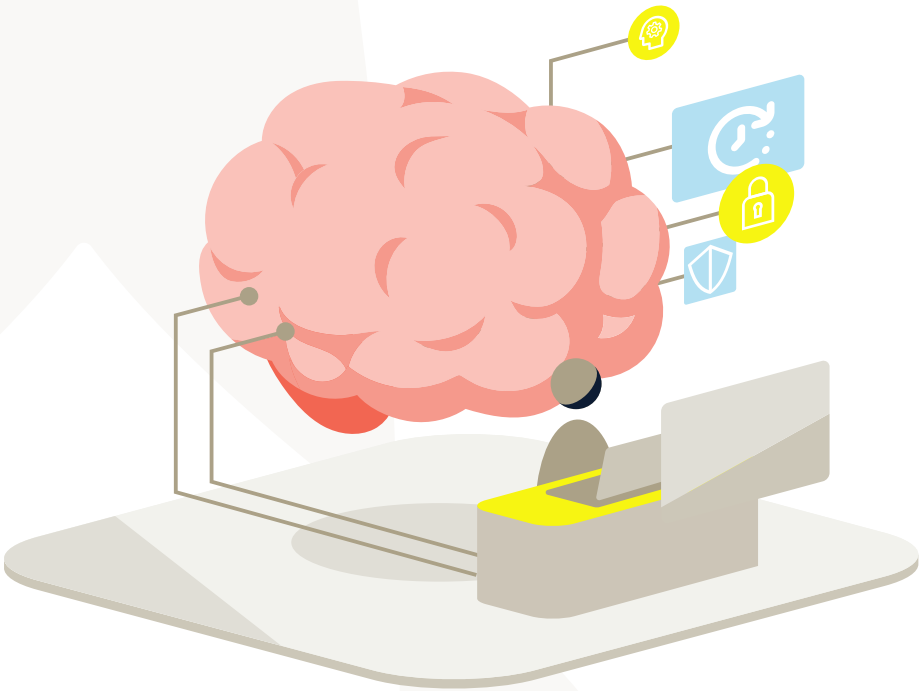
# An insider's guide to AI databases

Aerospike

Infinitely possible.™



# Introduction

Welcome to "An insider's guide to AI databases," an exploration into the crucial role databases play in the development and deployment of AI applications.

Since ChatGPT's explosive debut in November 2022, AI application development has become a top priority for most software organizations. This, in turn, has created a great deal of interest in AI databases, which must support new data types (e.g., vectors), query patterns (e.g., similarity search), and increasingly, combinations of different data types and query patterns to increase AI accuracy and relevance.

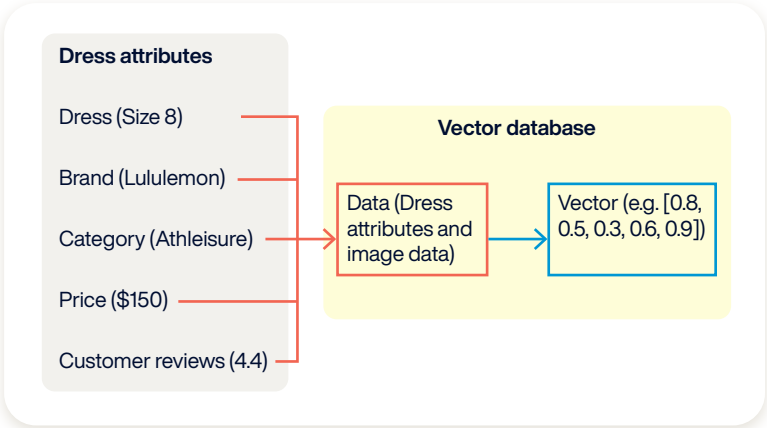
There are many database alternatives available to AI developers today. This paper aims to demystify the complex landscape of AI databases to help you match the right database with your AI requirements. It will explain:

