

watsonx.data™

competitive insights

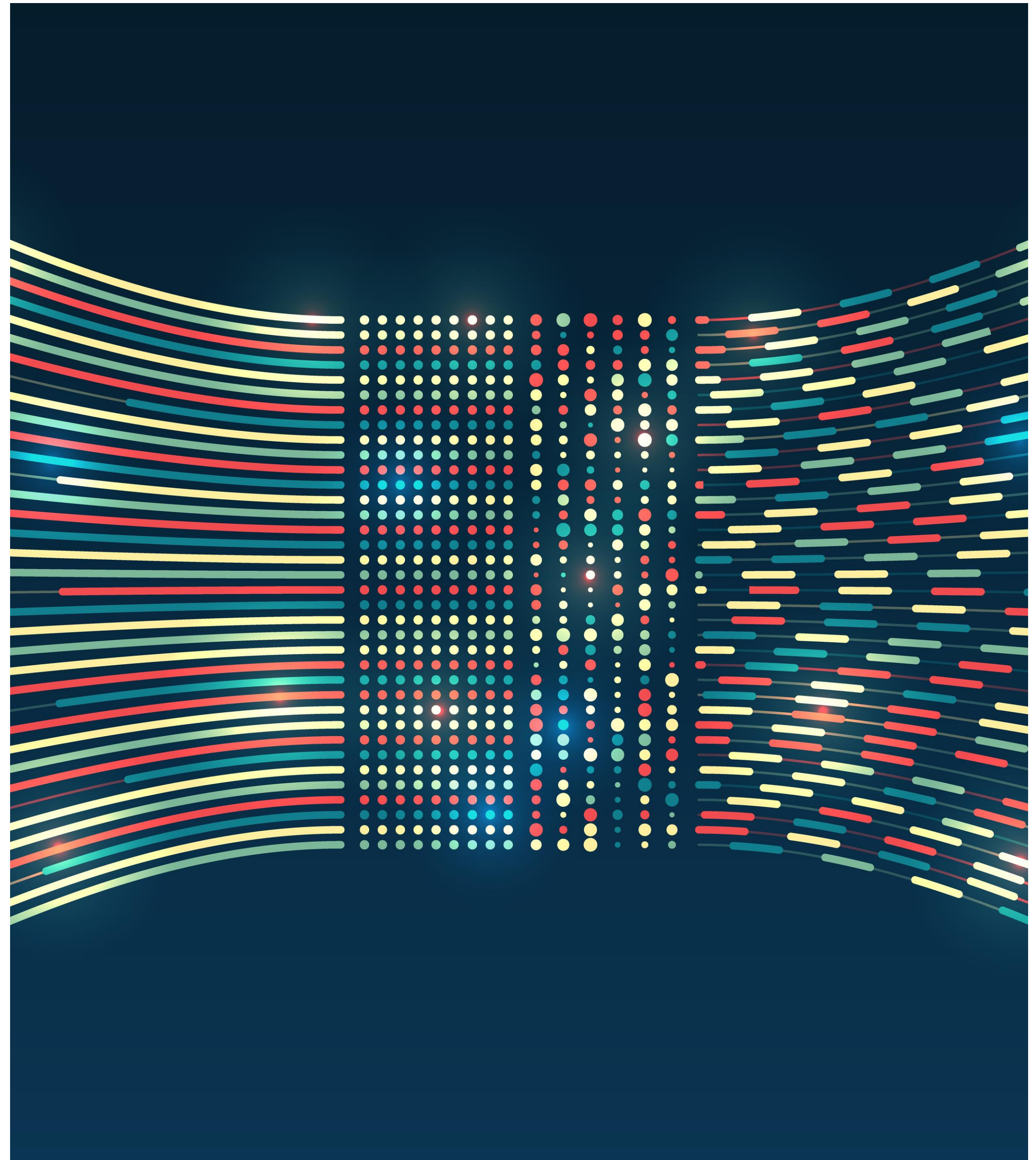
An introduction to Milvus with
positioning against competitors

Danny Arnold

Principal, Learning Content Development

Competitive Insights, AI & Data

darnold@us.ibm.com



Seller guidance and legal disclaimer

IBM and Business Partner
Internal Use Only

Slides in this presentation marked as **"IBM and Business Partner Internal Use Only"** are for IBM and Business Partner use and should not be shared with clients or anyone else outside of IBM or the Business Partners' company.

© IBM Corporation 2024.
All Rights Reserved.

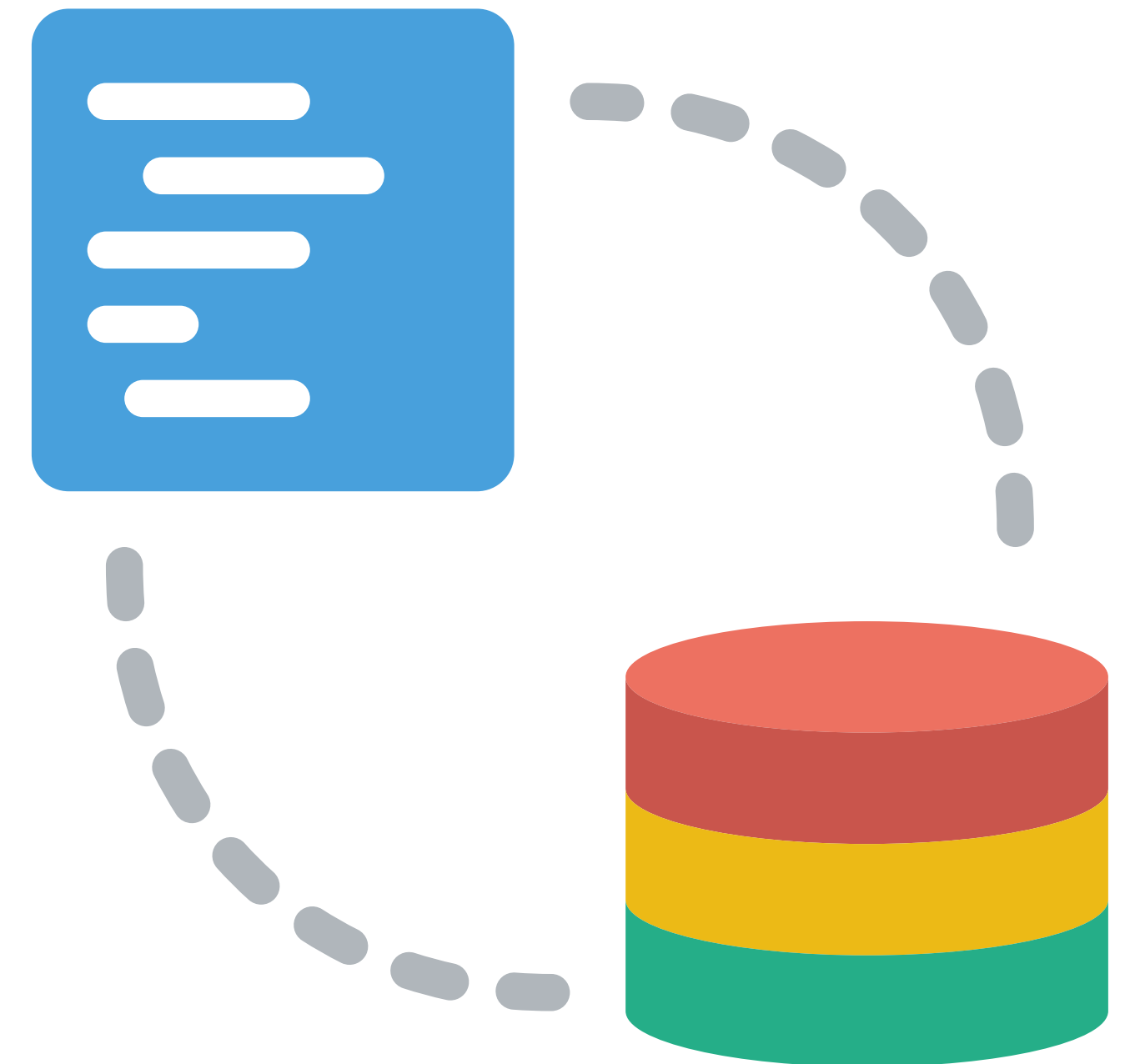
The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth, or other results.

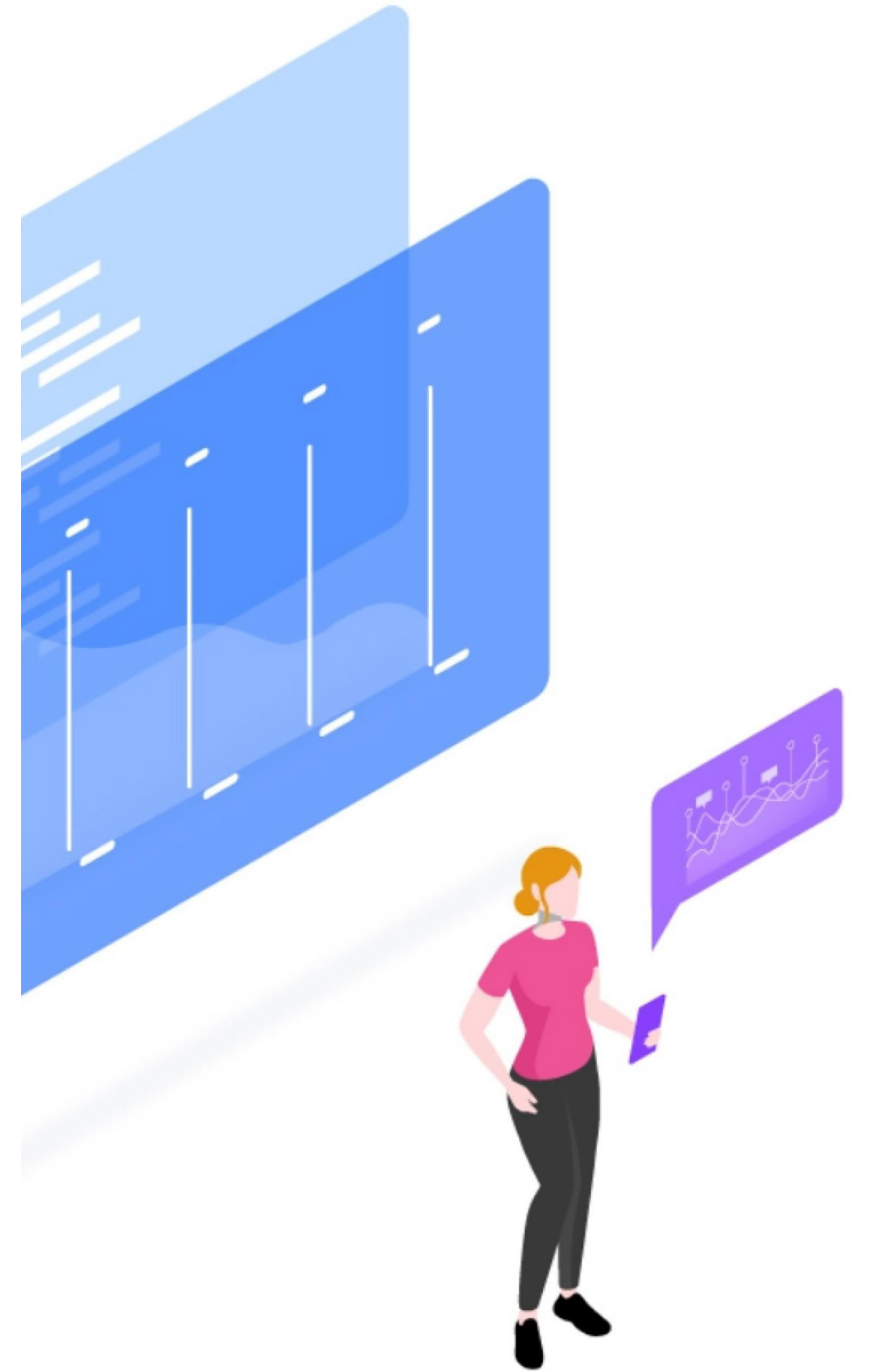
All client examples described are presented as illustrations of how those clients have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by client.

Topics for discussion

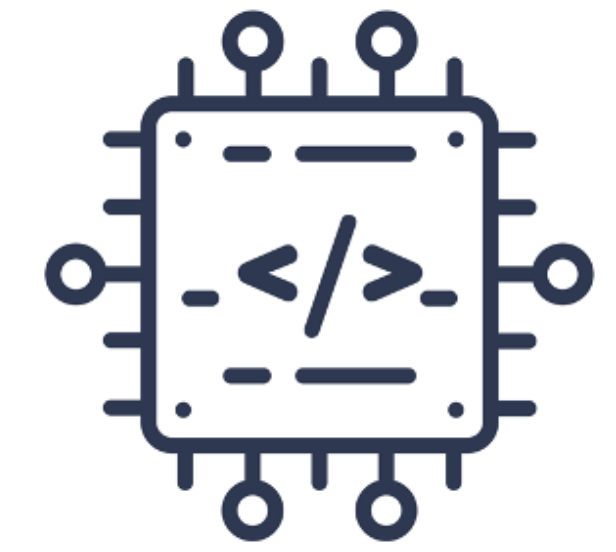
- Introduction to Milvus and vector database concepts
- Types of competitors
- Competitive landscape for vector databases
- Important considerations for selecting a vector database
- Background information on competitors
- Competitors' key strengths and weaknesses
- Quick view of competition and Milvus differentiators
- Objection handling
- Setting traps for competitors



Introduction and overview



What is a vector database?



