PostgreSQL, ElasticSearch, Big Query, MongoDB, etc with ANNS plug-ins

Vector Databases

- Purpose-built for vectors top support the requirements and lifecycle of vectors
- Billion+ scale
- CRUD, real-time search, top-k/range/hybrid search, multi-modal, multi-vector query, distributed
- Semantic Search is core to your business

Vector Databases are **purpose-built** to handle **indexing, storing, and querying** vector data.

 | © Copyright Zilliz

zilliz

Vector Search + Indexing + Filtering = 

 | © Copyright 2024 Zilliz

zilliz

zilliz

03

A Quick Introduction to Milvus

milvus

LF AI & DATA

Easy Setup

Pip-install to start coding in a notebook within seconds.

Reusable Code

Write once, and deploy with one line of code into the production environment

Integration

Plug into OpenAI, Langchain, LlmaIndex, and many more

Feature-rich

Dense & sparse embeddings, filtering, reranking and beyond

Milvus is **an open-source vector database** for **GenAI** projects. `pip install` on your laptop, plug into popular AI dev tools, and push to production with a single line of code.

29.4K+
GitHub Stars

2,800+
Forks

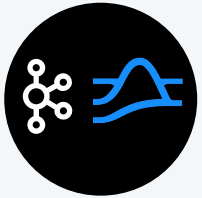
25M+
Downloads

250+
Contributors

Milvus 🤝 Open-Source

MINIO

Store Vectors and Indexes
Enables Milvus' stateless architecture



Kafka/ Pulsar

Handles Data Insertion stream
Internal Component Communications
Real-time updates to Milvus

Prometheus / Grafana

Collects metrics from Milvus
Provides real-time monitoring dashboards

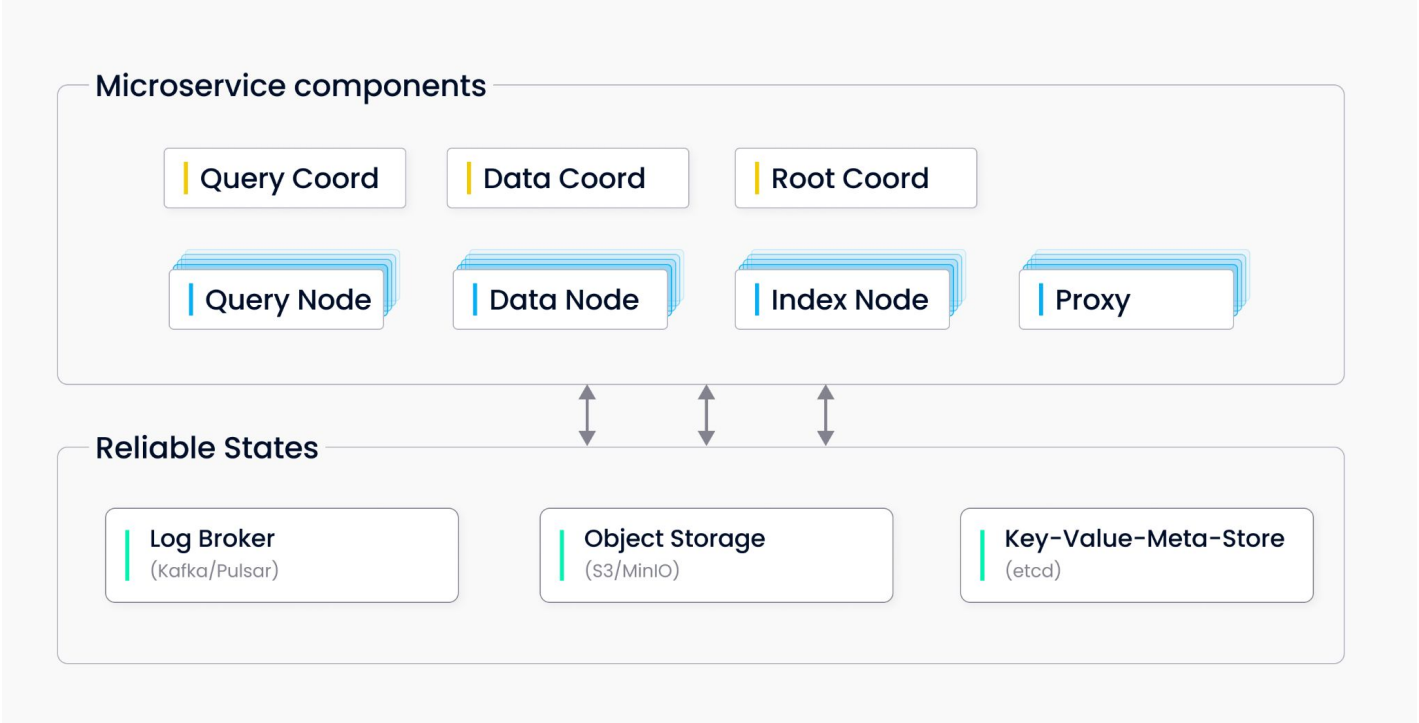
Kubernetes

Milvus Operator CRDs

 | © Copyright Zilliz

zilliz

Distributed Architecture



zilliz

Dynamic Scaling 🚀

Stateless components for **Easy Scaling**

- **Query**, **Index**, and **Data Nodes** can be scaled **independently**
- Allows for **optimized resource allocation** based on workload characteristics

Data sharding across multiple nodes

- **Distributes** large **datasets** across multiple **Data Nodes**
- Enables **parallel processing** for improved query performance

Horizontal Pod Autoscaler (**HPA**)

- **Automatically scales** up and down
- **Custom metrics** can be used (e.g., query latency, throughput)

 | © Copyright Zilliz

zilliz

Stateless Architecture

Stateless Components

All Milvus components are deployed **Stateless**.

Object Storage

Milvus relies on **Object Storage** (MinIO, S3, etc) for data **persistence**.
Vectors are stored in **Object Storage**, **Metadata** is in **etcd.**

Scaling and Failover

Scaling and failover **don't** involve traditional **data rebalancing.** When **new pods** are added or existing ones fail, they can immediately start handling requests by **accessing data** from the **shared object storage.**

 © Copyright Zilliz

zilliz

Different Consistency levels

Ensures every node or replica has the same view of data at a given time.

- **Strong**: Guaranteed up-to-date reads, highest latency
- **Bounded**: Reads may be slightly stale, but within a time bound
- **Session**: Consistent reads within a session, may be stale across sessions
- **Eventually**: Lowest latency, reads may be stale

Trade Offs

- **Strong** consistency for **critical applications** requiring accurate results
- **Eventually** consistency for **high-throughput**, latency-sensitive apps

 © Copyright Zilliz

zilliz

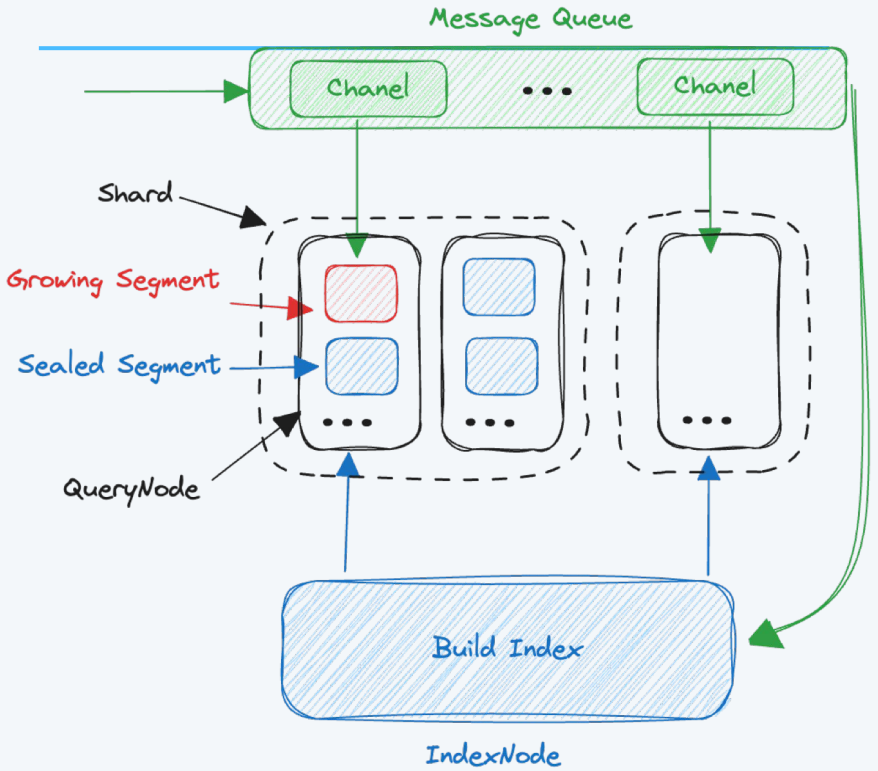
Milvus Data Layout - **Segments**

Growing Segment:

- In-memory segment replaying data from the Log Broker.
- Uses a FLAT index to ensure data is fresh and appendable.