- AI data management fundamentals
- A framework for comparing AI database alternatives
- The pros and cons of popular AI database alternatives
- Practical strategies to help you succeed with AI databases to drive innovation within your company and market

# 1 An introduction to AI data

## Vector data and vector search

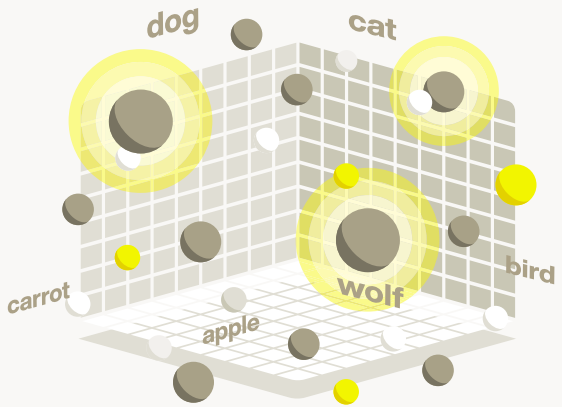
At the heart of AI data management are vector data and vector search. Vectors are the currency of AI applications. They are generated by AI **embedding models** and represent the essence of a piece of information, such as the meaning of some text, the appearance of an image, or the attributes of a product on an e-commerce website.




**Vectors** are arrays of numbers; the numbers are coordinates in an n-dimensional space. The number of dimensions depends on the embedding model that generated the vector. A large language model (LLM) like GPT contains thousands of dimensions and can encode significant semantic meanings.

Other specialized models can include fewer dimensions and be optimized for specific properties or measurements of your data.

**Vector search** finds the vectors located near each other in n-dimensional space. Vectors located near one another represent things that are similar in meaning or appearance. Similarity search, semantic search, proximity search, and nearest neighbor search are sometimes used synonymously with vector search.




## Why vectors?

Vectors and vector search make it possible to search large volumes of almost any kind of unstructured data. This allows you to build applications using object–text docs, JSON, images, video, audio, and others—no matter how complex. The AI models do the hard work, generating the vectors for your data objects. Vectors allow you to put more information to work faster by eliminating the time-consuming process of keyword and metadata engineering.

# 2 AI use cases and data access platforms

Vector search powers a variety of semantic search and generative AI (GenAI) applications, such as self-service sales and customer support chatbots, fraud detection, recommendation engines, ad targeting, route planning, and more.

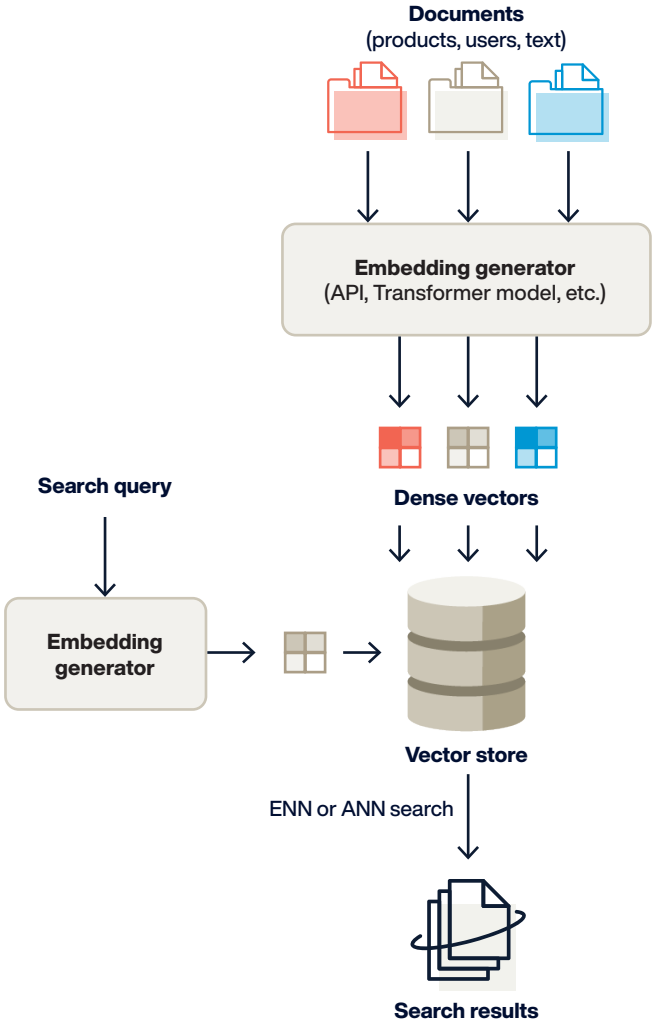
In order to understand which AI database management system (DBMS) is the best fit for your application, it's useful to understand how vector data is processed. There are three patterns that are often found in AI applications:

## Generating a vector embedding

Data is sent to a model, which returns a vector. The vector is stored with the object data it represents (or a reference to it if the data is stored externally). Different model types generate different vectors, e.g., LLMs vs. image classification, etc.

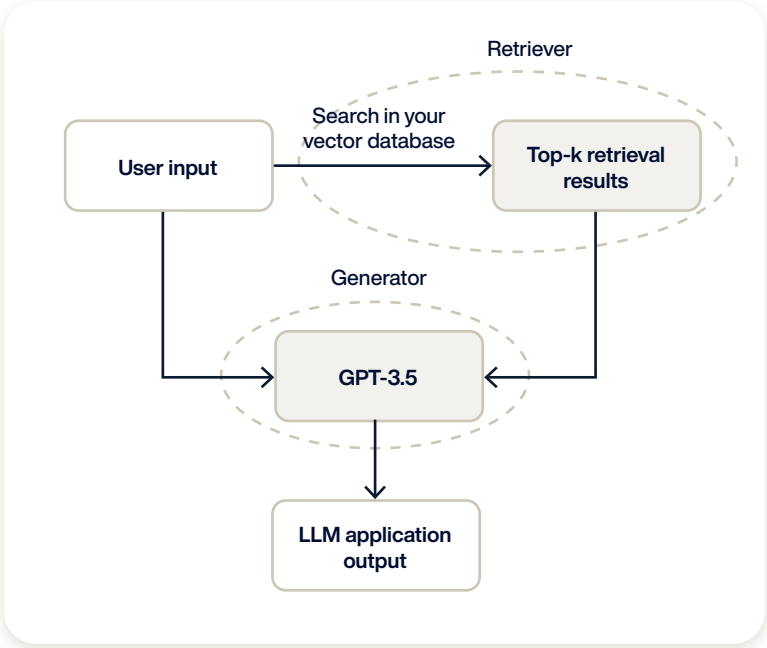
## Vector search

A search term is received and sent to a model to be vectorized. The returned vector is used as a query input to execute a similarity (nearest neighbor) search, finding data items similar in meaning or appearance to the vector in the query. As the sidebar explains, vector search can be combined with other searches (graph, SQL, etc.) to improve AI accuracy.



## GenAI and retrieval augmented generation (RAG)

GenAI and RAG use AI to formulate new content, such as answers to questions, marketing copy, and even application source code. It works similarly to the vector search pattern above, except that the query vector sent to the GenAI model includes an extra step: retrieving relevant content to pass to the model. This content enables the model to generate responses based on specific or proprietary information like product manuals, user data, etc. As the sidebar explains, a variety of search techniques can be used to retrieve the augmenting information.




Next, let's look at the role databases play in AI applications and how to find the best database for your AI project requirements.

## Combining vectors and graphs to improve AI accuracy

The example below highlights how combining different search techniques, such as vectors with knowledge graphs, can improve AI capabilities and accuracy.