According to Wikipedia

From Wikipedia, the free encyclopedia

A **vector database management system (VDBMS)** or simply **vector database** or **vector store** is a **database** that can store vectors (fixed-length lists of numbers) along with other data items. Vector databases typically implement one or more **Approximate Nearest Neighbor (ANN)** algorithms,^{[1][2]} so that one can search the database with a query vector to retrieve the closest matching database records.

Vectors are mathematical representations of data in a high-dimensional space. In this space, each dimension corresponds to a **feature** of the data, and tens of thousands of dimensions might be used to represent sophisticated data. A vector's position in this space represents its characteristics. Words, phrases, or entire documents, and images, audio, and other types of data can all be vectorized.^[3]

Vector databases store vector embeddings that enable the database to compare vectors using specific measures of similarity

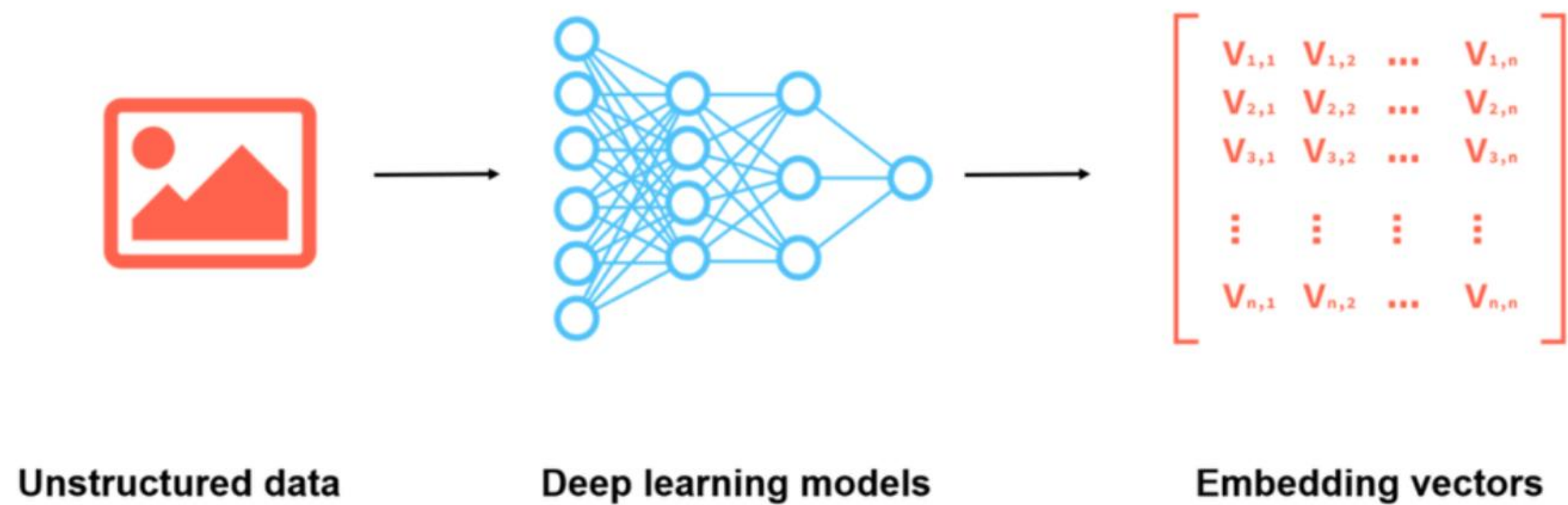
Vector embeddings are arrays of real numbers of a fixed length that represent unstructured data

Rather than looking for exact data matches, like a relational database, vector databases use algorithms that perform a similarity search between the query vector and the vector embeddings within the database

- Approximate nearest neighbor (ANN) search algorithms are used to find similarities

Vector embedding

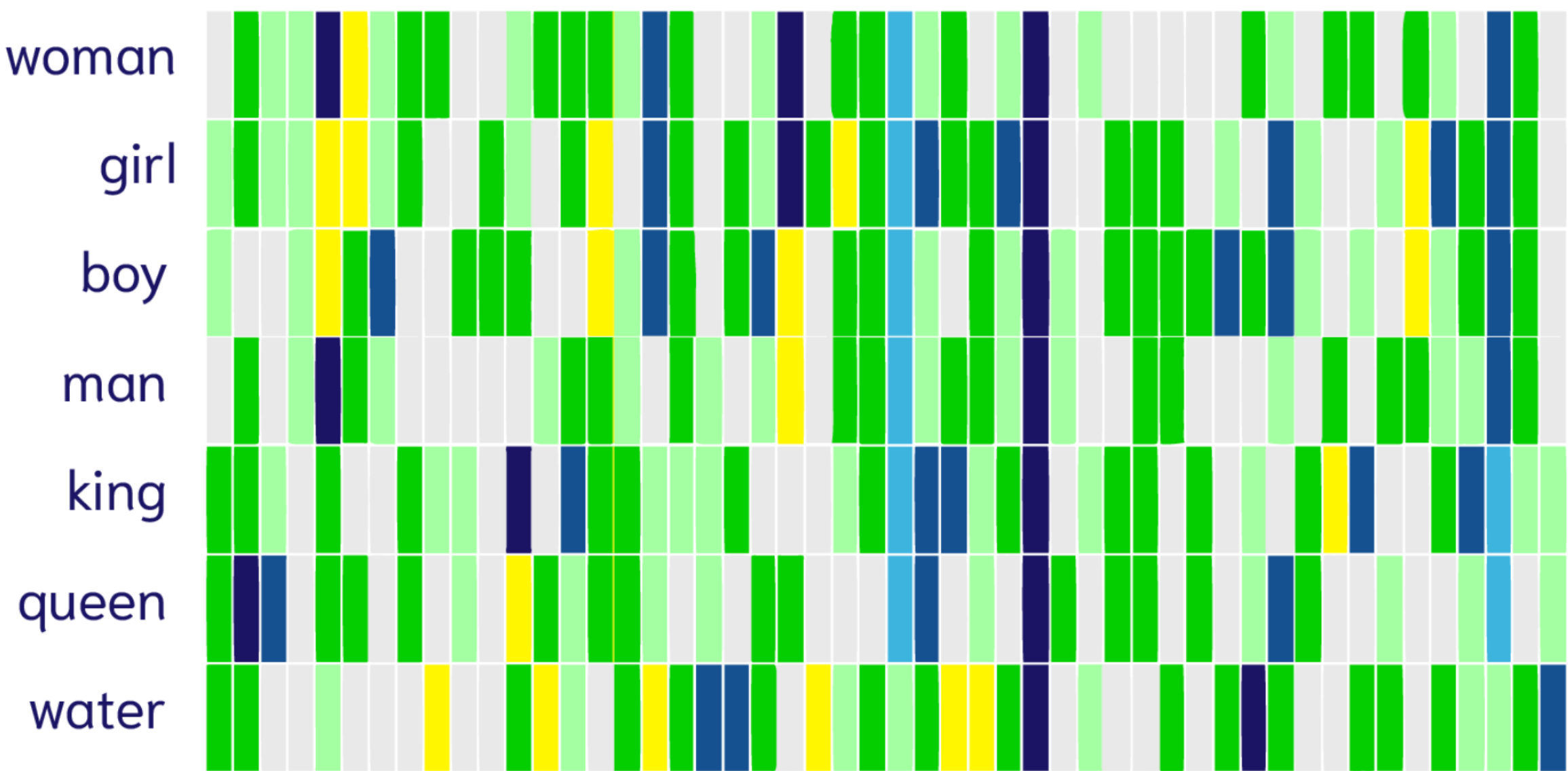
The flow of unstructured data from its source uses a large language model (LLM) to generate the vector embedding prior to loading the data into a vector database



[Source](#)

Vector embeddings are created using the same algorithm and are the same length

An example vector embedding showing words represented by an array of 50 numbers



[Source](#)

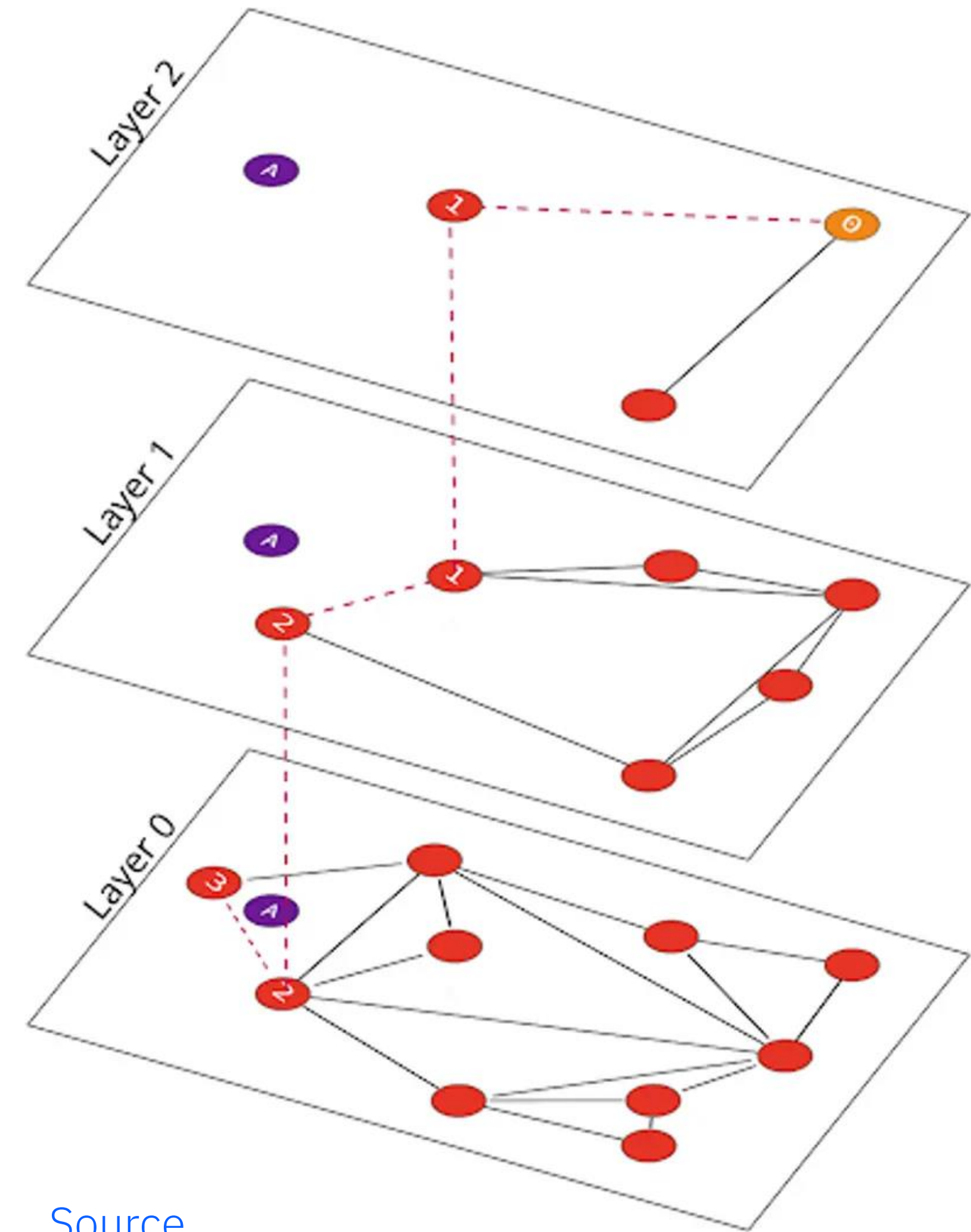
Indexes within vector databases

Vector databases use several types of indexes to assist with vector similarity search performance

Hierarchical Navigable Small Worlds (HNSW)

graphs are among the most popular to efficiently find nearest neighbors in large, multi-dimensional datasets