Sealed Segment:

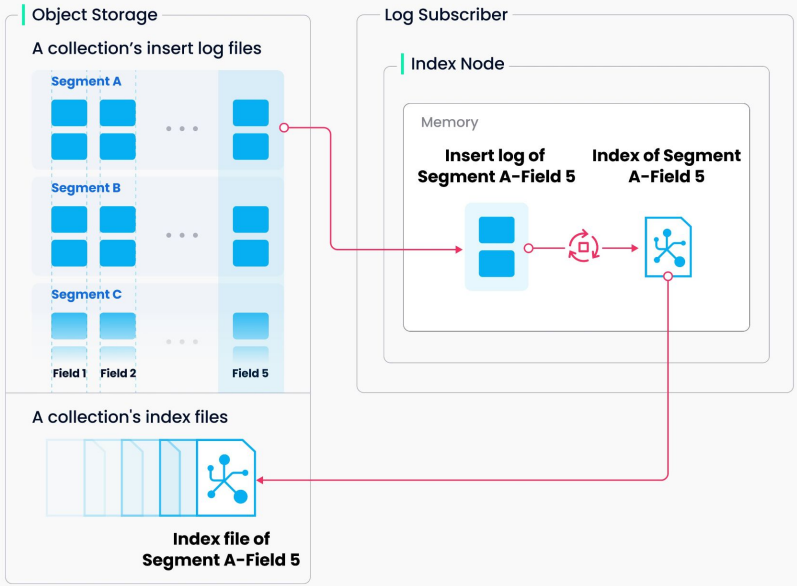
- Immutable segment using alternative indexing methods for efficiency.



 | © Copyright Zilliz

zilliz

Index Building



To **avoid frequent index building** for data updates.

A **collection** in Milvus is **divided** further **into segments**, each with its own index.

 | © Copyright Zilliz

zilliz

Picking an Index

- 100% Recall – Use `FLAT` search if you need 100% accuracy
- 10MB < `index_size` < 2GB – Standard IVF
- 2GB < `index_size` < 20GB – Consider `PQ` and `HNSW`
- 20GB < `index_size` < 200GB – Composite Index, `IVF_PQ` or `HNSW_SQ`
- Disk-based indexes

 | © Copyright Zilliz

zilliz

Indexes

Most of the vector index types supported by Milvus use approximate nearest neighbors search (ANNS),

- **HNSW**: HNSW is a graph-based index and is best suited for scenarios that have a high demand for search efficiency. There is also a GPU version **GPU_CAGRA**, thanks to Nvidia's contribution.
- **FLAT**: FLAT is best suited for scenarios that seek perfectly accurate and exact search results on a small, million-scale dataset. There is also a GPU version **GPU_BRUTE_FORCE**.
- **IVF_FLAT**: IVF_FLAT is a quantization-based index and is best suited for scenarios that seek an ideal balance between accuracy and query speed. There is also a GPU version **GPU_IVF_FLAT**.
- **IVF_SQ8**: IVF_SQ8 is a quantization-based index and is best suited for scenarios that seek a significant reduction on disk, CPU, and GPU memory consumption as these resources are very limited.
- **IVF_PQ**: IVF_PQ is a quantization-based index and is best suited for scenarios that seek high query speed even at the cost of accuracy. There is also a GPU version **GPU_IVF_PQ**.

© Zilliz 2024 All rights reserved.

Indexes Continued.

- **SCANN**: SCANN is similar to IVF_PQ in terms of vector clustering and product quantization. What makes them different lies in the implementation details of product quantization and the use of SIMD (Single-Instruction / Multi-data) for efficient calculation.
- **DiskANN**: Based on Vamana graphs, DiskANN powers efficient searches within large datasets.

zilliz © Zilliz 2024 All rights reserved.

New Stuff

The Zilliz Sep '24 Launch

Supercharge Your GenAI with Production-Ready Data Infrastructure

Data Sovereignty

Migration Services

Secure, complete and easy data migration to Zilliz from Milvus, PGVector, Elastic or between Zilliz clusters

Fivetran Source Connector

Enable unstructured data retrievals from 500+ system through Fivetran

Fivetran

High-performance

Auto-Scale *(Private Preview)*

Dynamically increase cluster capacity based on workload to reduce operational burden and ensure continuous service

Multi-replica *(Public Preview)*

Replicated collections for increased throughput and fault tolerance

Security & Reliability

Metrics & Alerts

18 core metrics for proactive issue identification and 39 alerts to ensure optimal performance

Auth0-based SSO

(Private Preview)

Simplify IT management and improve security w/ Google/Github/SSO/Email login

99.95% Uptime SLAs

 © Copyright Zilliz

zilliz

New Stuff

https://github.com/milvus-io/milvus-sdk-java/releases/tag/v2.4.4

Milvus 2.4 introduces several new features and improvements:

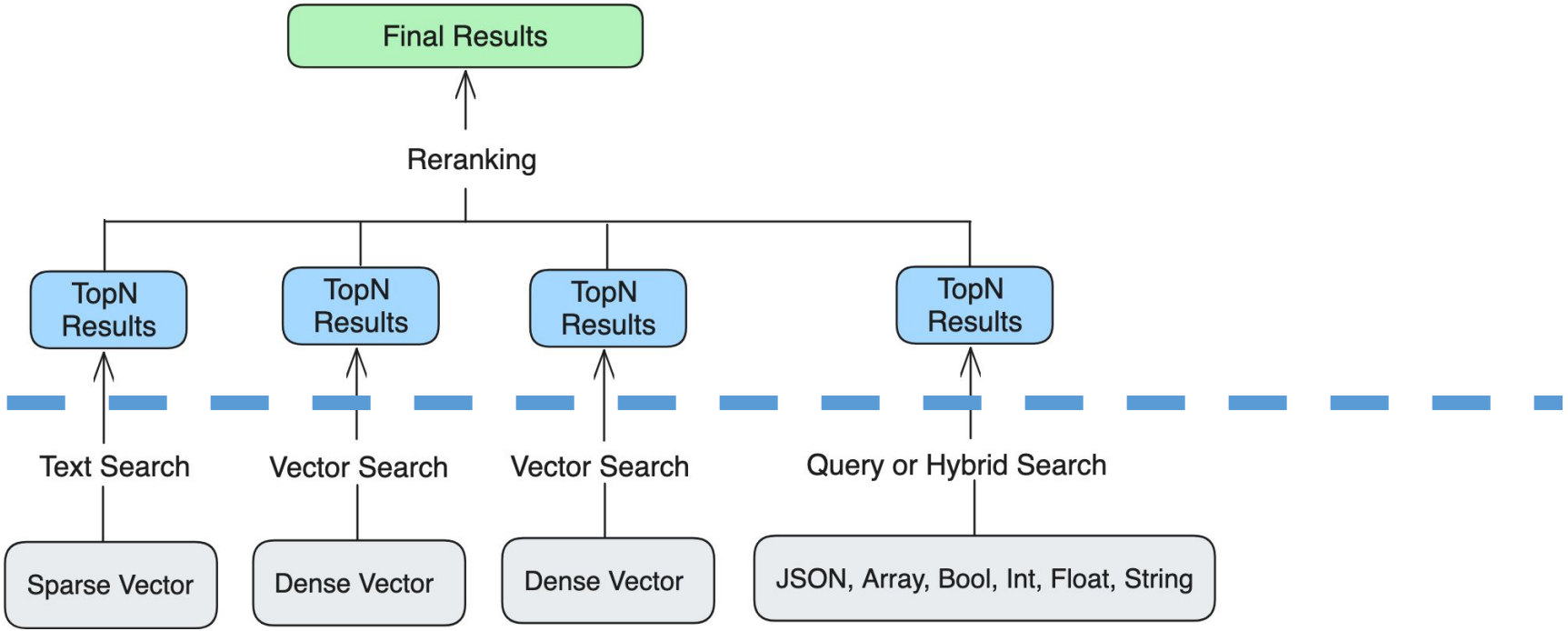
1. **New GPU Index - CAGRA**: This GPU-based index offers significant performance improvements, especially for batch searches
2. **Multi-vector and Hybrid Search**: This feature allows storing vector embeddings from multiple models and conducting hybrid searches.
3. **Sparse Vectors Support (Beta)**: Milvus now supports sparse vectors for processing in collections, which is particularly useful for keyword interpretation and analysis
4. **Grouping Search**: This feature enhances document-level recall for Retrieval-Augmented Generation (RAG) applications by providing categorical aggregation
5. **Inverted Index and Fuzzy Matching**: These capabilities improve keyword retrieval for scalar fields
6. **Float16 and BF16 Vector Data Type Support**: Milvus now supports these half-precision data types for vector fields, which can improve query efficiency and reduce memory usage.
7. **L0 Segment**: This new segment is designed to record deleted data, enhancing the performance of delete and upsert operations.
8. **Refactored BulkInsert**: The bulk-insert logic has been improved, allowing for importing multiple files in a single bulk-insert request.

© Zilliz 2023 All rights reserved.