**Bank fraud and anti-money laundering:** A vector search indicates that a transaction appears similar to other fraudulent activity, and a graph search immediately reveals whether the person, device, or accounts are related to other transactions.

**Recommendation engines:** As a shopper browses an e-commerce site, vector search reveals similar product catalog items (JSON) to display to the user; graph queries can further enhance the experience by displaying "frequently bought with" items.

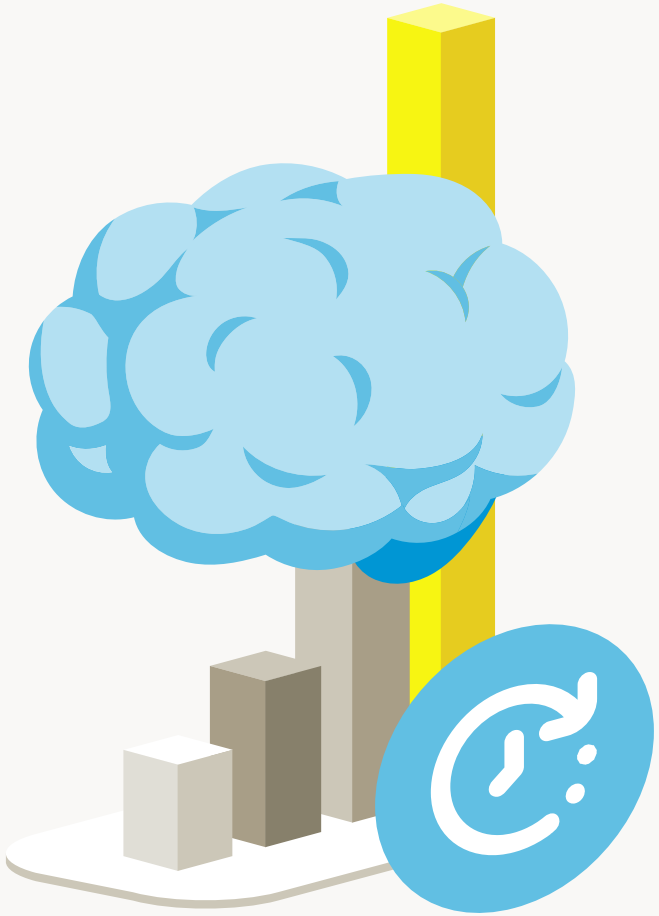
**More accurate RAG results:** A graph search can find items that are more highly related to a query and pass those to a GenAI model for more accurate and helpful responses.

