Innovation Insight: Vector Databases

Published 4 September 2023 - ID G00788401 - 13 min read

By Analyst(s): Arun Chandrasekaran, Radu Miclaus

Initiatives: Digital Products and Services; Artificial Intelligence; Generative AI Resource Center

The explosion in unstructured data and generative artificial intelligence models that harness its power has increased enterprises' interest in vector databases. CTOs should evaluate the benefits, risks and opportunities presented by these databases to tap their business value for their AI use cases.

Overview

Key Findings

- The growth of generative artificial intelligence applications is creating increased interest in vector databases, which help enterprises store and retrieve their data as a vector embedding, enabling semantic search and quick retrieval of that data.
- Popular use cases for vector databases include product recommendations, similarity search, fraud detection and generative-AI-powered, question-and-answer applications.
- Vector databases can provide long-term memory for stateless generative AI models and can boost their accuracy and reduce their hallucinations through prompt augmentation.
- There is a looming battle for customer wallet share between closed-source and open-source vector databases, as well as between purpose-built vector databases versus incumbent databases and search vendors adding vector storage and retrieval as an add-on capability.

Recommendations

Enterprise architecture/technology innovation (EA/TI) leaders responsible for their enterprises' digital futures should:

- Assess the extent to which they are developing their own generative AI applications to judge the relevance of vector databases to their organizations.
- Identify projects where adding a vector database could improve scalability, performance, accuracy and lower risks of their AI use cases, if they have decided to move ahead with a build approach.
- Invest in training and education about vector database capabilities to avoid expensive misuse and misalignment.
- Start with cloud-based managed services to reduce operational complexity and optimize the total cost of ownership in the short term.

Strategic Planning Assumptions

By 2026, more than 30% of enterprises will have adopted vector databases, which is a significant increase from fewer than 2% today.

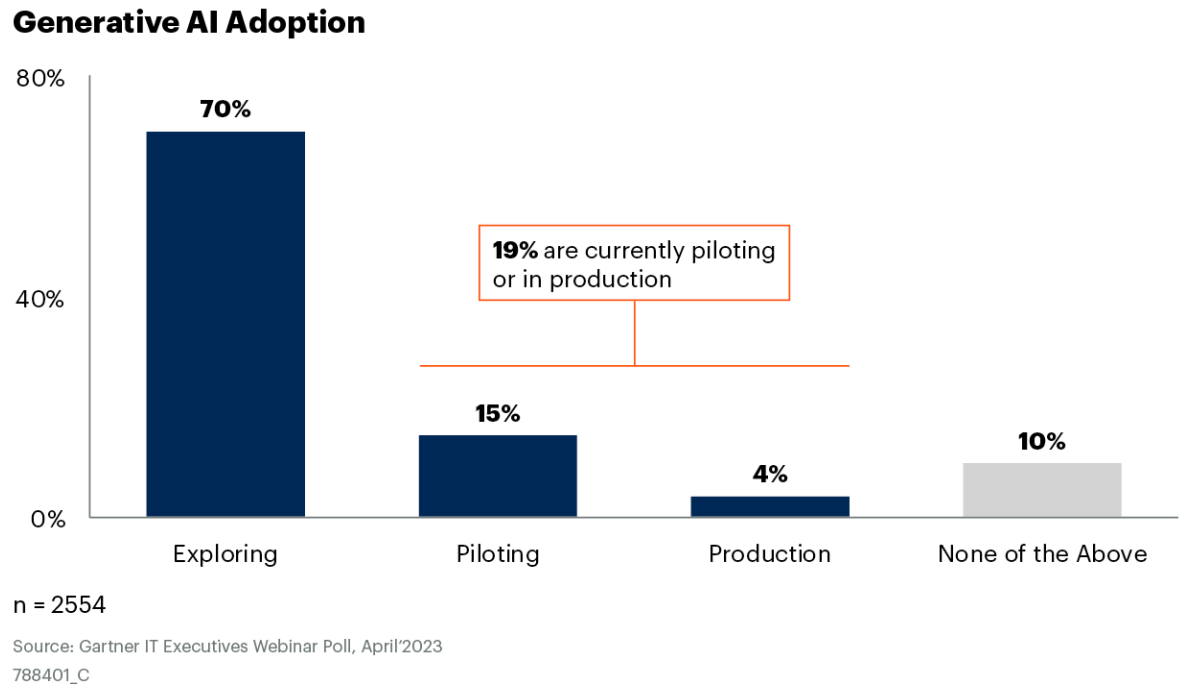
By 2026, more than 70% of generative artificial intelligence (AI) natural language processing (NLP) use cases for questions and answers, will leverage vector databases to "ground" the AI foundation models.