 | © Copyright Zilliz

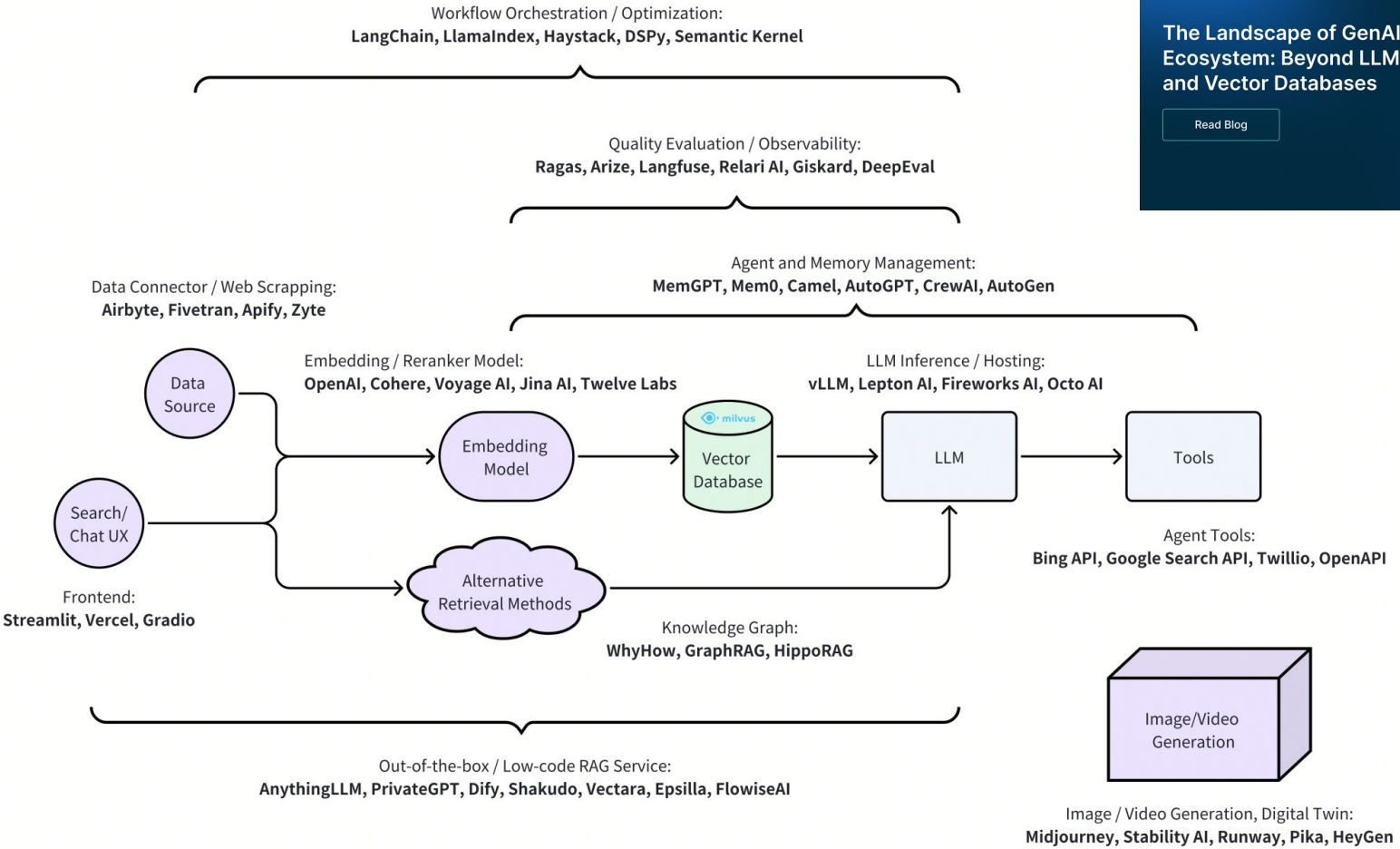
zilliz

Hybrid Search



 © Copyright Zilliz

zilliz



zilliz

The Landscape of GenAI Ecosystem: Beyond LLMs and Vector Databases

Read Blog

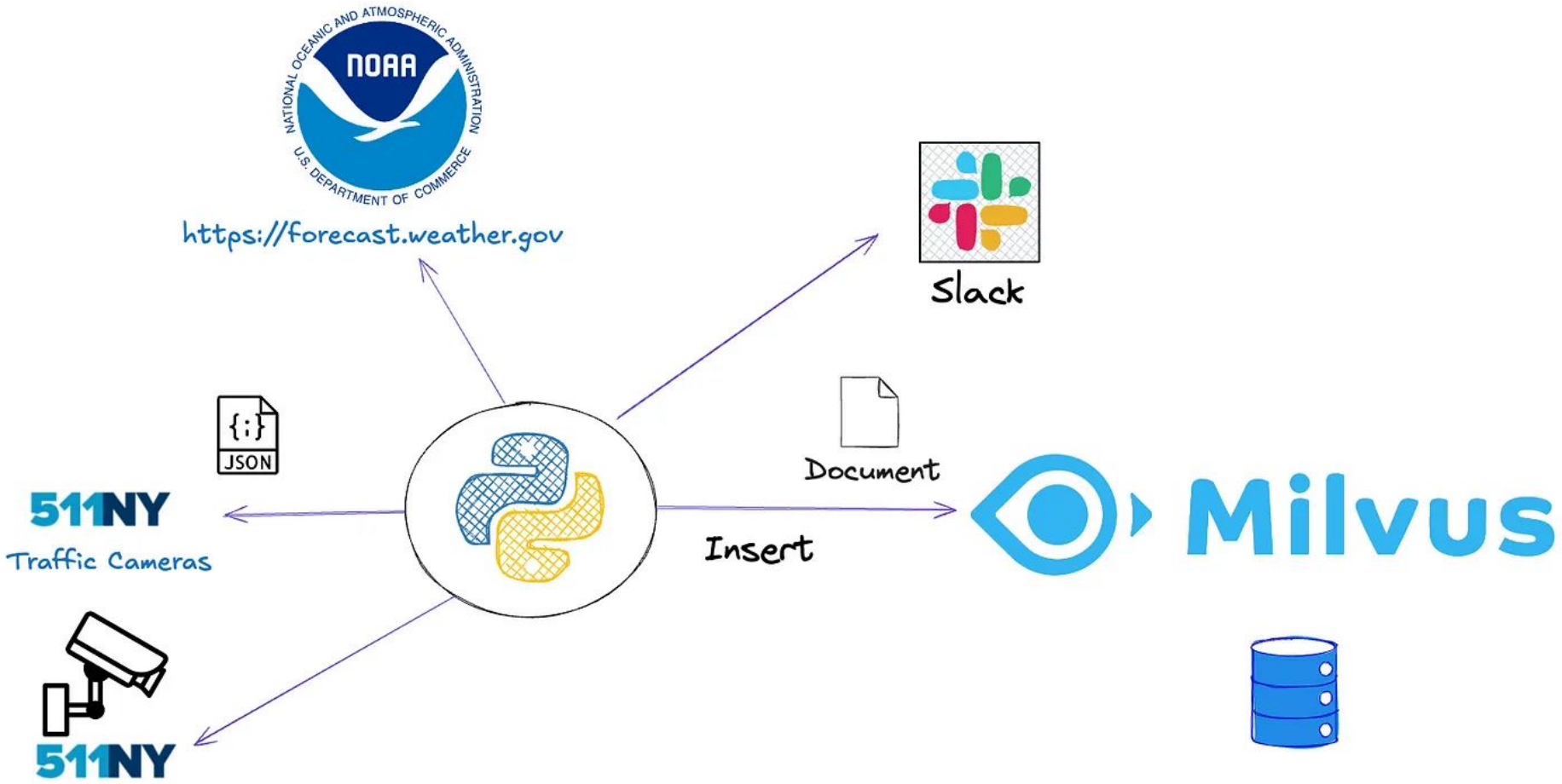
GenAI

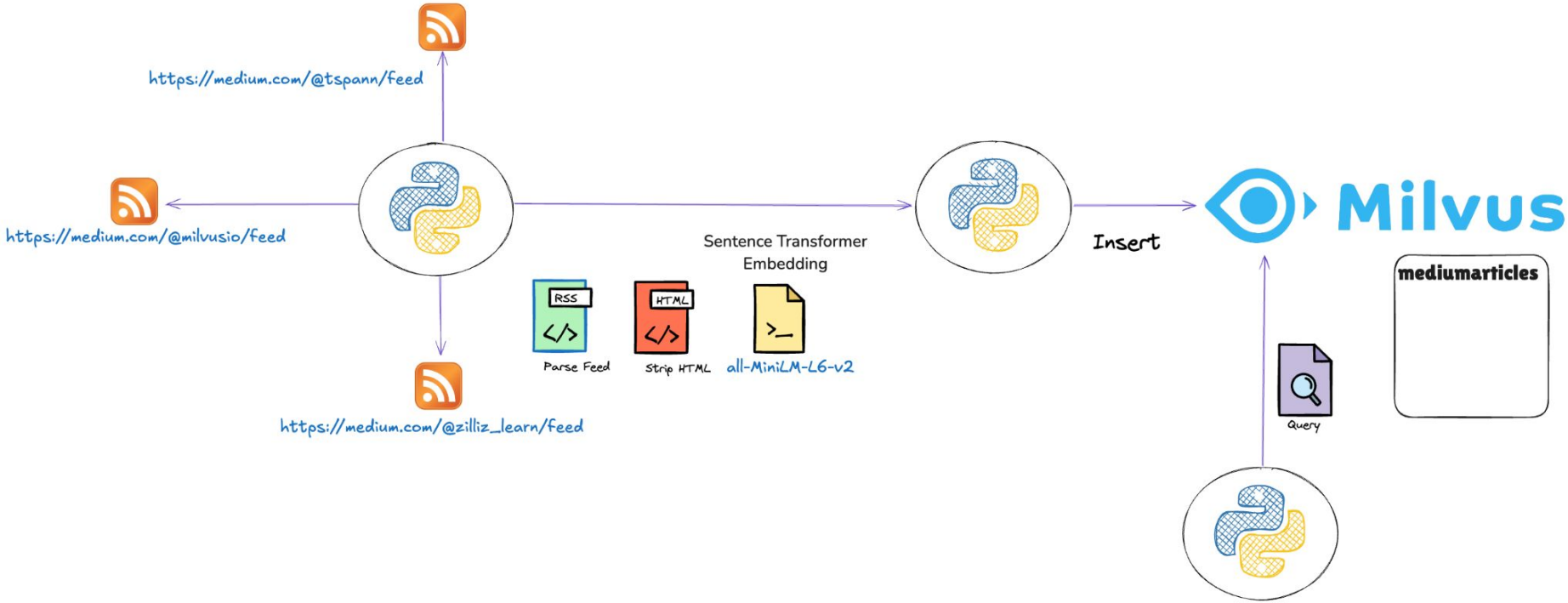
zilliz

04

APPS

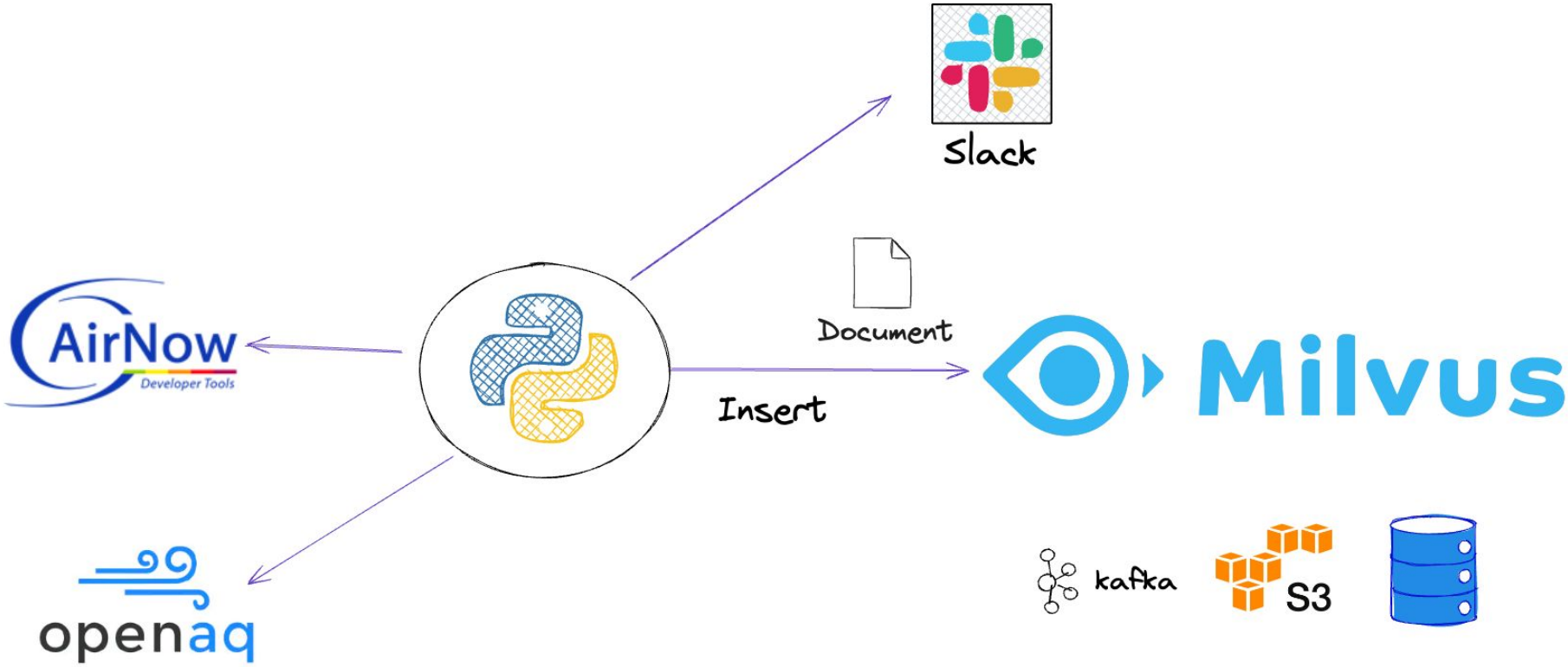
zilliz


NOAA
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE
https://forecast.weather.gov
Slack
JSON
Document
511NY
Traffic Cameras
Insert
Milvus
511NY



 | © Copyright Zilliz

zilliz



 | © Copyright Zilliz

zilliz

05

Q & A

zilliz