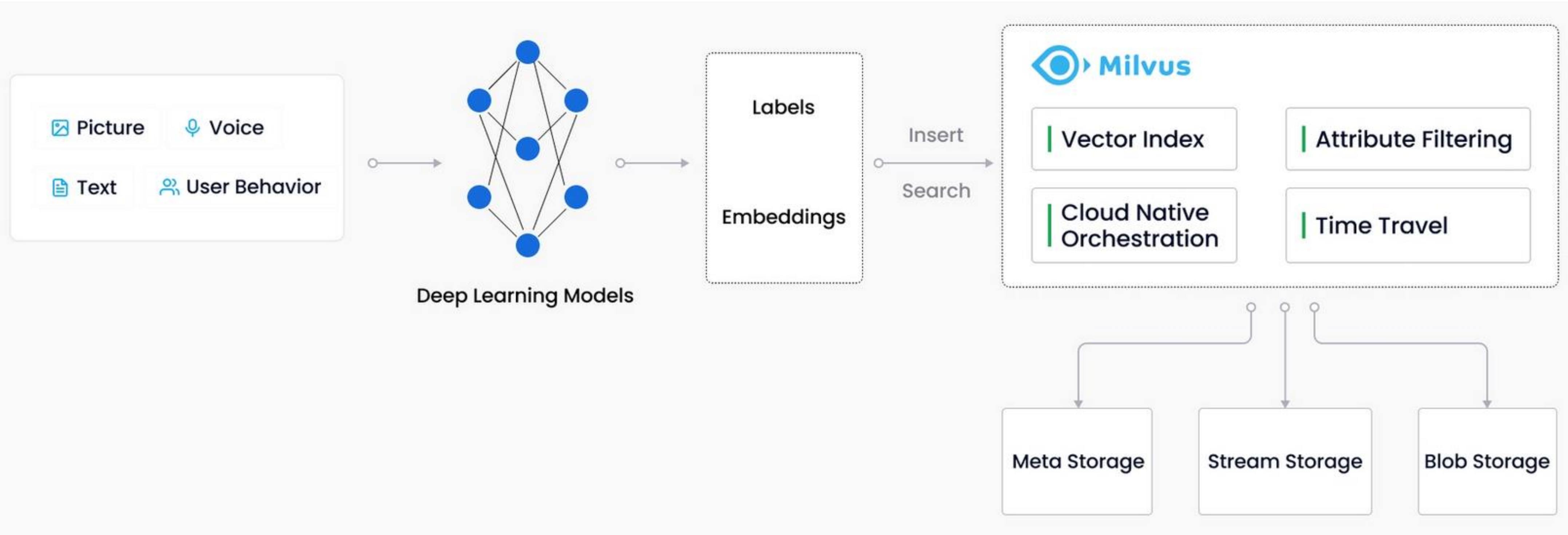
- Each layer is a "small world" with the top layers providing coarser granularity searches
- The lower layers provide most of the similarity search results (the approximate nearest neighbors or ANN)



[Source](#)

Other types of indexes

Index type	Description
Flat indexing	Each vector is stored as-is with no modifications. Simple and easy to implement with very accurate results. Similarity between the query vector and every other vector is computed and the K vectors with the smallest similarity score are returned and are exact nearest neighbors. This is the slowest indexing method for performance.
Locality Sensitive Hashing (LSH) indexes	Index is built using a hashing function and vector embeddings that are nearby each other are hashed to the same bucket. These similar vectors are stored in the same table or bucket. Query vector provided is hashed and then a similarity value is computed for all vectors in the table with the vectors that were hashed to the same bucket. This results in a much smaller search than using the flat indexing method and provides the approximate nearest neighbors in a much faster query result.
InVerted File indexes (IVF)	Same goals as the LSH indexing method, but partitions or clusters the vector space first. The centroids of each cluster are calculated. For a given query vector, the closest centroid is located and then for that centroid, vectors in the associated cluster are searched. This method finds the approximate nearest neighbors. A problem occurs when a query vector is on the edge of multiple clusters and requires a multi-cluster search.



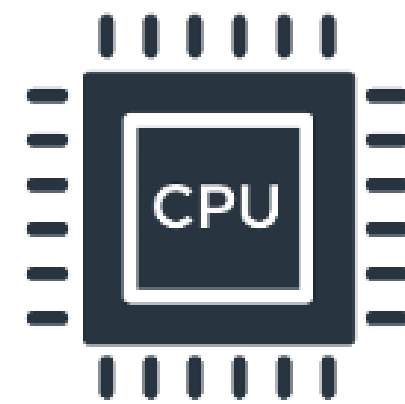
[Source](#)

The difference between IBM Milvus and open source Milvus



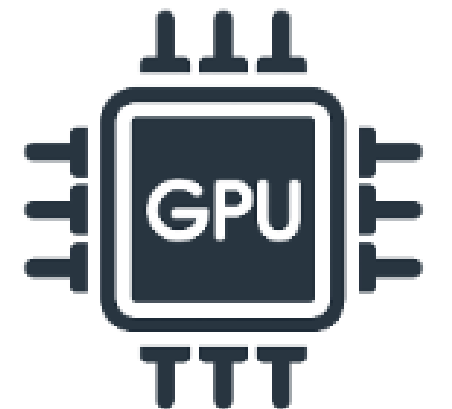
IBM Milvus supported vector index types

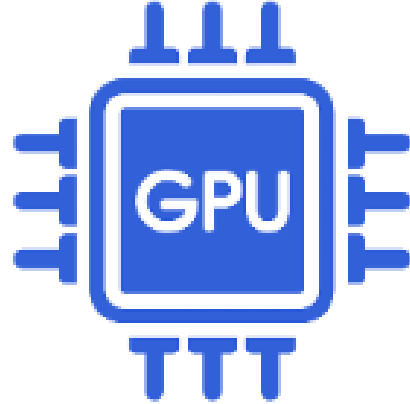
- FLAT
- IVF_FLAT
- IVF_PQ
- HNSW
- SCANN



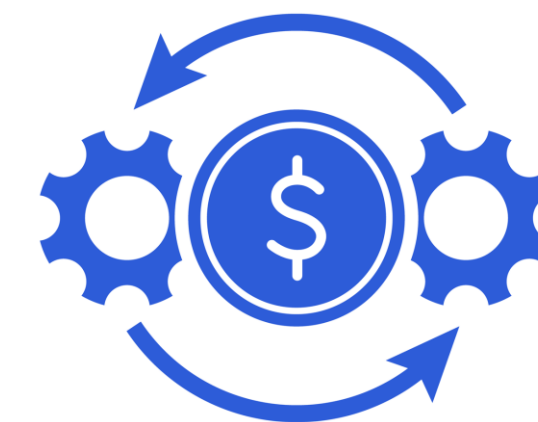
Milvus vector index types that are not supported by IBM Milvus

- GPU_IVF_FLAT
- GPU_IVF_PQ
- IVF_SQ8



NO 
REQUIREMENTS

IBM Milvus delivers cost effective compute within a vector database by removing index types that require Graphical Processing Units (GPUs)



Types of competitors

There are two types of solutions in the vector database market

- Dedicated vector databases
- Databases that support vector search

Dedicated vector databases

- Chroma
- LanceDB
- Marqo
- Pinecone
- Qdrant
- Vald
- Vespa
- Weaviate

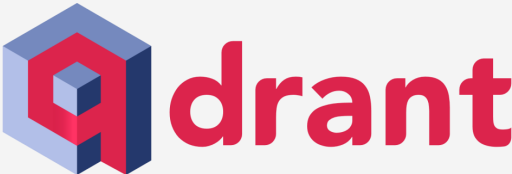



Databases that support vector search

- Cassandra
- ClickHouse
- Elasticsearch
- OpenSearch
- PostgreSQL
- Redis
- Rockset
- SingleStore

Competitive landscape





Dedicated vector databases

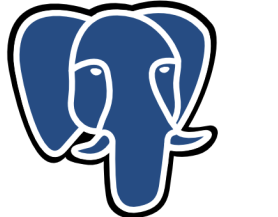



Details	 Chroma	 LanceDB	 marqo	 Pinecone
Open source software	Yes	Yes	Yes	Proprietary
Deployment options	<ul style="list-style-type: none">Self managedFully managedEmbeddable	<ul style="list-style-type: none">Self managedFully managedEmbeddable	<ul style="list-style-type: none">Self managedFully managed	<ul style="list-style-type: none">Fully managed
Integrations	10 to 15	10 to 15	Less than 10	15 to 20
Year released	2022	2023	2024	2021

Details	 drant	 Vald	 vespa	 Weaviate
Open source software	Yes	Yes	Yes	Yes
Deployment options	<ul style="list-style-type: none">Self managedFully managed	<ul style="list-style-type: none">Self managedFully managed	<ul style="list-style-type: none">Self managedFully managed	<ul style="list-style-type: none">Self managedFully managed
Integrations	More than 20	Less than 10	Less than 10	10 to 15
Year released	2021	2021	2017	2019

Competitive landscape

Databases that support vector search

Details	 <i>cassandra</i>	 ClickHouse	 elasticsearch	 OpenSearch
Open source software	Yes	Yes	No Free version with license restrictions available.	Yes
Deployment options	<ul style="list-style-type: none"> Self managed Fully managed 	<ul style="list-style-type: none"> Self managed Fully managed 	<ul style="list-style-type: none"> Self managed Fully managed 	<ul style="list-style-type: none"> Self managed Fully managed
Integrations	Less than 10	Less than 10	Less than 10	Less than 10
Year released	2023	2023	2022	2023

Details	 PostgreSQL	 redis	 [ROCKSET]	 SingleStore
Open source software	Yes	Yes	Yes Based on RocksDB.	No Free version available.
Deployment options	<ul style="list-style-type: none"> Self managed Fully managed 	<ul style="list-style-type: none"> Self managed Fully managed 	<ul style="list-style-type: none"> Fully managed 	<ul style="list-style-type: none"> Self managed Fully managed
Integrations	Less than 10	Less than 10	10 or more	Less than 10
Year released	2023	2023	2023	2024

Note: Year released is the year the vector search capability was introduced

Important considerations when selecting a vector database

Vector search can require large amounts of data and may require analysis of billions of vectors

- Watch out for vector databases that can only support single node deployments
- Watch out for vector databases that have limits on the number of vectors that can be analyzed in the millions versus billions

GitHub star ratings provide a measure of community size and usage of open source vector databases

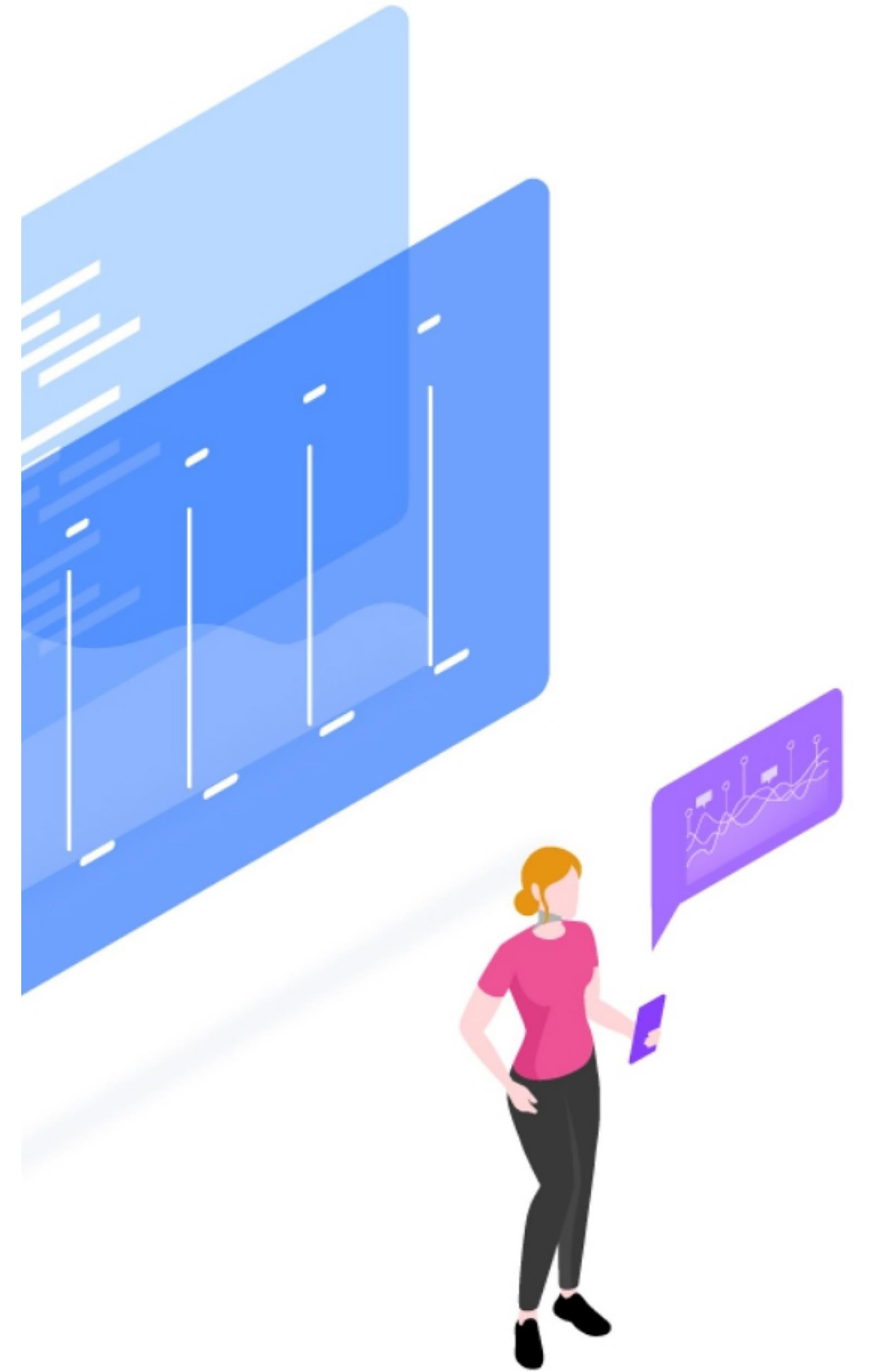
- Watch out for vector databases with less than 5K stars, they are either very new and/or have very small communities of support