By 2026, more than 60% of vector database deployments will be cloud-based managed services, rather than self-managed products.

Introduction

There has been an explosion of innovation in the field of AI, specifically generative AI, during the past five years. IT leaders have shown a keen interest in harnessing AI's potential to build sustainable, competitive differentiation. According to a Gartner Webinar poll, nearly one-fifth of organizations are already piloting or have implemented generative AI applications, while another 70% are exploring and looking to deploy imminently (see Figure 1).

Figure 1: Generative AI Adoption



Although this growth is impressive, aligning massive, pretrained generative AI models with appropriate use cases and integrating them with enterprise data is fundamental for success with generative AI initiatives. Generative AI models are a significant technical advancement; however, they still suffer from several limitations, such as lack of domain specificity and the frequent occurrence of "hallucinations." Vector databases enable enterprises to retrieve data based on its meaning or context and to use that data to augment the generative AI prompts. This improves the quality and relevance of the resulting generated outputs. As a result, vector embedding management is emerging as an important component of generative AI architectures.

Although vector databases predate the adoption of generative AI in the enterprise, their ability to enable rapid scalability and performance of generative AI applications has resulted in an explosion of their popularity and adoption during the past 12 months. They have several capabilities to aid generative AI use cases at scale; however, whether they will be a separate market category or will converge with broader database markets is yet to be determined.

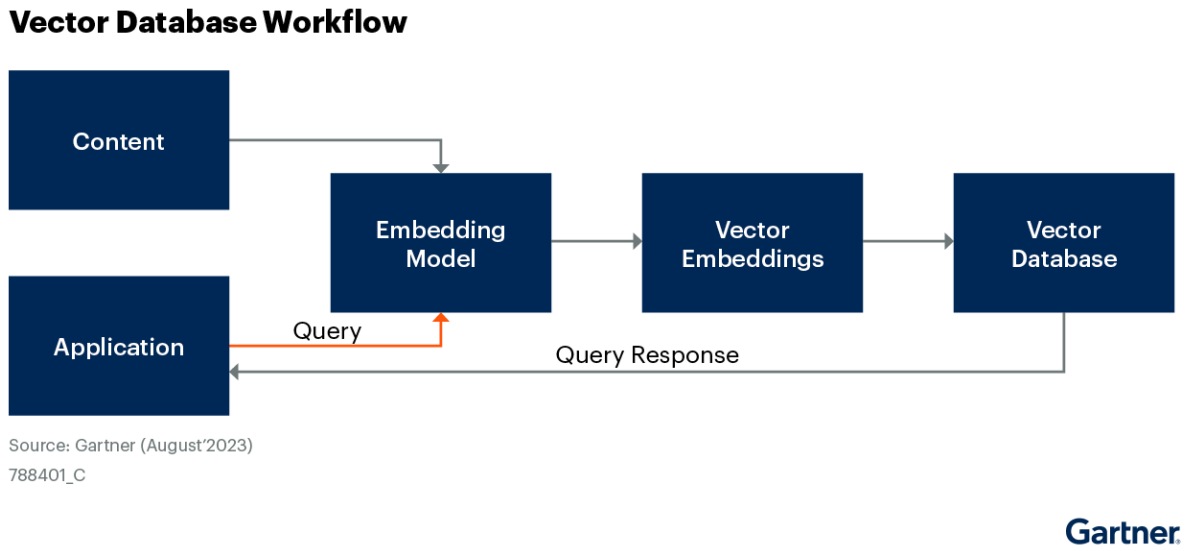
Description

Vector databases store data in multidimensional space. In such databases, each data point is represented by a vector with a fixed number of dimensions, which can be compared via mathematical operations, such as distance measures, with other data. They are typically powered by indexes or graphs, such as k-nearest neighbor (k-NN), Hierarchical Navigable Small World (HNSW) and Inverted File Index (IVF).