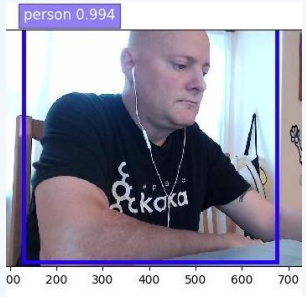
RESOURCES

 | © Copyright Zilliz

zilliz

Vector Database Resources

Give Milvus a Star!



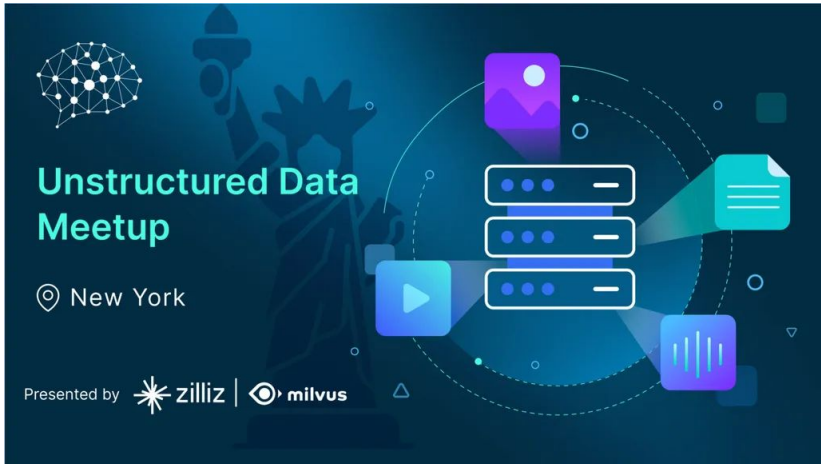
Chat with me on Discord!

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

 | © Copyright Zilliz

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

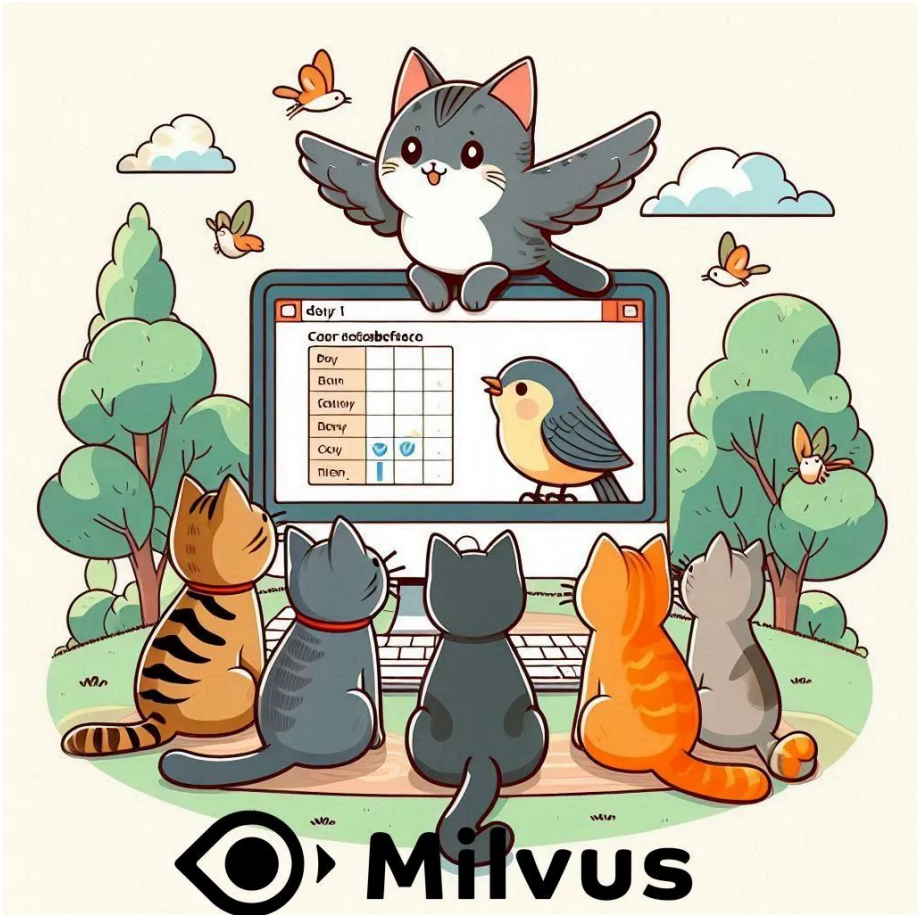
 | © Copyright Zilliz

zilliz

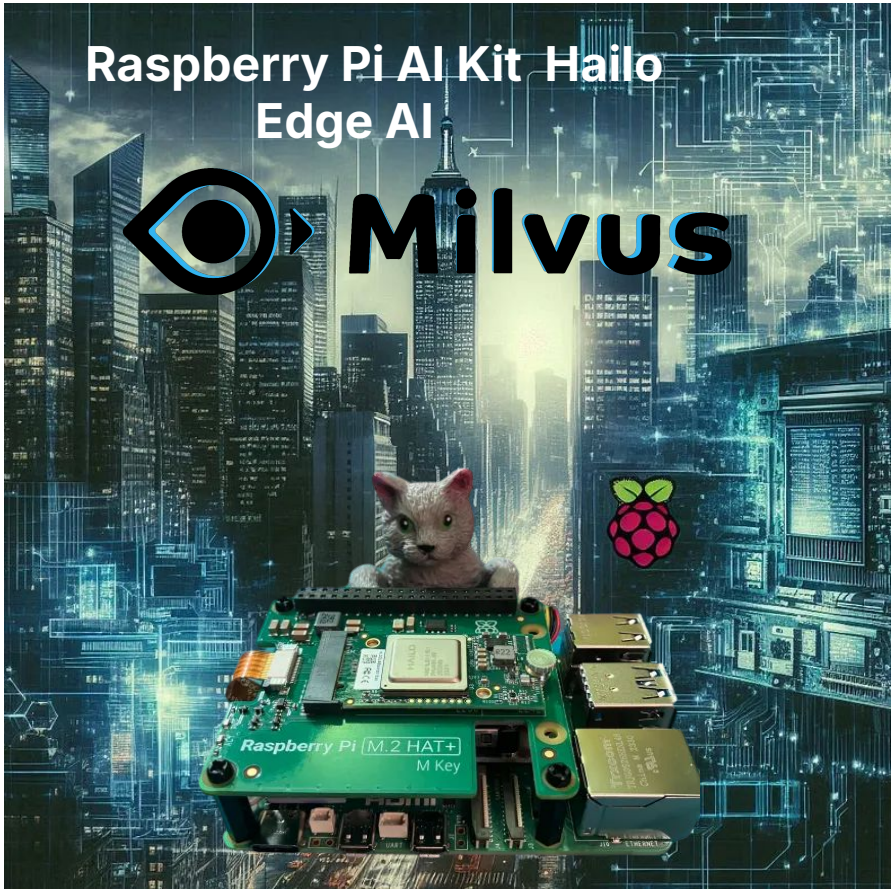

Learning Vector

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.

 | © Copyright 2024 Zilliz

zilliz

Thank you!

milvus.io

github.com/milvus-io/

@milvusio

Connect with me!