Vector indexes are important to achieve good performance of similarity vector searches

- Watch out for vector databases that have only have support for 1 or 2 vector index types



Quick view of competition

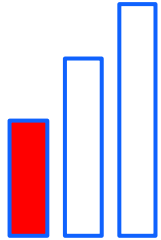
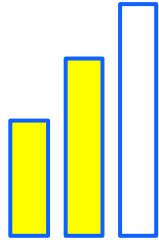
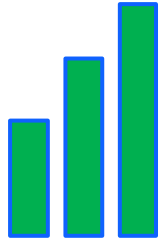











Vector database

Quick view of competition

Integration with AI and governance components

Dedicated vector databases

 Minimal or no integration with other components	 Some integrations with AI and governance components	 Fully integrated in an AI and ML platform
 Vald 	 Weaviate  Pinecone  marqo  Chroma  drant  LanceDB	 IBM milvus

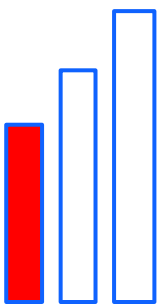
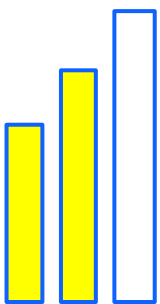
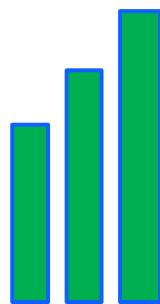




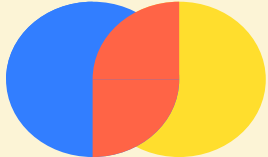






Vector database

Quick view of competition

Open source based

Dedicated vector databases

 Proprietary	 Open source with a single vendor focused community	 Open source and strong community
 Pinecone	<div> Weaviate</div> <div> drant</div> <div> marqo</div> <div> Chroma</div> <div> vespa</div> <div> LanceDB</div> <div> Vald</div>	 milvus

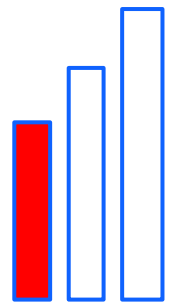
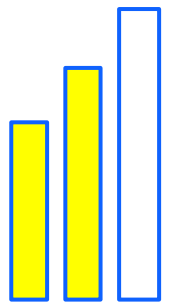
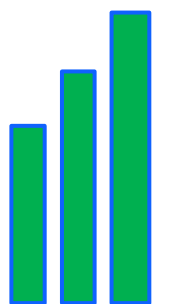







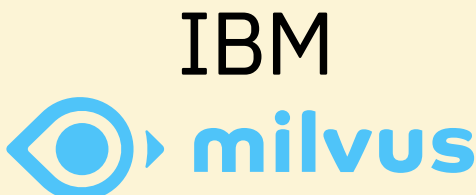



Vector database

Quick view of competition

Vector indexes supported

Dedicated vector databases

 Single index type supported	 Two index types supported	 More than two index types supported
<div>Pinecone</div> <div>Vald</div> <div>marqo</div> <div>Chroma</div>	<div>vespa</div> <div>drant</div> <div>LanceDB</div>	<div>IBM milvus</div> <div>Weaviate</div>

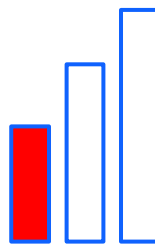
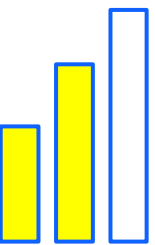
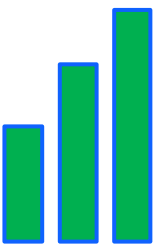






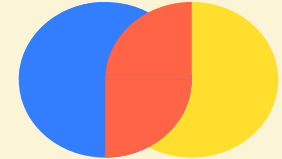


Worst  Best

Vector database

Quick view of competition

Community popularity and awareness

Dedicated vector databases

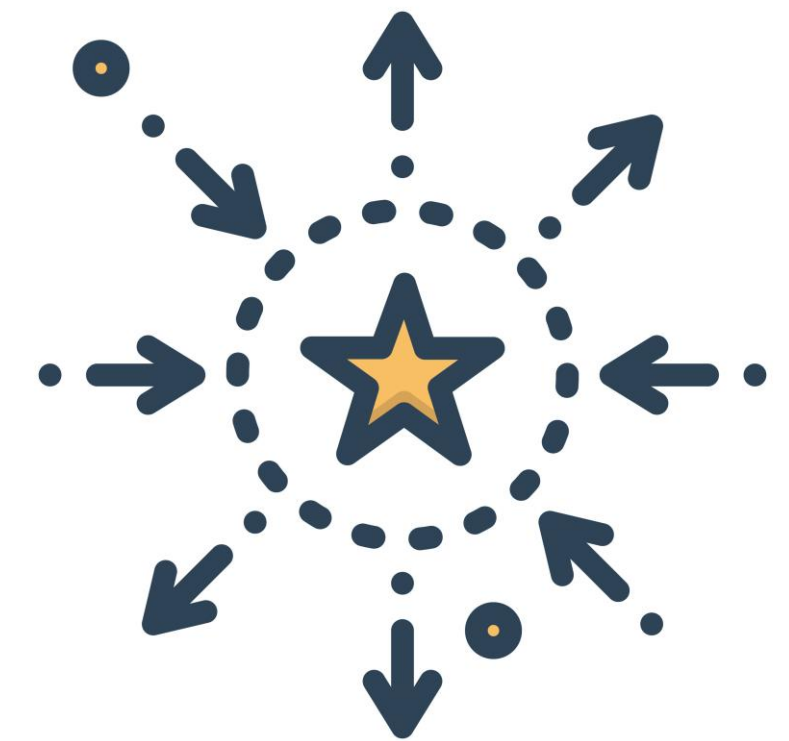
 Proprietary, no GitHub star rating	 GitHub star rating of 10K or less	 GitHub star rating of greater than 10K
 Pinecone	<div><div> 4K stars marqo</div><div> 2.4K stars LanceDB</div><div> 1.4K stars Vald</div></div> <div><div> 9K stars Weaviate</div><div> 5.2K stars vespa</div></div>	<div><div> 11.3K stars Chroma</div><div> 25.8K stars milvus</div><div> 16.5K stars drant</div></div>



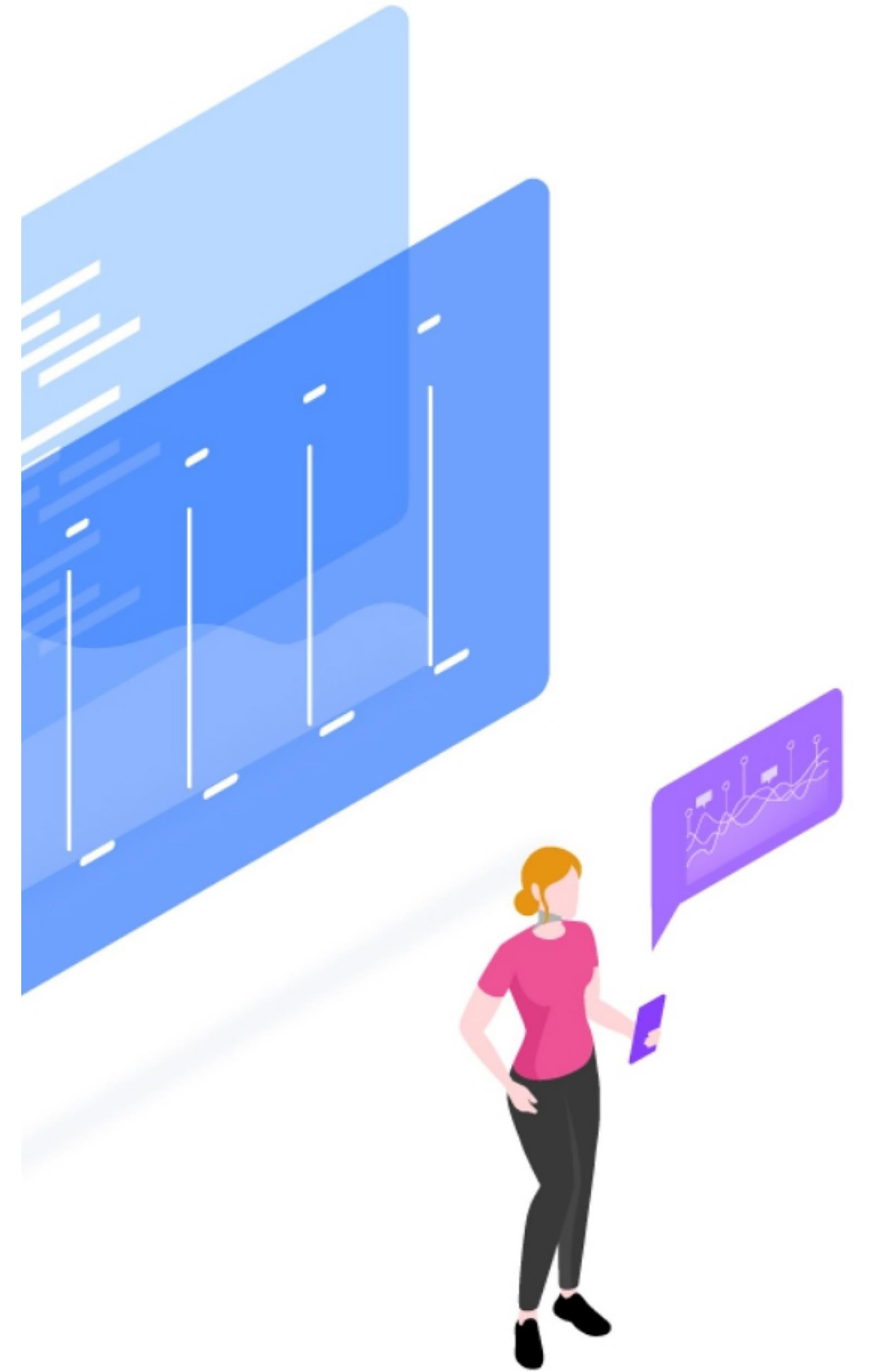
Milvus

Differentiators

- Milvus is tightly integrated with the IBM watsonx platform that includes a data lakehouse, AI models, and governance
- Milvus is the most popular vector database and has ~2x the GitHub star rating of the next closest dedicated vector database
- Milvus provides the most vector index types to accommodate different use cases and data volumes
- Milvus provides vector scalability above billion scale to ensure future requirements are met
- Milvus provides compute flexibility from a single node to a cluster to accommodate different dataset sizes and use cases
- IBM watsonx delivers a hybrid cloud platform for AI and ML workloads



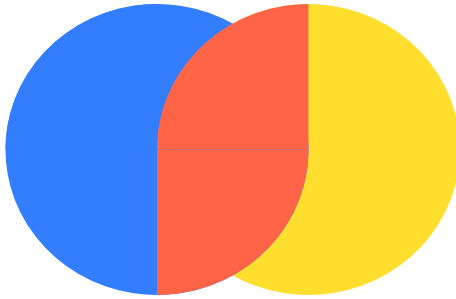

Background information



Vector database

Competitors

Background (1 of 8)

Competitor	Background information
<div>Chroma</div>	<ul style="list-style-type: none">Chroma is a company developing an open source embedding database (or vector database), also known as Chroma. The database aims to simplify the development of large language model (LLM) apps by making knowledge, facts, and skills pluggable for LLMs. Chroma offers tools to store embeddings and their metadata, embed documents and queries, and search embeddings. Chroma consists of a Python client SDK, JavaScript/TypeScript client SDK and a server application. In Python, Chroma can run in memory or in client/server mode. In JavaScript, Chroma runs in client/server mode and talks to a Python back end.The Chroma project is coordinated by a small team of full-time employees based in Potrero Hill. The company was cofounded by Jeff Huber and Anton Troynikov in April 2022. The Chroma database is released under an Apache 2.0 license.
<div>LanceDB</div>	<ul style="list-style-type: none">LanceDB is a new open source vector database that can support low-latency billion-scale vector search on a single node. Built around a new columnar data format, LanceDB makes it incredibly easy to build applications for generative AI, recommender systems, search engines, content moderation, and more.LanceDB is a developer-friendly, open source vector database for multi-modal AI with zero management overhead. Installs in seconds and scales to billions of embeddings at a fraction of the cost of other vector databases. You can even stream data directly from object storage for training or fine-tuning.