Here's how the vector databases (see Figure 2) typically enable generative AI use cases:

- **Preprocessing and Indexing** — The organizational data is fed into an embedding model. This is a specialized model (could be a large language model [LLM]) that converts organizational data into a vector embedding. The vectors are indexed into the structures, using the algorithms described above. The output of an embedding model is stored in a vector database. The embeddings are placed into an index, so that the database can quickly perform searches.
- **Query Response** — When a user query (or a prompt, in the case of generative AI applications) comes in, for each query, a vector embedding is computed using the same model that was used for the data. The database then finds the closest vectors to the given vector computed for the query. This process is used to augment the system prompt and provides information to be included in the response of the generative AI model.
- **Postprocessing** — In some cases, the vector database retrieves the final nearest neighbors from the dataset and postprocesses them to return the final results. This step can include reranking the nearest neighbors, using a different similarity measure.

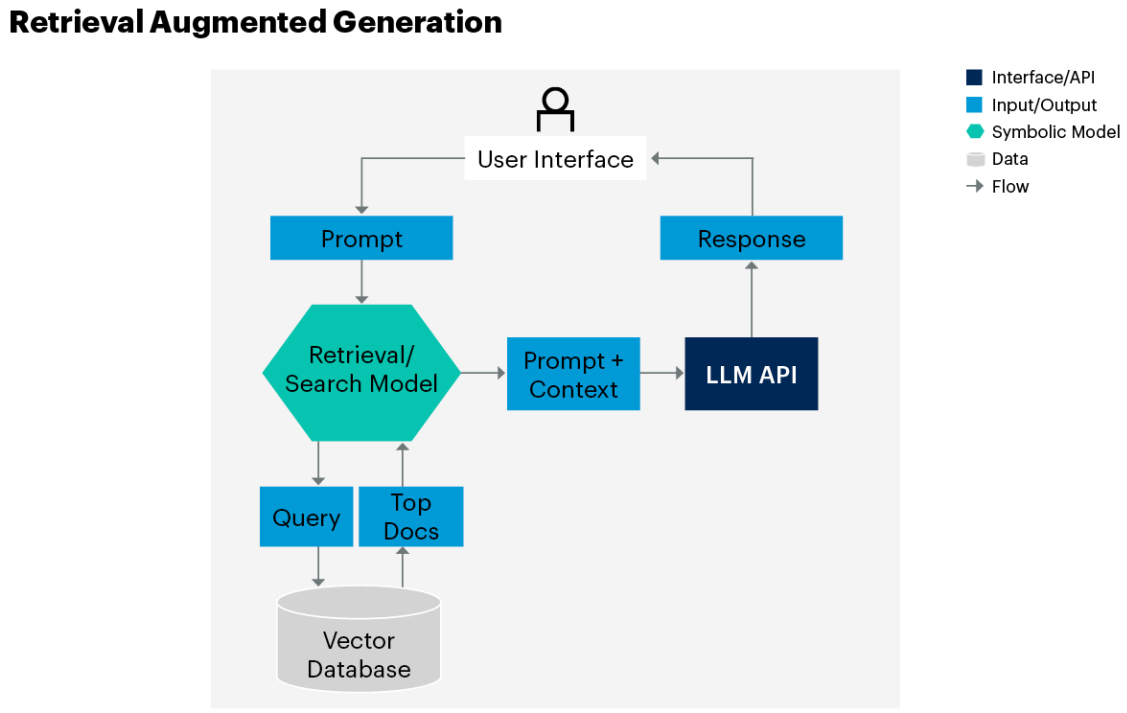
Figure 2: Vector Database Workflow



Vector databases play an important role in "grounding" generative AI models by boosting accuracy through reducing the hallucinations in model response. Retrieval augmented generation (RAG) is a key technique for grounding these models. A RAG approach often consists of preprocessing and runtime components. In preprocessing, an organization's information is divided into chunks, and that data can be stored in a vector database as embeddings. A user prompt or instruction would trigger a similarity search on the vector database.