@paasDev

/in/timothyspann

 | © Copyright 2024 Zilliz

zilliz

Vector Search at Scale

 | © Copyright 8/16/23 Zilliz

zilliz

Indexing Strategies

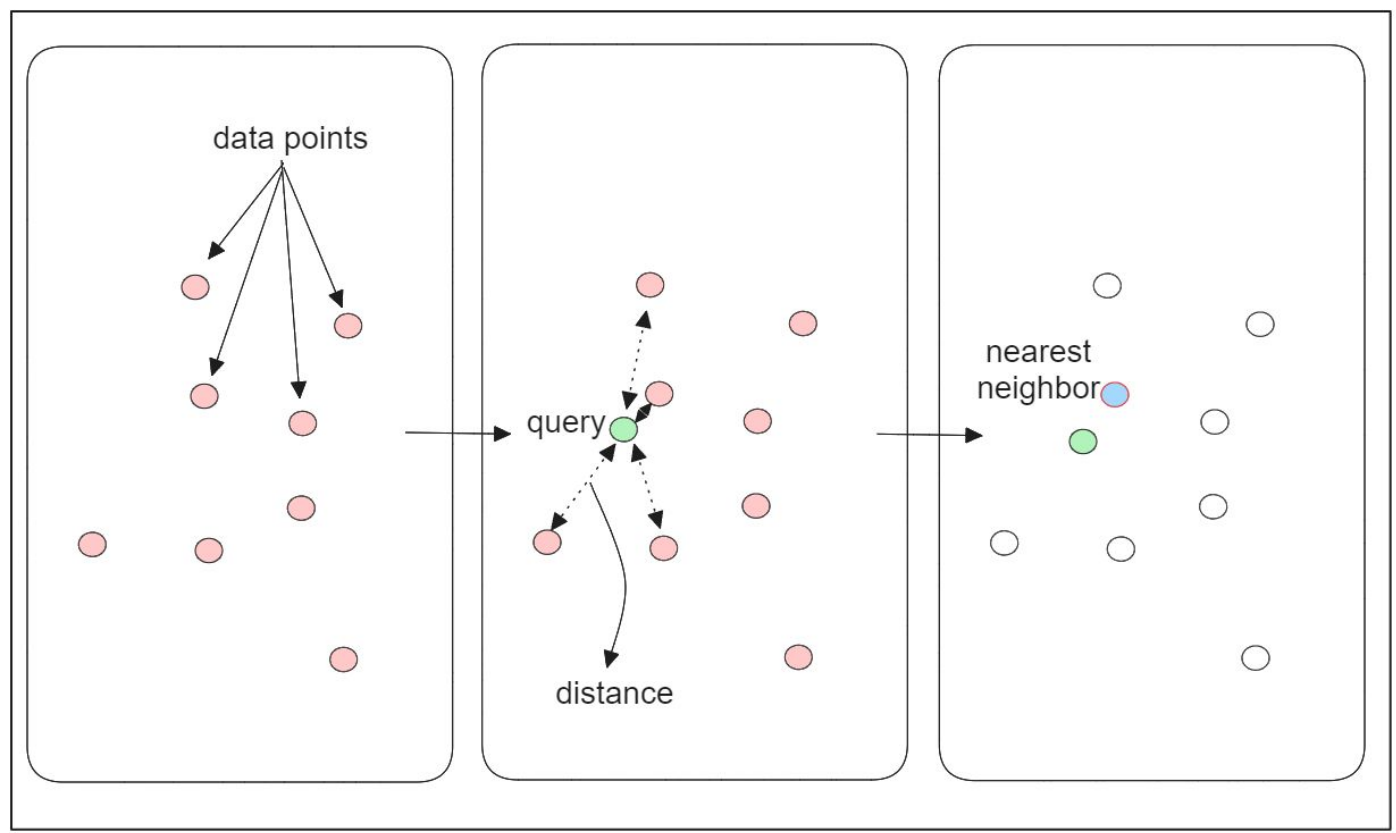
- Tree based
- Graph based
- Hash based
- Cluster based

FLAT

 | © Copyright 9/25/23 Zilliz

zilliz

FLAT Index



 | © Copyright 2024 Zilliz

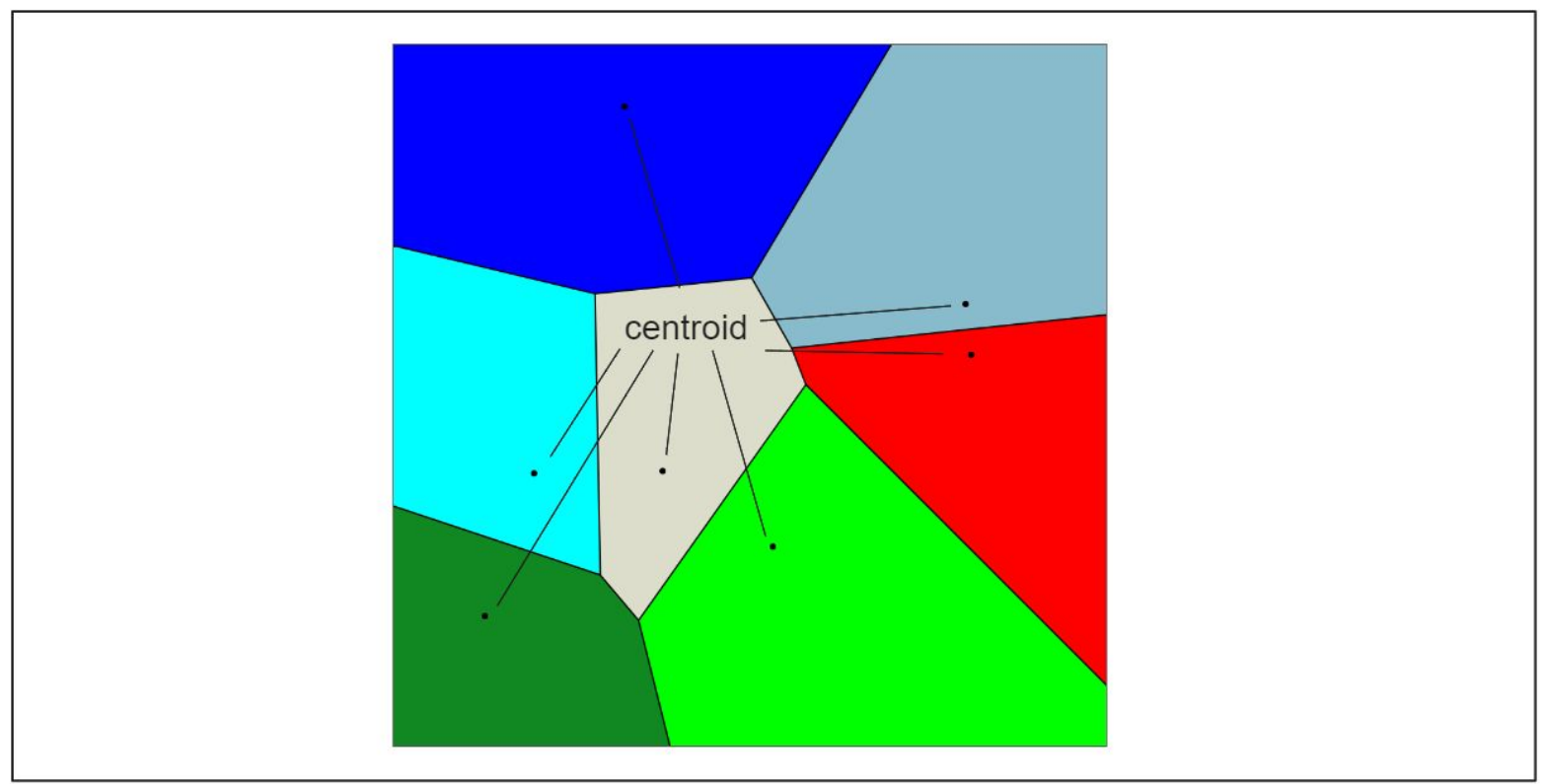
zilliz

Inverted File FLAT
(IVF-FLAT)