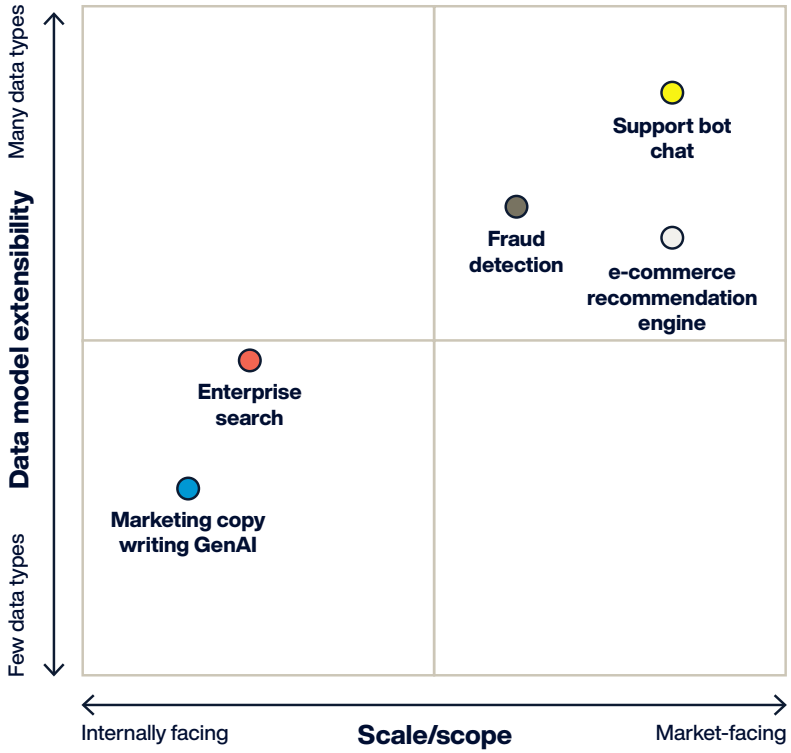
# 3 The role of the AI database

A DBMS is a critically important piece of the AI application stack. At the most basic level, the AI application database will facilitate the following:

- Storage of vector data
- Maintaining indexes of vector data to speed up searching
- Executing vector searches
- Securing and controlling access to data

Where AI databases differ is in their ability to support the scale, performance, extensibility, and budget aspects of various AI projects. For example, a database that works well for an internal AI proof of concept project (most databases do), might not work well at all for a market-facing AI application with thousands of users at a time, searching billions of vectors, and with developers facing management pressure to get to market ASAP.






The diagram above provides a framework for understanding AI application data management requirements.

- **Internally facing applications:** 1s-100s of concurrent users, as with a marketing copy generator used by the marketing team. Downtime and slow performance are tolerable (the business might be disrupted, but it won't affect customers or the brand).
- **Market-facing:** 100s - 1000s of concurrent users, as with an e-commerce or streaming media recommendation engine. Fast performance, 24x7 is required to provide customers with a positive experience.
- **Simple data model:** Smaller volumes of homogenous data (vectors only), accessed via a single type of search (e.g., vector similarity), as with an enterprise document search portal.
- **Complex data model:** Large volumes-terabytes of heterogeneous data and indexing, with a wider variety of query types, as with a fraud detection system combining vector search with graph and filtered search to identify illegal activity.

To help you match your project requirements with the best AI database for the job, consider using the following comparison points to evaluate technology and business fit:

- **Extensibility:** Does the DBMS support a wide range of data (e.g., vectors, key-value, graphs, SQL)?
- **Throughput:** Can the database handle many simultaneous user requests with fast performance?
- **Data volume:** Can it cost-effectively store, index, update, and search large volumes (terabytes) of data? Can it update data while also executing queries?
- **Production readiness:** Does the database have features to protect data and ensure non-stop performance?

Next, let's look at how some popular AI database alternatives compare.

# 4 Choosing an AI database

Let's look at how three popular categories of AI databases compare, using the comparison criteria we defined earlier:

- Vector-only database (e.g., Pinecone)
- Relational database (e.g., PostgreSQL + PgVector)
- Multi-model database (e.g., Aerospike)

## Vector database

| Extensibility | Throughput | Data volume | Production readiness |
|---|---|---|---|
| Low | Low (10s to 100s of qps) | Low (vectors only) | Low |

### Overview

Vector databases are used exclusively for vector data management and search. One of the best-known, Pinecone, is a fully managed cloud vector database that integrates with popular LLMs and other AI models to facilitate the generation of vector embedding data and perform similarity searches on vectors.

Pinecone only stores vector records (which can contain up to 40K of user-defined metadata for filtering). Because data volume is limited to vectors only, any other data management (such as the source data the vectors represent) or searching (e.g., relational, graphs, full-text, etc.) must be performed elsewhere, which complicates extensibility.

### Performance

Pinecone has different performance options: serverless or pod-based, the latter gives you more control over performance. Pods can execute 10 to 50 queries per second, depending on pod type and database size. Batch upserts should contain 100 or fewer vectors. Since it's a cloud service, network latency factors into performance, adding additional delays to the delivery of query results and other database request responses.

Vector databases like Pinecone are new and relatively immature when it comes to production-quality capabilities. Pinecone lacks database backup and restore, data integrity features, database monitoring and observability, while its database access control lags behind other databases.