Vector database

Competitors

Background (2 of 8)



Competitor

Marqo

Background information

- Marqo is an open source tensor search framework that powers end user search, information retrieval, and machine learning applications.
- Most search experiences in websites and applications are built on lexical search. For example, when you search (“blue shirt”), it matches results that contain the words “blue” and “shirt”. This is problematic because end users usually interact with language in human ways — if the user makes a typo, searches using a synonym or phrases their query as a question they are unlikely to retrieve relevant results. This is a long-known short-coming of lexical search and where tensor search can help. Marqo uses a tensor based search and does not have this shortcoming.
- Tensors allow us to use neural networks to structure documents, images and other data in such a way that it can be searched with human-like understanding. Using Marqo, developers can build and deploy tensor search experiences in a few lines of code. It’s designed for the cloud, horizontally scalable, and provides a query DSL language for efficiently filtering results.

Pinecone





- Pinecone is a cloud-native vector database that handles high-dimensional vector data. The core underlying approach for Pinecone is based on the ANN search that efficiently locates faster matches and ranks them within a large dataset.
- Pinecone provides ultra-low query latency, even with billions of items. This means that users can search large datasets. Pinecone indexes are updated in real-time, so users always have access to the most up-to-date information.
- Pinecone allows clients to combine vector search with metadata filters to get more relevant and faster results. For example, a filter by product category, price, or customer rating.

Vector database

Competitors



Background (3 of 8)

Competitor	Background information
<div></div> <div>Qdrant</div>	<ul style="list-style-type: none">• Qdrant is a vector database & vector similarity search engine. It deploys as an API service providing search for the nearest high-dimensional vectors. With Qdrant, embeddings or neural network encoders can be turned into full-fledged applications for matching, searching, recommending, and other use cases.• Easy to use API provides the OpenAPI v3 specification to generate a client library in almost any programming language.• Fast and accurate by implementing a unique custom modification of the Hierarchical Navigable Small Worlds (HNSW) algorithm for ANN search. Search with state-of-the-art speed and apply search filters without compromising on results.• Effectively utilizes your resources. Developed entirely in Rust language, Qdrant implements dynamic query planning and payload data indexing. Hardware-aware builds are also available for Enterprises.
<div></div> <div>Vald</div>	<ul style="list-style-type: none">• Vald is a highly scalable distributed fast approximate nearest neighbor dense vector search engine.• Vald is designed and implemented based on the Cloud-Native architecture. It uses the fastest ANN algorithm Neighborhood Graphs and Trees (NGT) to search neighbors. Vald has automatic vector indexing and index backup, and horizontal scaling which made for searching from billions of feature vector data. Vald is easy to use, feature-rich and highly customizable.• Originally developed for a project within Yahoo! Japan.

Vector database

Competitors

Background (4 of 8)

Competitor	Background information
<div>Vespa</div> <div></div>	<ul style="list-style-type: none">• Vespa is a full-featured search engine with full support for traditional information retrieval as well as modern vector embedding based techniques. And since Vespa allows these approaches to be combined efficiently in the same query and ranking model, you can create hybrid solutions that combines the best of both.• Recommendation, content personalization, and ad targeting is all the same thing when it comes to implementation. For a given user or context, evaluate machine-learned content recommender models to find the best items and show them to the user. Vespa makes it possible to do the whole process online when the recommendation is needed, which ensures recommendations are up-to-date and makes it affordable to make them specifically for each user or situation.
<div>Weaviate</div> <div></div>	<ul style="list-style-type: none">• Weaviate is an open source vector database that stores both objects and vectors. This allows for combining vector search with structured data filtering.• Weaviate is a low-latency vector database with out-of-the-box support for different media types (text, images, etc.). It offers Semantic Search, Question-Answer Extraction, Classification, Customizable Models (PyTorch/TensorFlow/Keras), etc. Built from scratch in Go, Weaviate stores both objects and vectors, allowing for combining vector search with structured filtering and the fault tolerance of a cloud-native database. It is all accessible through GraphQL, REST, and various client-side programming languages.

Competitors

Background
(5 of 8)



Competitor

Cassandra

Background information

- Cassandra is an open source NoSQL distributed database that manages large amounts of data across commodity servers. It is a decentralized, scalable storage system designed to handle vast volumes of data across multiple commodity servers, providing high availability without a single point of failure.
- Cassandra was created for Facebook but was open sourced and released to become an Apache project (maintained by the Americal non-profit, Apache Software Foundation) in 2008. After that, it found top priority in 2010 and is now among the best NoSQL database systems in the world. Cassandra is trusted and used by thousands of companies because of the ease of expansion and, better still, its lack of a single point of failure. Currently, the solution has been deployed to handle databases for Netflix, Twitter, Reddit, and others.
- Vector Search is a new feature added to Cassandra 5.0. It is a powerful technique for finding relevant content within large datasets and is particularly useful for AI applications.

ClickHouse



- ClickHouse is a highly scalable open source DBMS that uses a column-oriented structure. It's designed for OLAP and is highly performant. ClickHouse can return processed results in real time in a fraction of a second. This makes it ideal for applications working with massive structured data sets: data analytics, complex data reports, data science computations, and others.
- The main advantages of using ClickHouse for vector search compared to using more specialized vector databases include using ClickHouse's filtering and full-text search capabilities to refine your dataset before performing a search.

Competitors


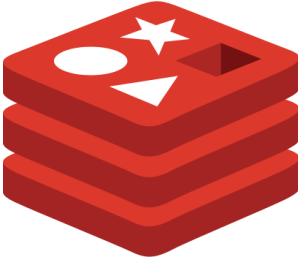
Background
(6 of 8)

Competitor	Background information
Elasticsearch	<ul style="list-style-type: none">Elasticsearch is the distributed search and analytics engine at the heart of the Elastic Stack. Elasticsearch is where the indexing, search, and analysis happens.Elasticsearch provides near real-time search and analytics for all types of data. Whether you have structured or unstructured text, numerical data, or geospatial data, Elasticsearch can efficiently store and index it in a way that supports fast searches. Go beyond simple data retrieval and aggregate information to discover trends and patterns in your data. And as your data and query volume grows, the distributed nature of Elasticsearch enables your deployment to grow seamlessly right along with it.Vector search provides the foundation for implementing semantic search for text or similarity search for images, videos, or audio.
OpenSearch	<ul style="list-style-type: none">OpenSearch is the flexible, scalable, open source way to build solutions for data-intensive applications. Explore, enrich, and visualize your data with built-in performance, developer-friendly tools, and powerful integrations for machine learning, data processing, and more.Using OpenSearch as a vector database brings together the power of traditional search, analytics, and vector search in one complete package. OpenSearch’s vector database capabilities can accelerate artificial intelligence (AI) application development by reducing the effort for builders to operationalize, manage, and integrate AI-generated assets. Bring your models, vectors, and metadata into OpenSearch to power vector, lexical, and hybrid search and analytics, with performance and scalability built in.





Competitors

Background
(7 of 8)

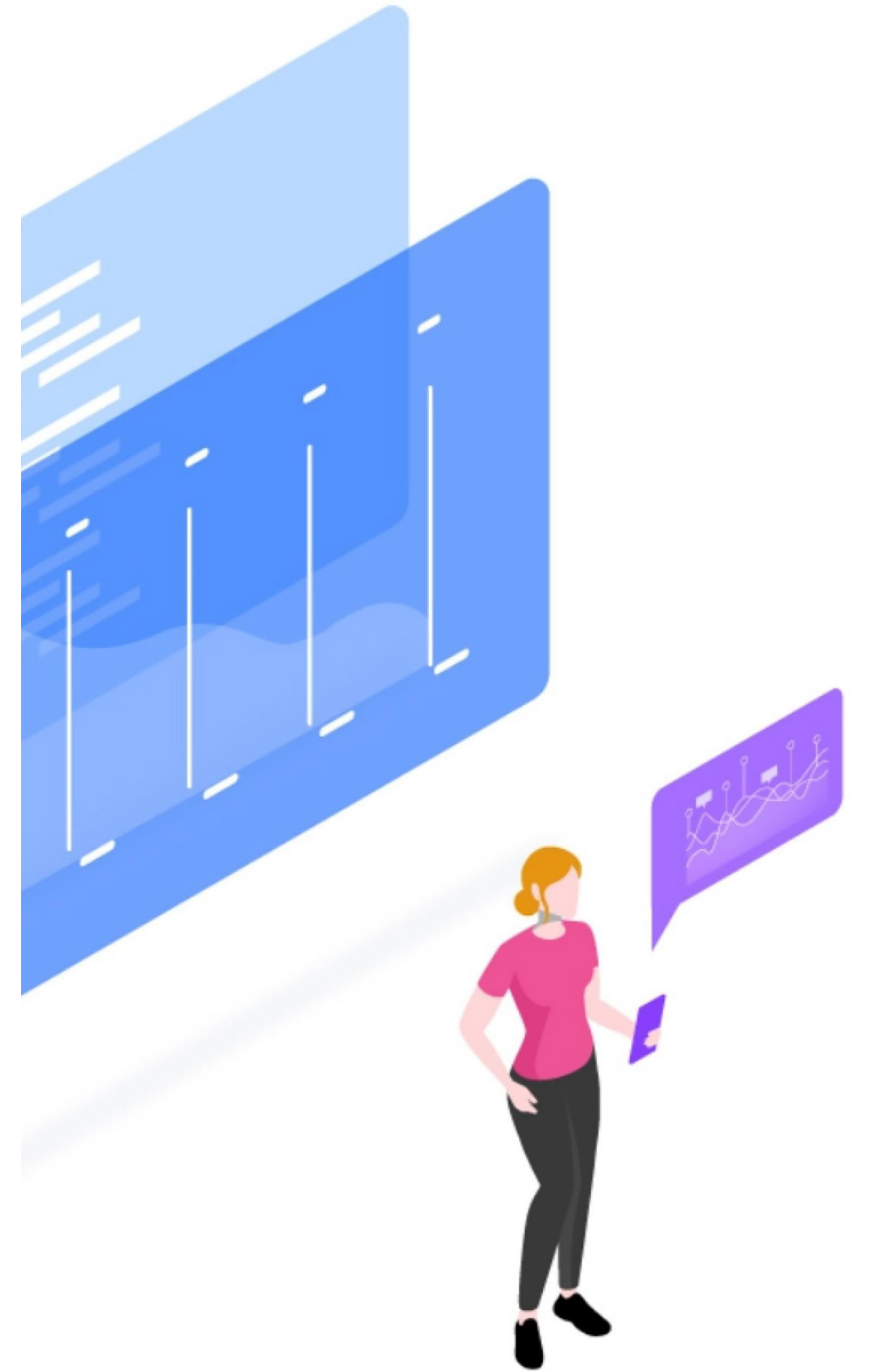
Competitor	Background information
<div>PostgreSQL</div> <div></div> <div>PostgreSQL</div>	<ul style="list-style-type: none">• PostgreSQL is a powerful, open source object-relational database system with over 35 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.• pgvector is an open source vector database extension for PostgreSQL. It supports exact and approximate nearest neighbor search; L2 distance, inner product, and cosine distance; and any language with a Postgres client.
<div>Redis</div> <div> redis</div>	<ul style="list-style-type: none">• Redis is an open source (BSD licensed), in-memory data structure store used as a database, cache, message broker, and streaming engine. Redis provides data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs, geospatial indexes, and streams. Redis has built-in replication, Lua scripting, LRU eviction, transactions, and different levels of on-disk persistence, and provides high availability via Redis Sentinel and automatic partitioning with Redis Cluster.• Redis Enterprise manages vectors in an index data structure to enable intelligent similarity search that balances search speed and search quality. Choose from two popular techniques, FLAT (a brute force approach and is not an acronym) and HNSW ((Hierarchical Navigable Small World) a faster, and approximate approach), based on your data and use cases.