 | © Copyright 9/25/23 Zilliz

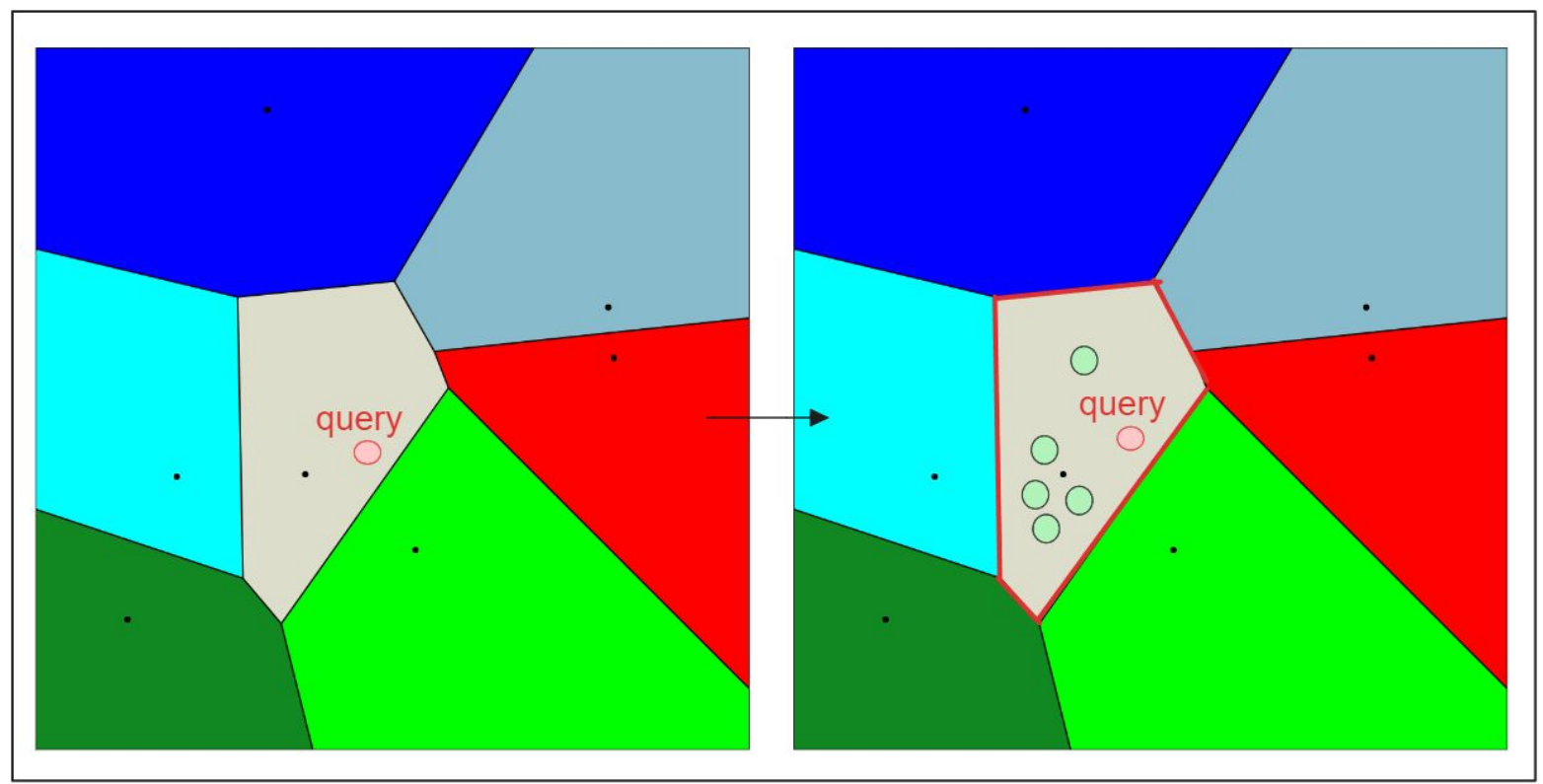
zilliz

IVF-FLAT Index



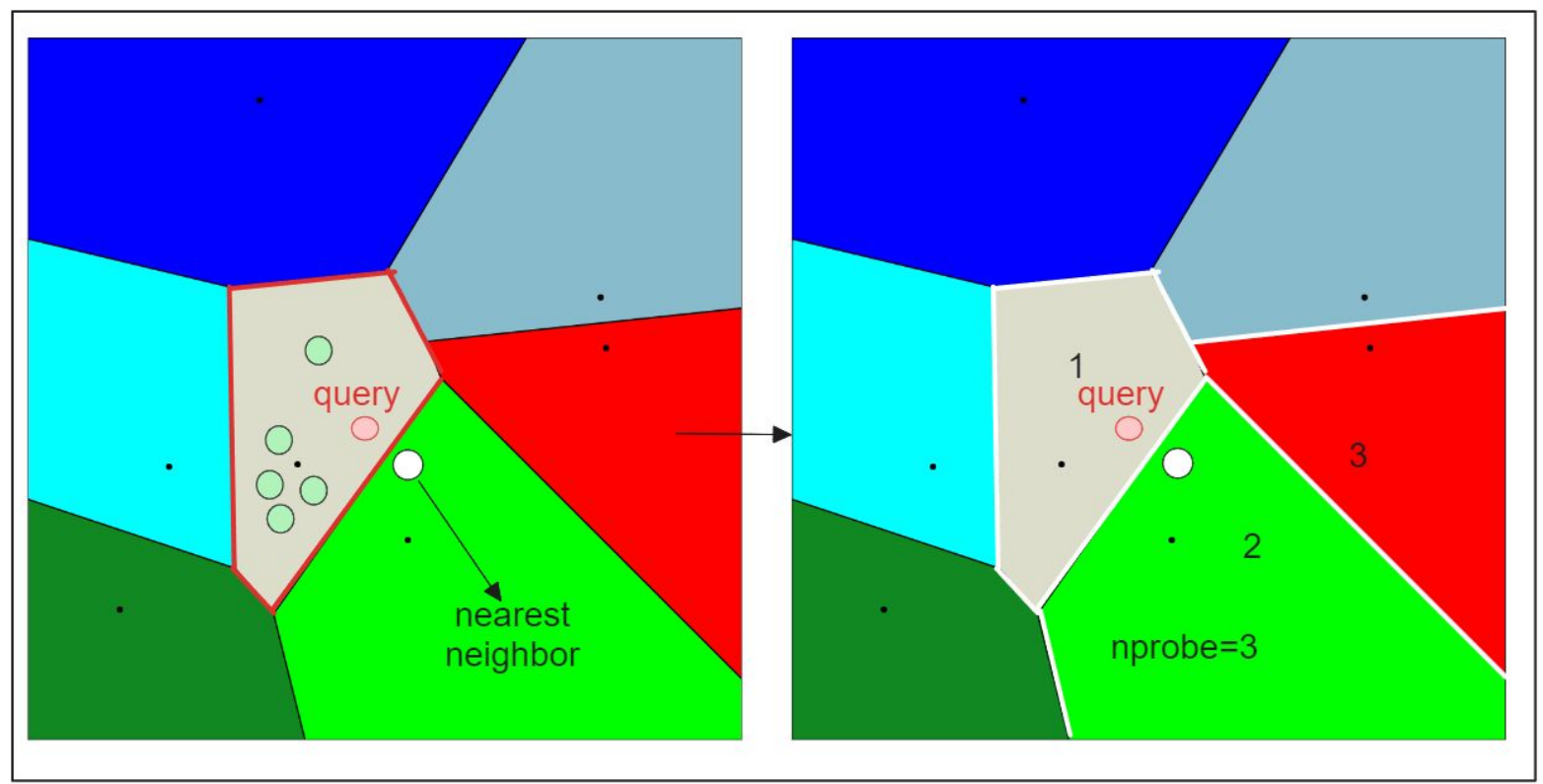
zilliz

IVF-FLAT Index



zilliz

IVF-FLAT Index



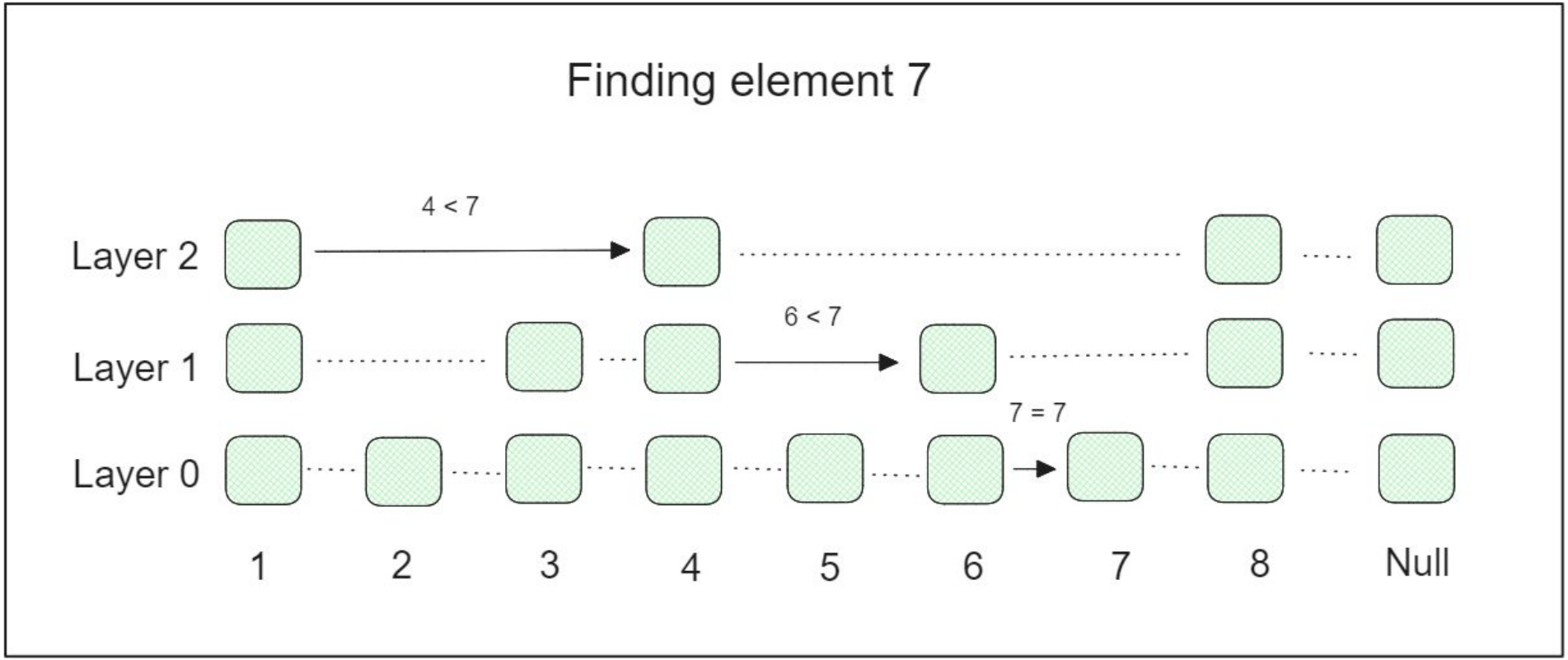
zilliz

Hierarchical Navigable Small World (HNSW)

 © Copyright 9/25/23 Zilliz

zilliz

HNSW - Skip List



 | © Copyright 2024 Zilliz

zilliz

HNSW - NSW Graph

- Built by randomly shuffling data points and inserting them one by one, with each point connected to a predefined number of edges (M).