### Bottomline

Vector databases like cloud-based Pinecone (or open-source alternatives like Weaviate and Qdrant) serve a single purpose: vector search. They are adequate for simpler projects like proof of concept or internal semantic search applications. Extending the use case beyond vector search turns the project into a multi-database solution, which complicates development, slows performance, and increases infrastructure costs. Immature or missing production-quality features like backup, security, and observability, increase administration costs and put data availability at risk.

# Relational database

| Extensibility | Throughput | Data volume | Production readiness |
|---|---|---|---|
| Medium | Medium (100s of qps) | Medium (100s of GBs) | High |

## Overview

Relational (SQL) databases are the most widely used DBMS, and many have added (or are adding) support for vector data indexing and search. The PgVector extension provides this capability for PostgreSQL, one of the most popular open-source relational databases.

The benefit of using an RDBMS with vector support is that you can store your vectors along with other relational data and search using SQL. This gives you more control over query results, with the advanced filtering, sorting, and processing capabilities of SQL.

Having all of your data in a single database simplifies development, leverages SQL database skills and tooling you already have in-house, and lowers infrastructure costs. You also benefit from very mature production deployment features such as ACID data integrity, thorough database security, performance monitoring, backup/restore, and more.

## Performance

Query and CRUD performance is faster than it is in vector databases and is proven at scale, up to a point. Relational databases do not scale out well to handle very large data sets (>1M rows) or large numbers of concurrent users. SQL Joins and over-indexing are known to inhibit relational database performance and throughput and make real-time vector search an impossibility in applications like ad targeting, experience personalization, recommendation engines, etc.

## Bottomline

Relational SQL databases are gaining support for vector indexing and search. If you're a heavy SQL RDBMS user and your preferred RDBMS has vector support, it's a safe choice for small—to medium-scale AI projects. If higher performance and scale (data volume and throughput) or query complexity is required, as with a market-facing AI system, you'll face data engineering challenges, poor performance, and high infrastructure costs.

# Multi-model database

| Extensibility | Throughput | Data volume | Production readiness |
|---|---|---|---|
| High | High (100,000s of qps) | High (100s of TBs) | High |

## Overview

A multi-model database can support a variety of data models. Aerospike, for example, supports a very broad spectrum of data models and can handle vectors, graphs, time series, JSON, geospatial, and others. Multi-model provides the greatest extensibility to handle any AI data processing requirement from a single database.

Multi-model provides great control over AI searches because it gives you many ways to retrieve data relevant to a query before passing it to AI models in a RAG architecture or to enrich the returned vector search results. By feeding more relevant data to AI models, you can generate more interesting insights for your users and more accurate AI responses (see sidebar on page 5).

Besides simplifying development via a single-database architecture, Aerospike also provides mature production deployment features for security, monitoring, SQL tooling compatibility, backup/restore, and more.

## Performance

A patented Hybrid Memory Architecture, parallel processing, and independently scalable compute and cache layers enable Aerospike to handle 100,000s of queries per second, even while ingesting new vector data at the same time. This makes Aerospike uniquely suited to handle large volumes of data (billions of vectors, hundreds of terabytes of data), many concurrent users, and also handle real-time AI use cases like fraud detection, ad targeting and more. In addition, Aerospike has been shown to operate much more cost-effectively than alternatives, typically requiring 80% less infrastructure to handle a given workload.

## Bottomline

A multi-model AI database like Aerospike offers the greatest extensibility to address any use case and provides high-throughput vector processing at scale–even supporting real-time use cases, with 80% lower infrastructure costs.