Competitors

Background
(8 of 8)

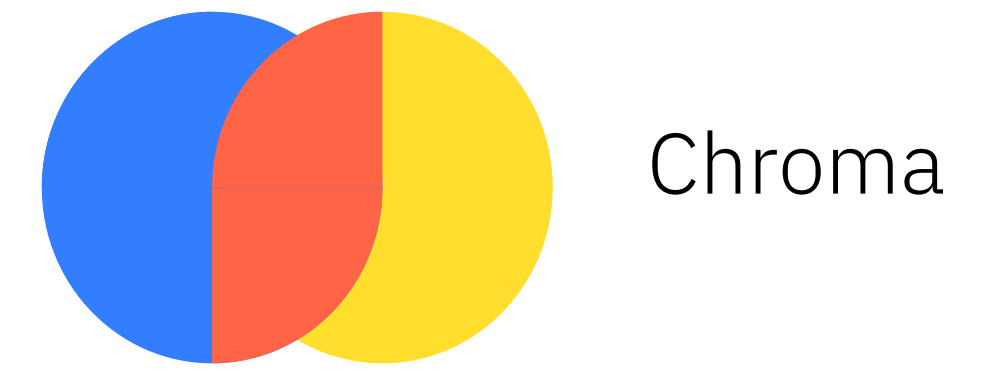
Competitor	Background information
<div>Rockset</div> <div></div>	<ul style="list-style-type: none">• Rockset is a real-time search and analytics database designed to serve millisecond-latency analytical queries on event streams, CDC streams, and vectors.• Create vector embeddings using any machine learning model (Hugging Face, OpenAI, Cohere, etc.) and index them for fast similarity search. Build with LangChain or LlamaIndex. Use Rockset for retrieval augmented generation (RAG), personalization engines, semantic search, anomaly detection and more.
<div>SingleStore</div> <div></div>	<ul style="list-style-type: none">• SingleStoreDB is a distributed SQL database that offers high-throughput transactions (inserts and upserts), low-latency analytics and context from real-time vector data.• SingleStoreDB meets you wherever you are in your cloud journey — providing the flexibility to deploy wherever you need: self-managed on-premises, or as a fully managed cloud.• SingleStore supports vector database processing, which allows you to store and search vector data. A typical vector search locates the set of vectors that most closely match a query vector. Vectors usually come from objects: text, images, video, audio, etc. Vector database searches find data based on its content or meaning, even without exact matches. For example, vector search can allow a semantic search of text, where a query about "meals" could return information about "lunch" and "dinner" without using those words because they are similar in meaning.• SingleStore supports a native vector data type and indexed approximate-nearest-neighbor (ANN) search that provide high-performance vector search and easier building of vector-based applications.

Strengths & weaknesses



Strengths & weaknesses

(1 of 8)



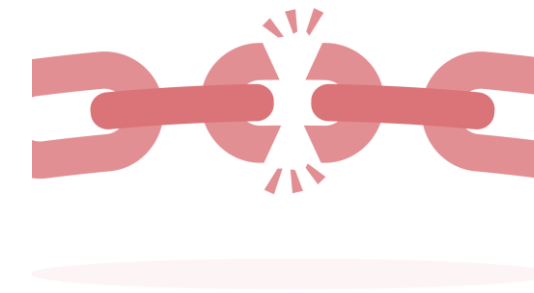
Flexible query capabilities including complex range searches and combinations of vector attributes



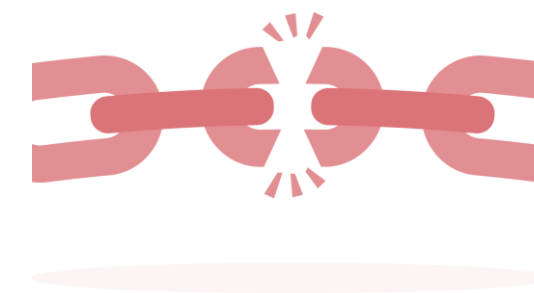
Excels with audio data making it ideal for audio-based searches, music recommendation applications, and other sound-based applications



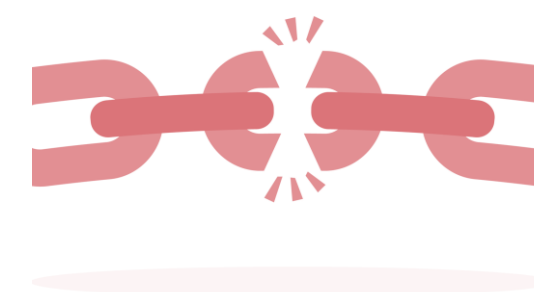
11.2K GitHub stars indicating popularity within the open source community



Limited to 1 million maximum vectors



Single node database, cannot scale beyond a single compute node



Only a single vector index type supported, HNSW

Strengths & weaknesses

(2 of 8)



LanceDB



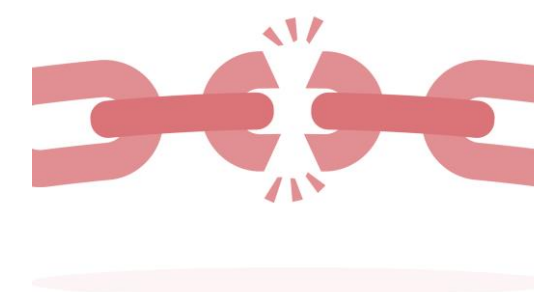
Provides two vector index types, IVF-PQ and DiskANN



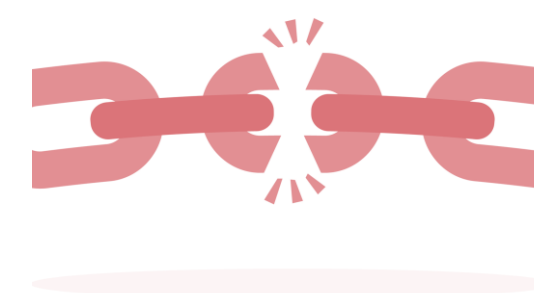
Is 100x faster than Parquet queries and written entirely in Rust for performance



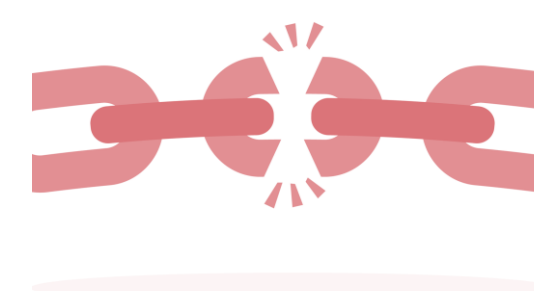
Provides self-managed, fully managed, and embedded deployment options



Only 2.3K GitHub stars indicating a small popularity within the open source community and low usage



Development and product direction controlled primarily by LanceDB



Does not have native Amazon S3 support, requires the use of a separate SDK for S3

Dedicated vector databases

IBM and Business Partner – Internal Use Only

Strengths & weaknesses

(3 of 8)



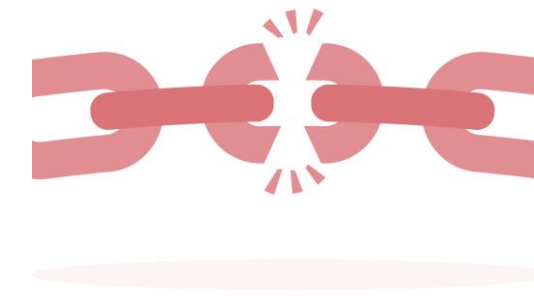
Easy to get started and implement, minutes instead of months



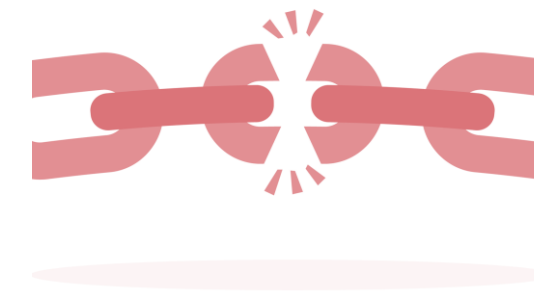
Can combine multiple data types for multimodal search capabilities such as audio, video, text, and images



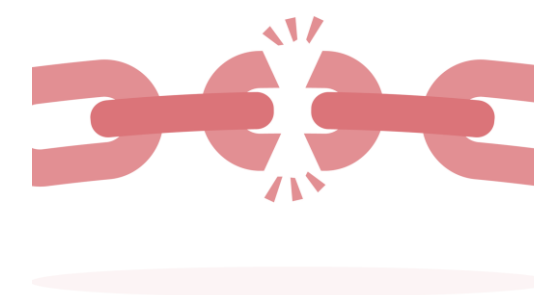
Has a fully managed solution on Marqo Cloud that is optimized for Marqo



Only 4K GitHub stars indicating a lower popularity and less adoption than most open source offerings



Development and product direction controlled primarily by Marqo



Has a single vector index type, tensor indexes

Strengths & weaknesses

(4 of 8)



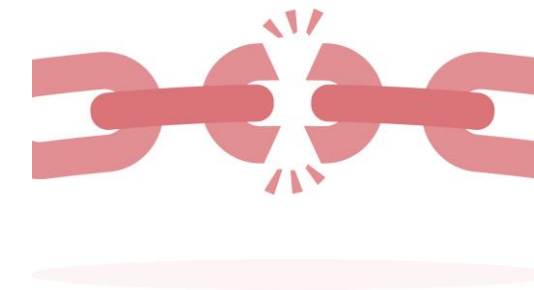
Real-time search capabilities



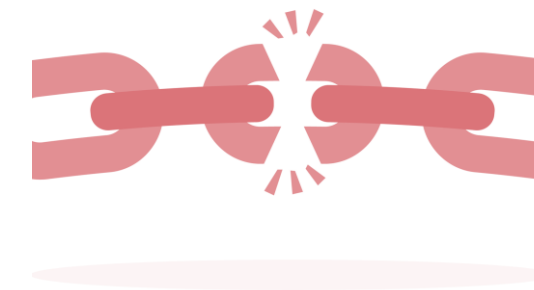
Automatic indexing, clients do not have to know vectorization or vector indexes



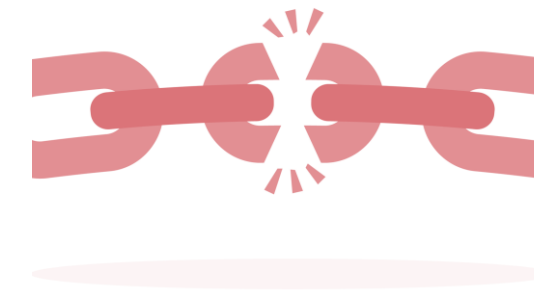
Offers a serverless compute environment where clients pay only for usage



Proprietary, the only dedicated vector database that is not open source



Uses proprietary composite vector indexing, clients have no visibility into algorithms



Difficult for clients to understand total cost until after real workloads are executed

Strengths & weaknesses

(5 of 8)



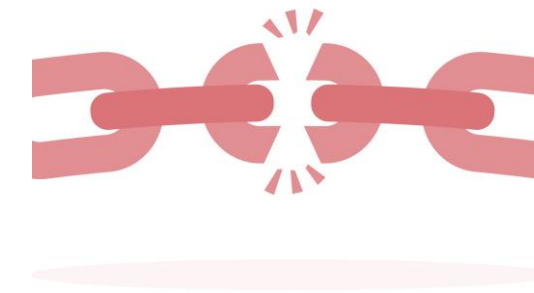
Has a 16.5K GitHub star rating indicating sizable popularity and usage



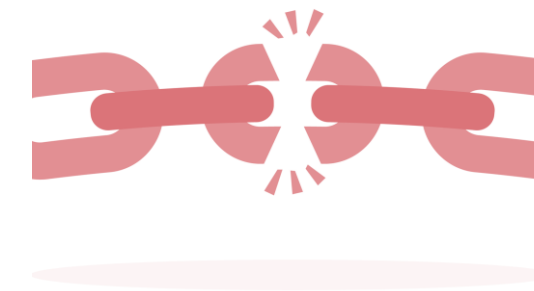
Provides billion scale vector support to handle large vector environments for AI use cases



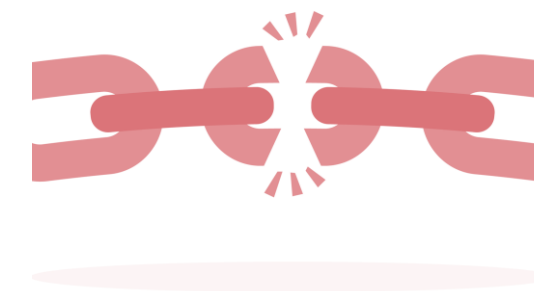
Released single node benchmark numbers indicating the highest RPS results



Static sharding which requires data re-sharding when new nodes are added



Only a single vector index type, HNSW



Playing catch-up with Milvus and other vector databases for query UI

Strengths & weaknesses

(6 of 8)