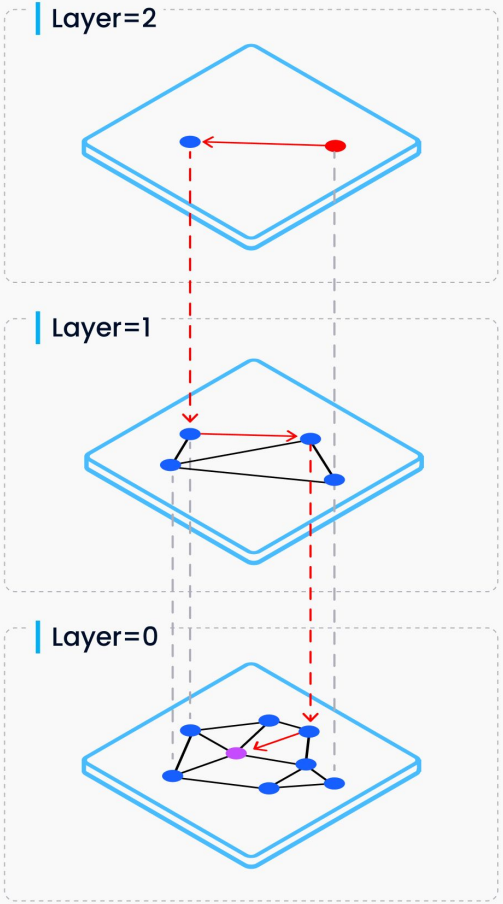
  ⇒ Creates a graph structure that exhibits the "small world".

  ⇒ Any two points are connected through a relatively short path.

 | © Copyright 2024 Zilliz

zilliz

HNSW



 | © Copyright 2024 Zilliz

zilliz

Category	Index	Accuracy	Latency	Throughput	Index Time	Cost
Graph-based	Cagra (GPU)	High	Low	Very High	Fast	Very High
	HNSW	High	Low	High	Slow	High
	DiskANN	High	High	Mid	Very Slow	Low
Quantization-based or cluster-based	ScaNN	Mid	Mid	High	Mid	Mid
	IVF_FLAT	Mid	Mid	Low	Fast	Mid
	IVF + Quantization	Low	Mid	Mid	Mid	Low

 © Copyrigt

zilliz

Picking an Index

- 100% Recall – Use `FLAT` search if you need 100% accuracy
- 10MB < `index_size` < 2GB – Standard IVF
- 2GB < `index_size` < 20GB – Consider `PQ` and `HNSW`
- 20GB < `index_size` < 200GB – Composite Index, `IVF_PQ` or `HNSW_SQ`
- Disk-based indexes

zilliz.com/learn/choosing-right-vector-index-for-your-project

 | © Copyright 2024 Zilliz

zilliz

Filtering

 | © Copyright 8/16/23 Zilliz

zilliz

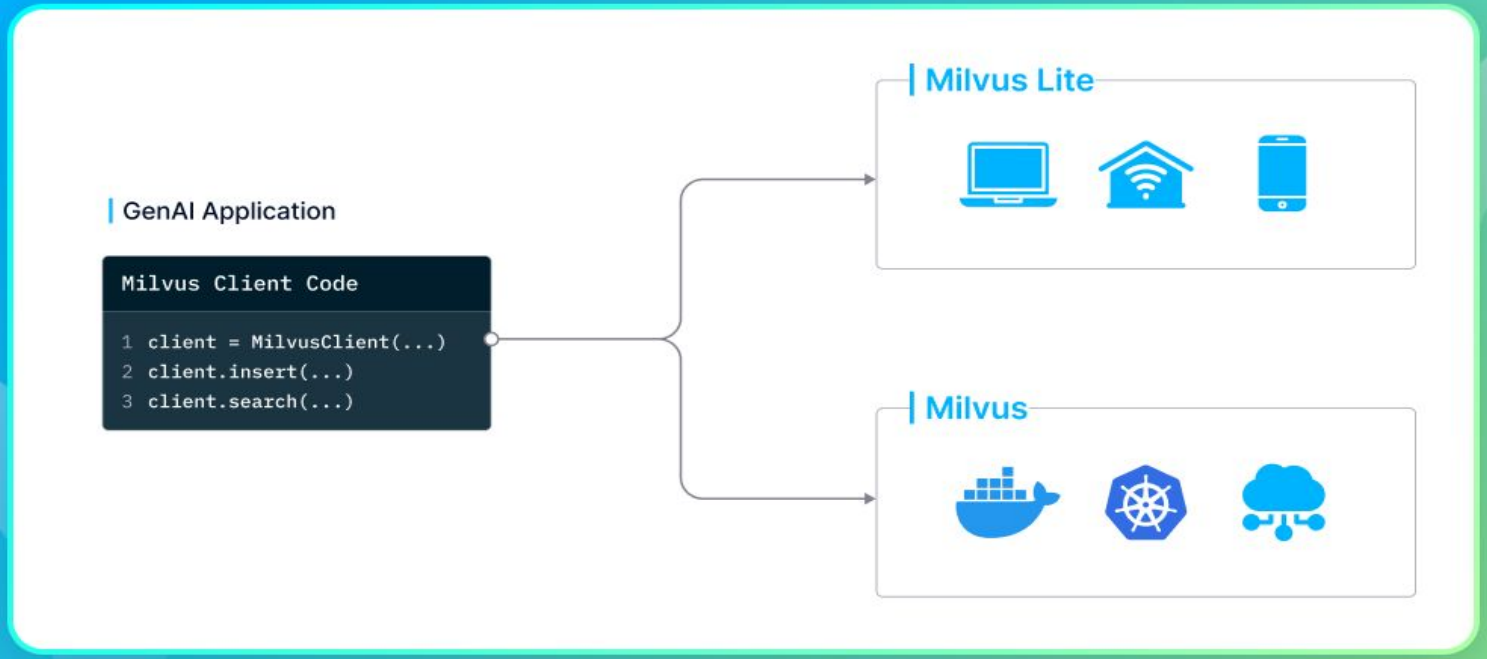
Filtering on Metadata

- Search Space Reduction w/ **Pre-Filtering**
- Bitset Wizardry 🧙
  - Use Compact Bitsets to represent Filter Matches
  - Low-level CPU operations for speed
- Scalar Indexing
  - Bloom Filter
  - Hash
  - Tree-based

 | © Copyright 2024 Zilliz

zilliz

Build Once
Deploy Anywhere

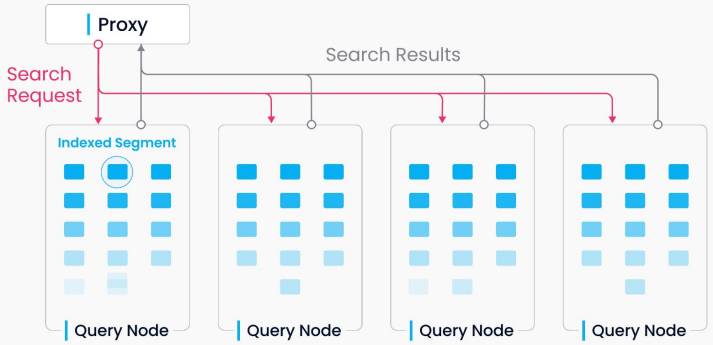


milvus

 © Copyright 2024 Zilliz

zilliz

Scalable Search



- **Distributed Search** across shards
- **Parallel** Processing
- **Query Optimization**

 | © Copyright 2024 Zilliz

zilliz

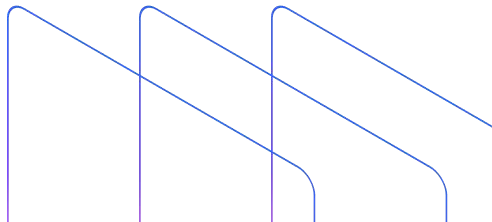
Why a Vector Database?