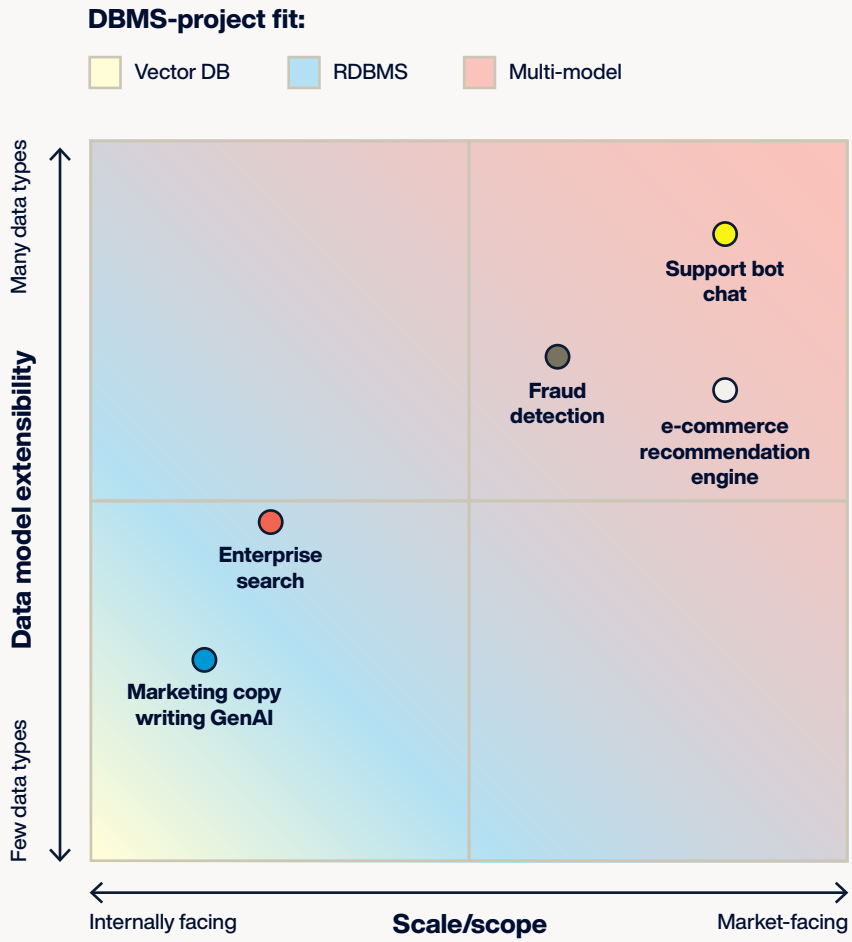
# In summary

The diagram shows the recommended AI project type-to-DBMS fit.

**Vector databases:** Niche databases suitable for proof-of-concept AI projects; their lack of extensibility and immature production capabilities create challenges for larger-scale usage. Extensibility and large data volumes require a multi-DBMS architecture.

**Relational databases + vector:** Eliminates multi-database architecture for many AI projects, but scaling to handle real-time AI, large user bases, or large data volumes can be challenging.

**Multi-model:** Broadest data model support and extensibility, plus its high-performance DBMS architecture make it the ideal fit for AI applications where real-time performance, large concurrent user bases, or AI accuracy requirements exist.