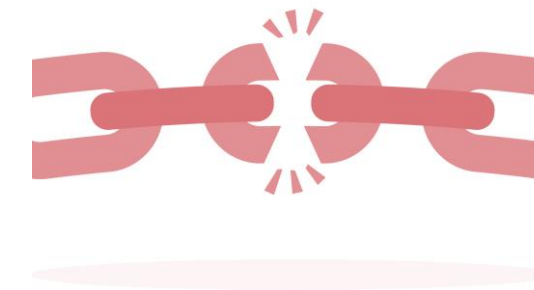
Open source and free to use with any Kubernetes deployment minimizing cost



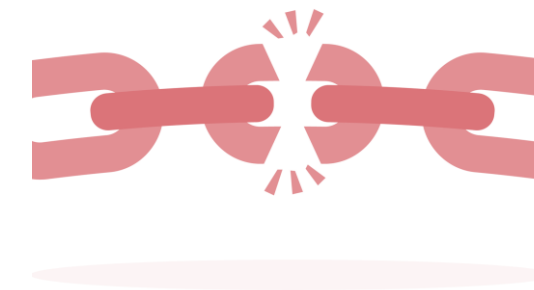
Uses the Neighborhood Graphs and Trees (NGT) deep learning library as the vector search engine



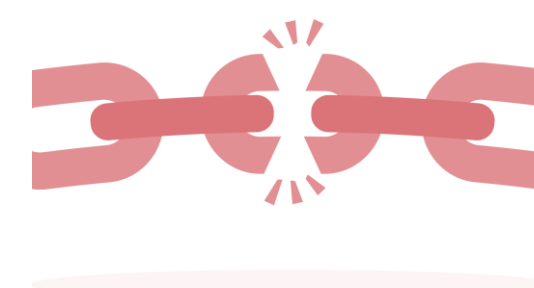
Offers Software Development Kits (SDKs) for Golang (Go), Java, NodeJS, and Python



A 1.4K GitHub star rating indicating low popularity and awareness



Focus is on the original Yahoo Japan use case versus wider use case testing & deployment



Development and product direction is primarily controlled by developers within Yahoo Japan

Strengths & weaknesses

(7 of 8)



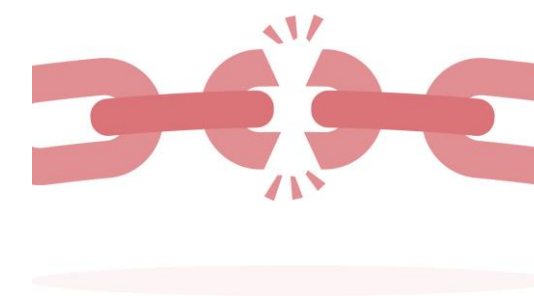
One of the earliest dedicated vector databases made available (2017)



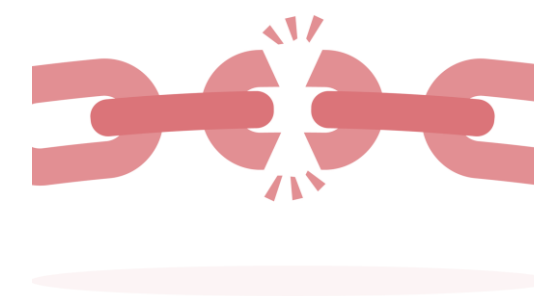
Offers self-managed and fully managed offerings



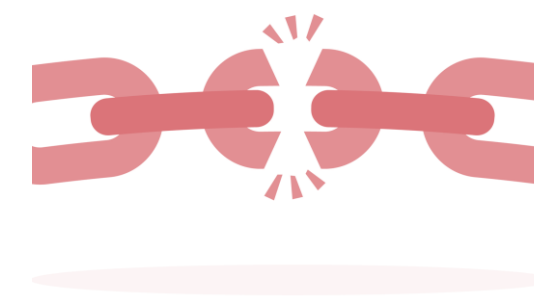
Offers two vector index types, HNSW and hybrid BM25



A 5.2K GitHub star rating which indicates lower popularity and community adoption rate



Software development and product direction primarily controlled by Yahoo



Vespa Cloud Enclave only supports AWS and GCP, no Azure support

Strengths & weaknesses

(8 of 8)



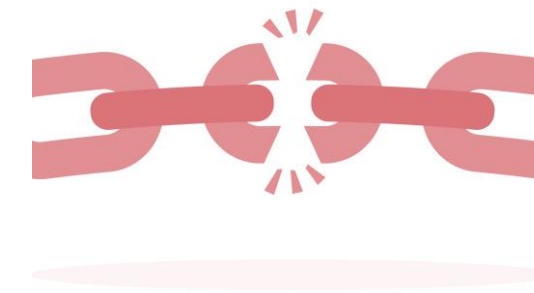
Stores both objects and vectors within the database



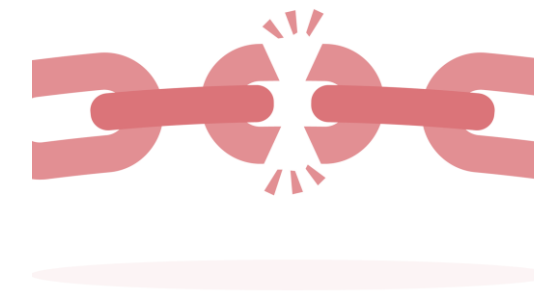
Excellent documentation with emphasis on good developer experience



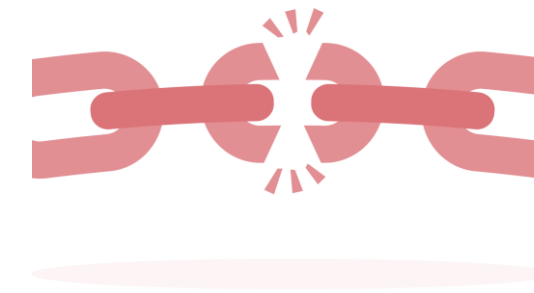
Offers both keyword and vector search functionality



Has 9K GitHub star rating which indicates lower community popularity

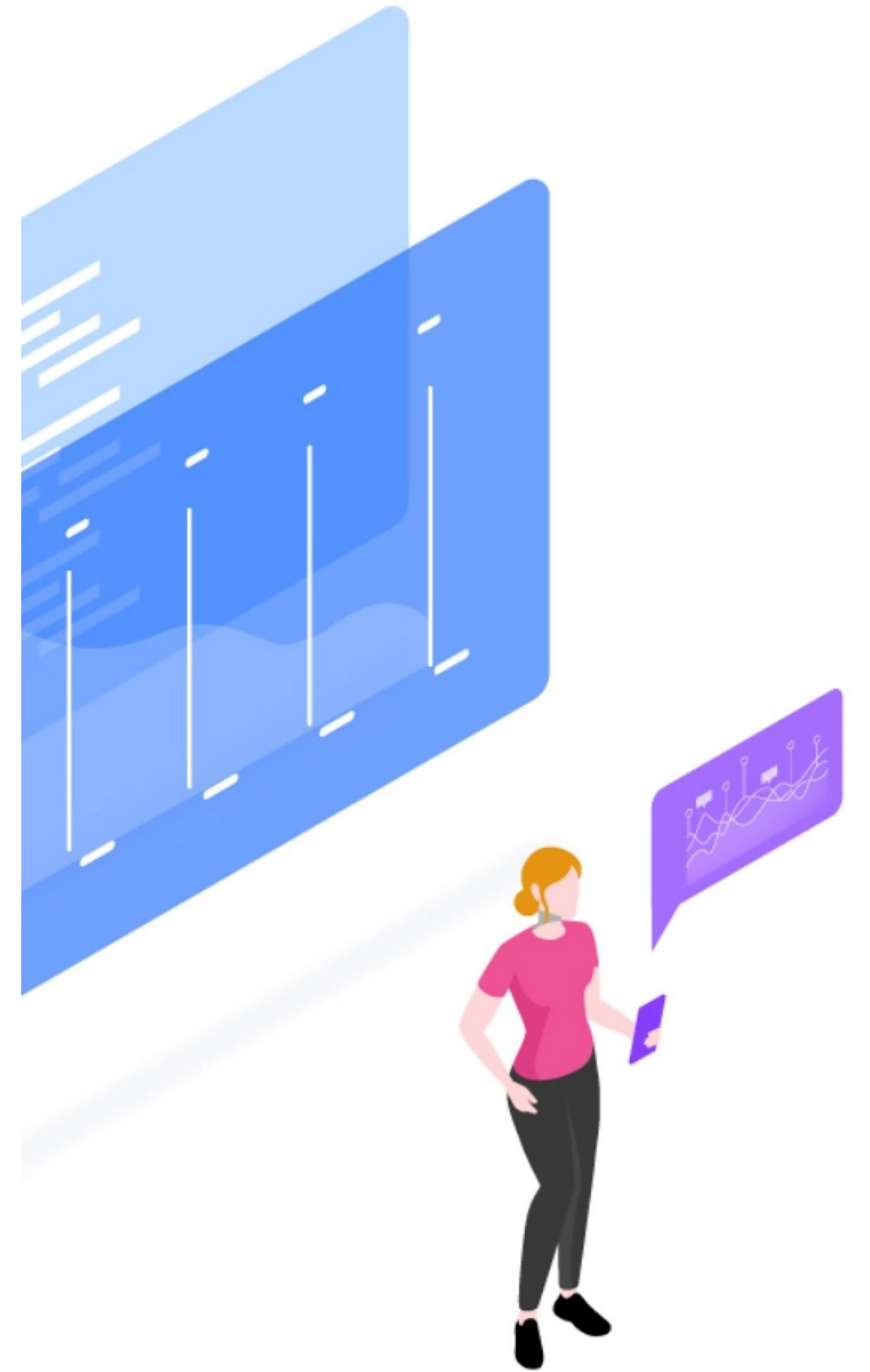


Requires larger compute infrastructure for large datasets than others

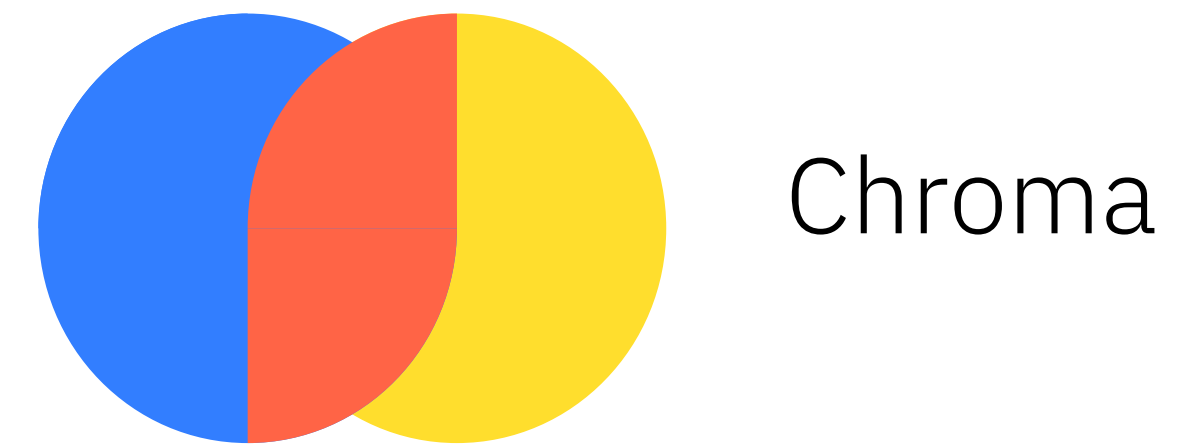


Cost implication of fully managed Weaviate for larger compute environments is unknown

Objection handling



Objection handling against Chroma



Objection

Chroma uses the Hierarchical Navigable Small Worlds (HNSW) vector index, which is recognized as one of the most performant vector indexes and is optimal for most of our use cases

IBM response

- Although the HNSW vector index is a good choice for many approximate nearest neighbor similarity searches, there are situations where other vector indexes may be better suited
- IBM Milvus supports the HNSW vector index and four other vector indexes
- For situations where HNSW may not be ideal, IBM Milvus offers alternative vector indexes to allow use case customization
 - For example, a small dataset where exact nearest neighbor results are preferred

Competitive objections

Objection handling against LanceDB



Objection

LanceDB provides the Inverted File Product Quantization (IVF-PQ) vector index, which matches our vector search workload requirements

IBM response

- IBM Milvus allows the IVF-PQ to be used as the vector index type as well as 4 other vector indexes to provide flexibility in vector search workloads that have characteristics that may benefit from using another vector index
- Vector search characteristics and new datasets may favor alternative vector index selection & this is possible with IBM Milvus

Competitive objections

Objection handling against Marqo



Objection

Marqo provides tensor indexes, which should be ideal for the intended workload being selected for the initial vector search

IBM response

- Support for a single vector index type is ok providing the intended workload, use case, and datasets never change
- IBM Milvus provides a choice of vector indexes to accommodate different workload characteristics and datasets
- Will all vector searches use the same dataset and vector query characteristics?