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



 | © Copyright 2024 Zilliz

zilliz

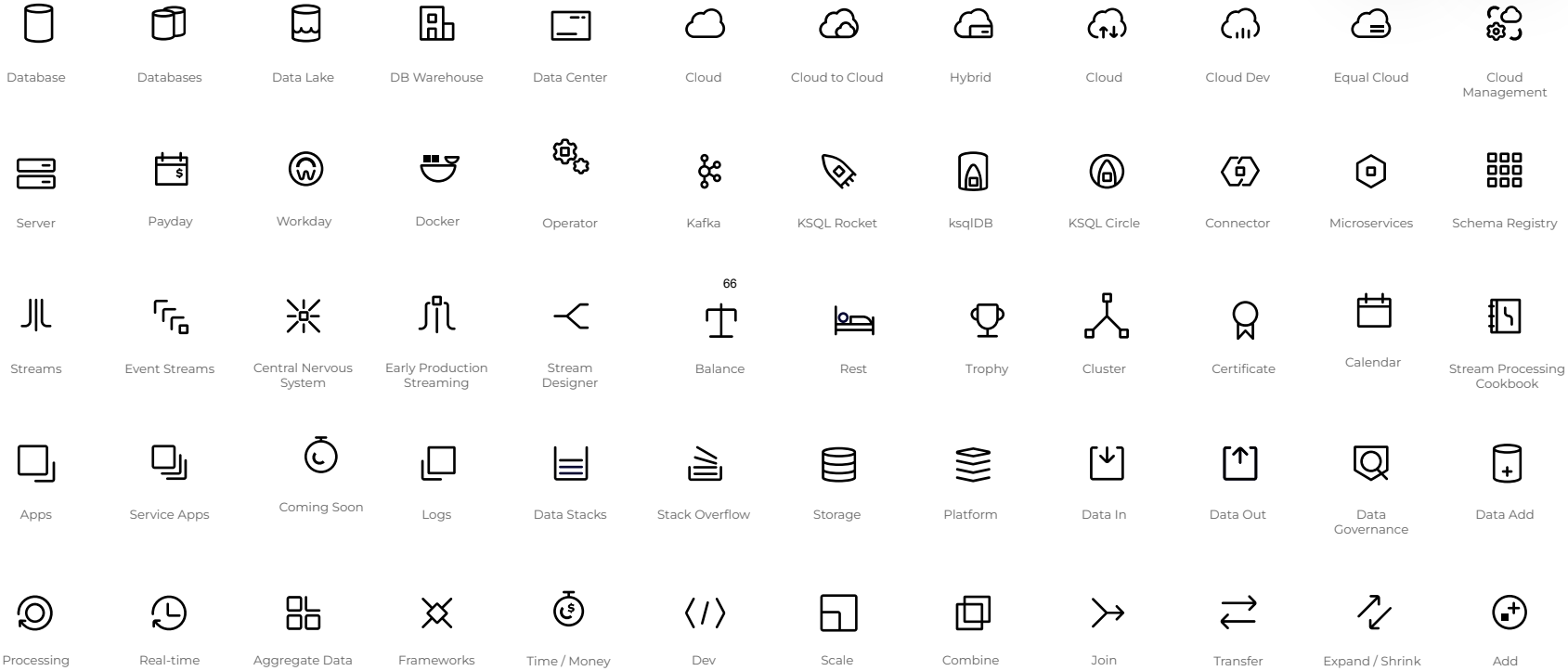
Comparison on functionalities

Feature	Milvus Lite	Milvus Standalone	Milvus Distributed
SDK / Client Lirary	Python gRPC	Python Go Java Node.js C# RESTful	Python Java Go Node.js C# RESTful
Data types	Dense Vector Sparse Vector Binary Vector Boolean Integer Floating Point VarChar Array JSON	Dense Vector Sparse Vector Binary Vector Boolean Integer Floating Point VarChar Array JSON	Dense Vector Sparse Vector Binary Vector Boolean Integer Floating Point VarChar Array JSON
Search capabilities	Vector Search (ANN Search) Metadata Filtering Range Search Scalar Query Get Entities by Primary Key Hybrid Search	Vector Search (ANN Search) Metadata Filtering Range Search Scalar Query Get Entities by Primary Key Hybrid Search	Vector Search (ANN Search) Metadata Filtering Range Search Scalar Query Get Entities by Primary Key Hybrid Search
CRUD operations	✓	✓	✓
Advanced data management	N/A	Access Control Partition Partition Key	Access Control Partition Partition Key Physical Resource Grouping
Consistency Levels	Strong	Strong Bounded Staleness Session Eventual	Strong Bounded Staleness Session Eventual

zilliz © Zilliz 2024 All rights r

Icons

ICON STYLE:



Current

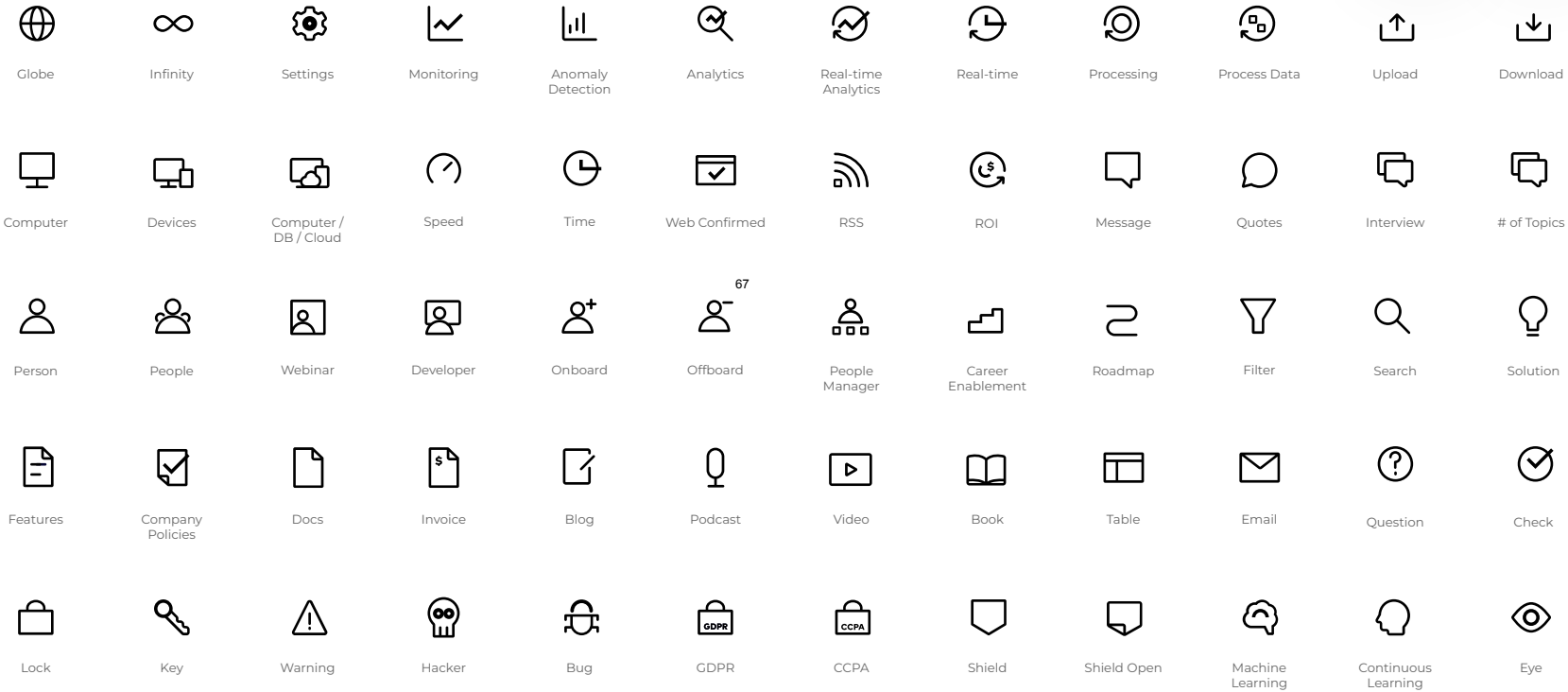
For the complete, most updated collection of Icons please go to: https://cnfl.io/Icons

 | © Copyright 2024 Zilliz

zilliz

Icons

ICON STYLE:



Current

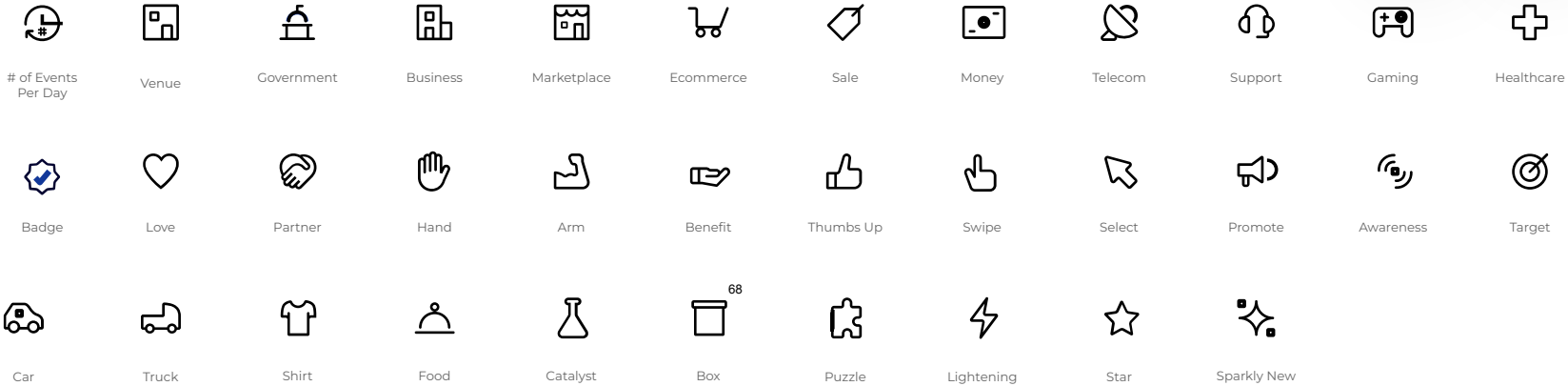
For the complete, most updated collection of Icons please go to: https://cnfl.io/Icons

 | © Copyright 2024 Zilliz

zilliz

Icons

ICON STYLE:



For the complete, most updated collection of Icons please go to: https://cnfl.io/Icons

 | © Copyright 2024 Zilliz