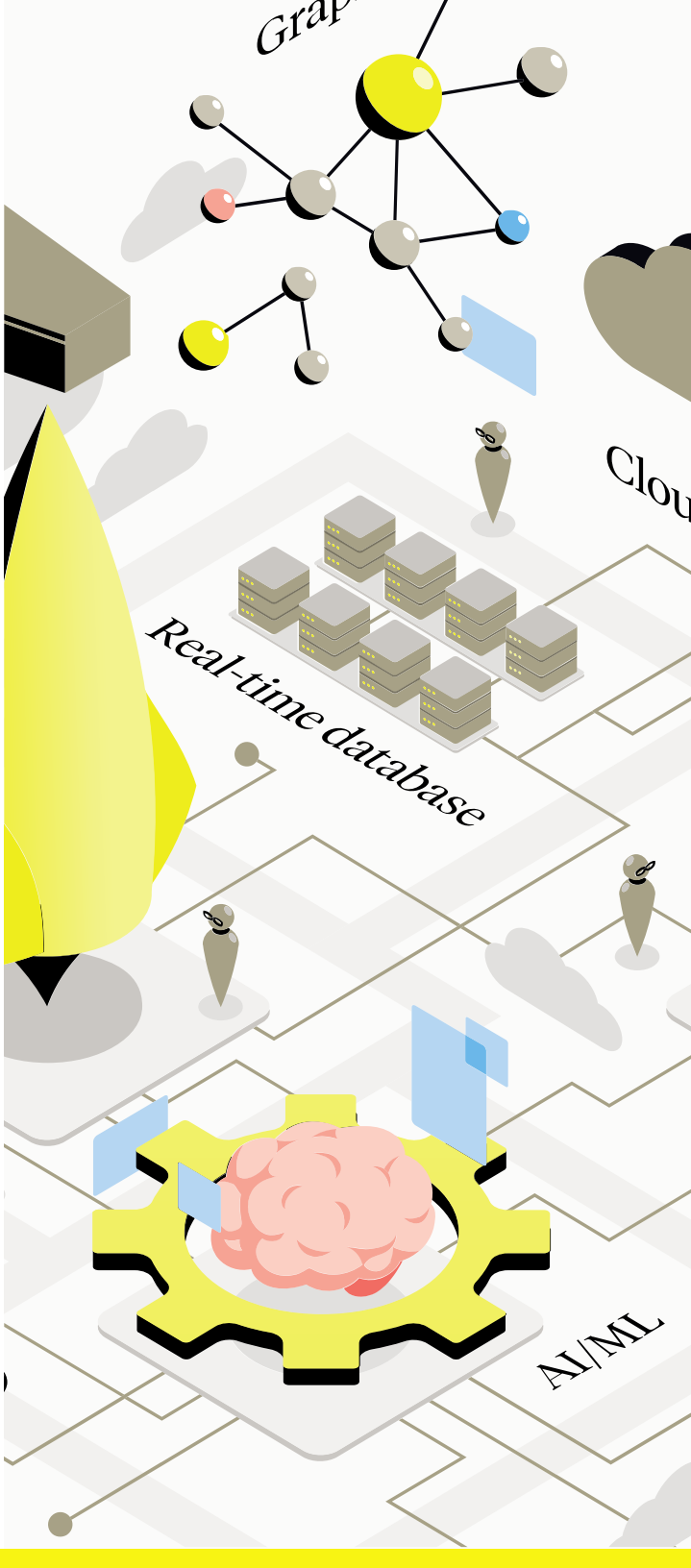


# The future of the AI database ecosystem

We hope this guide has helped you understand how to match AI database capabilities with AI project types. The one guaranteed constant is change. Today's vector databases will mature. Existing databases will add vector support. There will be market consolidation. AI models will evolve. Public policy will evolve, placing regulations on AI technology usage. To stay ahead in the rapidly evolving landscape of AI databases, be sure to revisit Aerospike regularly for future updates and access to informative content.

The important thing to do today is start building–with Aerospike or any other AI database–and gain first-hand experience working with vector data and AI models.

To get started with Aerospike, you can start using Aerospike Vector Search.



# About Aerospike

Aerospike is the real-time database built for infinite scale, speed, and savings. Our customers are ready for what's next with the lowest latency and the highest throughput data platform. Cloud and AI-forward, we empower leading organizations like Adobe, Airtel, Criteo, DBS Bank, Experian, PayPal, Snap, and Sony Interactive Entertainment. Headquartered in Mountain View, California, our offices include London, Bangalore, and Tel Aviv.

For more information, please visit https://www.aerospike.com.



**2440 W. El Camino Real, Suite 100, Mountain View, CA 94040 | (408) 462-2376**