Competitive objections

Objection handling against Pinecone



Objection

Pinecone is a popular vector database and is a good choice to meet our vector database requirements for our initial use cases

IBM response

- IBM Milvus is an open source offering that provides several vector index choices to accommodate different query characteristics and datasets
- Pinecone is a proprietary vector database with a single proprietary composite vector index
- Pinecone is only available as a fully managed cloud offering and does not offer hybrid cloud support

Competitive objections

Objection handling against Qdrant



Objection

Qdrant is a popular open source vector database and meets our immediate requirements for a vector database. What advantages does IBM Milvus have over Qdrant?

IBM response

- Although Qdrant is a popular vector database, Milvus has more than 1.5x the GitHub stars of Qdrant making it the most popular open source vector database
- Qdrant offers a single vector index type, Hierarchical Navigable Small Worlds (HNSW) versus the 5 vector index types offered by IBM Milvus
- Qdrant uses static sharding compared to Milvus and its dynamic segment placement, static sharding requires data re-sharding as nodes are added

Competitive objections

Objection handling against Vald



Objection

Vald is an open source vector database that meets our requirements, what advantages does Milvus provide over Vald?

IBM response

- Vald has low visibility and popularity in the open source community with a GitHub star rating of 1.4K compared to the 25.8K star rating of Milvus
- Vald has a single vector index based on the Neighborhood Graphs and Trees (NGT) library compared to the 5 vector index types provided by IBM Milvus
- Vald has a limited set of use cases that have been used in production compared to a broader set of use cases with Milvus

Competitive objections

Objection handling against Vespa



Objection

Vespa is a good open source vector database and has been in existence since 2017, why would Milvus be a better vector database choice?

IBM response

- Although Vespa has been in existence the longest of the dedicated vector databases, the Vespa GitHub star rating is 5.2K, very low for the length of time it has been available
- Milvus has been available since 2019 and has a GitHub star rating of 25.8K or almost 5x Vespa's rating indicating broader community support and awareness of the solution
- Vespa provides some flexibility by providing 2 vector index types, IBM Milvus provides 5 vector index types

Competitive objections

Objection handling against Weaviate



Objection

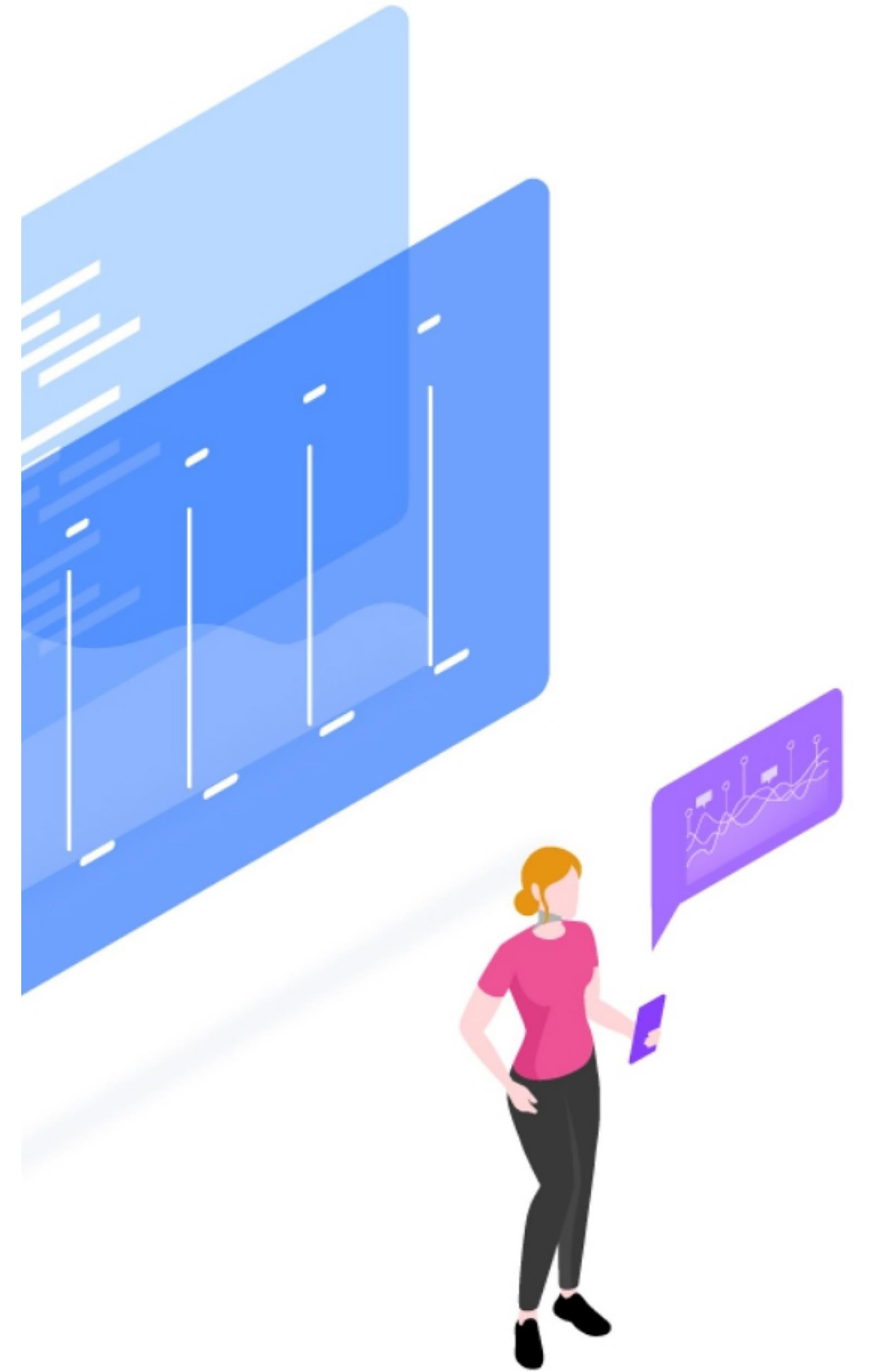
Weaviate provides the vector database capabilities that we require and can scale out to meet future growth, what advantages does Milvus provide over Weaviate?

IBM response

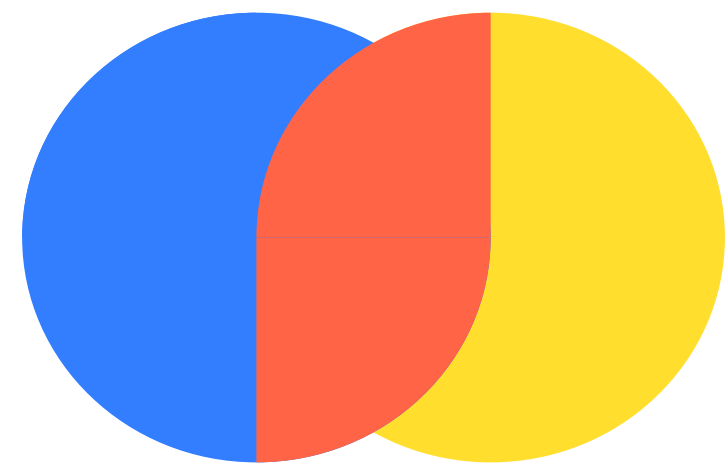
- IBM Milvus can scale to accommodate future growth comparable to Weaviate
- IBM Milvus as part of the IBM watsonx platform provides an integrated AI environment that Weaviate cannot match
- IBM Milvus provides 5 vector index types compared to Weaviate's 3 vector index types provides more flexibility for query optimization
- Milvus has over 2x the GitHub star rating of Weaviate indicating wider awareness and popularity in the open source community

Competitive objections

Setting traps



Setting traps against Chroma



Chroma

Trap to set/question to ask

Ask the client if they will ever need to use a dataset larger than 1 million vectors.

Reason

Chroma has a limit of 1 million vectors for processing.

Trap to set/question to ask

Ask the client if they see their vector database growing beyond a single node in size.

Reason

Chroma is limited to a single database node.

Setting traps against LanceDB



LanceDB

Trap to set/question to ask

Ask the client if they will have a variety of use cases and datasets that will be used with a vector database.

Reason

LanceDB only provides support for two vector index types. If a client will be using multiple use cases and datasets with a vector database, a vector database like IBM Milvus with 5 vector index types provides flexibility in optimizing the vector database for different use cases.

Trap to set/question to ask

Ask the client if they will be using AWS and S3 with their vector database.

Reason

LanceDB has no native support for Amazon S3 files requiring that clients use a separate Software Development Kit (SDK) such as AWS Boto3.

Setting traps against Marqo



Trap to set/question to ask

Ask the client if they will have a variety of use cases and datasets that will be used with a vector database.

Reason

Marqo provides a single vector index type (Tensor) that provides no optimization possibility for workloads or use cases that are not suited for this index type.

Trap to set/question to ask

Ask the client if they want a vector database that has a strong community of contributors to ensure a robust vector database over time incorporating new features.

Reason

Marqo is primarily controlled, & product direction determined by Marqo. Milvus has a strong community of contributors including IBM, Walmart, AT&T, and others.

Setting traps against Pinecone



Trap to set/question to ask

Ask the client if they want to use a vector database where the vector indexes and other internal details are closed & not accessible to them.

Reason

Pinecone is a proprietary, closed vector database that uses a proprietary composite vector index with details unknown.

Trap to set/question to ask

Ask the client if their vector database requirements include hybrid cloud and self-managed deployment options.

Reason

Pinecone is only available as a fully managed cloud solution with has no hybrid cloud or self-managed deployment options. Milvus offers a variety of deployment options and supports a hybrid cloud environment.

Setting traps against Qdrant



Trap to set/question to ask

Ask the client if they plan on their vector database cluster growing over time with more nodes being added.

Reason

Qdrant has static sharding, which means that the data must be re-sharded when a new node is added. Milvus has dynamic segment placement, which does not require data movement when nodes are added.

Trap to set/question to ask

Ask the client if they will have a variety of use cases and datasets that will be used with a vector database.

Reason

Qdrant provides a single vector index type (HNSW) that provides no optimization possibility for workloads or use cases that are not suited for this index type.

Setting traps against Vald



Trap to set/question to ask

Ask the client if they want to choose a vector database that has a small community and has only been implemented across a small set of use cases.

Reason

Vald has a GitHub star rating of 1.4K and has primarily been used with use cases around its original implementation within Yahoo Japan.

Trap to set/question to ask

Ask the client if they will have a variety of use cases and datasets that will be used with a vector database.

Reason

Vald provides a single vector index type (NGT) that provides no optimization possibility for workloads or use cases that are not suited for this index type.

Setting traps against Vespa



Trap to set/question to ask

Ask the client if they want to choose a vector database that has a small community and has been focused on a subset of use cases.

Reason

Vespa has a GitHub star rating of 5.2K and has been used with use cases primarily around text searches.

Trap to set/question to ask

Ask the client if they will have a variety of use cases and datasets that will be used with a vector database.

Reason

Vespa only provides two vector index types (HNSW and BM25) compared to the five vector indexes provided by IBM Milvus. More vector index types provide flexibility in optimizing query performance for use cases that not be ideal for a default vector index type.

Setting traps against Weaviate



Trap to set/question to ask

Ask the client if they want a vector database that has a strong community of contributors to ensure a robust vector database over time incorporating new features.

Reason

Weaviate is primarily controlled, and product direction determined by Weaviate. Milvus has a strong community of contributors including IBM, Walmart, AT&T, and others.

Trap to set/question to ask

Ask the client if they plan on their vector database cluster growing over time with more nodes being added.

Reason

Weaviate has static sharding, which means that the data must be re-shared when a new node is added. Milvus has dynamic segment placement, which does not require data movement when nodes are added.

Summary

IBM Milvus provides these differentiators for client adoption of a vector database

- Large choice of vector index types for workload optimization and increased performance
- Ability to scale from a single node to a cluster
- Tight integration with the IBM watsonx platform for a complete AI environment
- Most popular open source vector database based on GitHub star rating



- The selection of open source vector databases available today means a proprietary vector database solution is unnecessary and expensive
- Milvus is integrated in watsonx.data and the overall IBM watsonx platform for AI
- Flexibility in choosing vector indexes maximizes the ability to achieve query optimization for each use case and workload

Additional references

(1 of 2)

Technical information about vector databases

- [Article on what makes vector databases different from each other](#)
- [Blog about Hierarchical Navigable Small Worlds \(HNSW\) vector indexes](#)
- [DataStax article on HNSW vector indexes](#)
- [Technical paper on vector database systems](#)
- [General article on the vector database landscape](#)

Additional references

(2 of 2)

Technical information about vector databases

- [Article comparing many of the vector databases](#)
- [Article providing an honest comparison of open source databases](#)
- [Blog on vector database comparisons](#)
- [Zilliz comparative article on various vector databases](#)

NOTE: Zilliz is the commercial version of Milvus