The results are then ranked, taking into account the limits of the context window of the model, and the constructed prompt is sent to the generative AI model to produce a response. The prompt augmentation is an important step in ensuring accurate responses, as well as lowering hallucinations. The data retrieval process here (see Figure 3) is inherently more simple than fine-tuning a model with lower cost and complexity.

Figure 3: Retrieval Augmented Generation



Source: Gartner
788401_C

Gartner

The combination of generative AI models (such as GPT-3 and GPT-4) with vector databases and model API orchestration tools such as LangChain is becoming more common in generative AI deployments.

For enterprises choosing a vector database to enhance their generative AI applications, the following factors are important considerations:

- **Performance** — Generative AI use cases can rely on batch-oriented or real-time data. Depending on the use case, the throughput and/or latency of the database will be a key consideration. Also, if you are feeding data in real-time, the ability to rapidly chunk and index it will be important.
- **Scalability** — The efficiency of your database in chunking and storing data, as well as scaling horizontally to accommodate a high degree of vector embeddings.

- **Reliability** — Built-in data management features for high availability, backup and recovery are key considerations. If offered as a service, the service-level agreements (SLAs) offered by your provider on availability and recoverability should be considered.
- **Ease of Use** — Robust user experience (UX), quality of API documentation and global availability of technical support are key considerations when choosing a vector database vendor. The developer experience can't be ignored.
- **Security** — Role-based access control (RBAC) and how providers ensure end-to-end encryption of data are important. For cloud-based services, cloud security controls — such as data safeguards, the ability to meet compliance mandates, quality of third-party audits, risk management techniques and incident notification — are critical.
- **Integration** — Vector databases need to integrate closely with your generative AI models. These include LLMs; models that generate embeddings; API orchestration tools (e.g., LangChain); and machine learning operationalization (MLOPs) tools you may have for prompt engineering, model monitoring or content moderation.
- **Open Source Versus Closed Source** — The choice of open source versus closed source is made based on the need for deployment flexibility, potential customization, in-house skills and time-to-market needs.
- **Deployment Model** — Determine whether you need a self-hosted or fully managed vector database. Given the newness and the operational complexity involved, most clients are opting for managed vector databases.
- **Total Cost of Ownership (TCO)** — Most vendors offer a freemium pricing, with usage-based pricing beyond the free tier. Consider the long-term TCO, which is a function of price, usability and skills availability.

Benefits and Uses

Vector databases, which can greatly enhance performance and scalability across various applications, including nongenerative AI ones offer several benefits over traditional databases:

- **Faster Processing** — Vector databases are designed to store and retrieve unstructured data, such as documents and images, efficiently, enabling faster retrieval of large datasets (e.g., search platforms).

- **Scalability** — Vector databases can scale horizontally, making it possible to store and retrieve huge volumes of unstructured data.
- **Tighter Affinity With Generative AI Use Cases** — The use of vector databases is growing across a variety of generative AI applications. This is due to the predominance of unstructured and semistructured data, with vector databases becoming a prominent part of the technical architecture for question and answer (Q&A) applications.

The key use cases for vector databases are:

- **Q&A Applications** — Vector databases enable enterprises to leverage the power of generative AI models, but mitigate their risks by "grounding" them. Vector embeddings serve as an important mechanism to augment the user prompts with organizational data, thereby improving the quality of model output and reducing model hallucinations.
- **Recommendation Engines** — Vector databases can be used to represent user preferences and recommended items, allowing the databases to find the best matches for user queries and provide personalized suggestions.
- **Content Search** — Vector databases enable users to evolve from keyword search to semantic search. Since the data is represented as a multidimensional vector, similarity searches are quicker and easier. This is especially useful in e-commerce or customer chatbots, where users can search for items using descriptions or images or pull up information, based on the intent and context of the question.
- **Fraud Detection** — Vector databases are also helpful in fraud detection. They may be applied to find data patterns that point to fraud. For example, a specific set of anomalous transactions with similar vector representations might indicate fraud.

Risks

Just like any technology experiencing a substantial increase in interest, there are some risks to the technology, process and talent decisions that IT leaders need to consider as they explore vector databases.

- **Risks of Early-Stage Market** — Vector databases are a relatively new category of products. Whether this will exist as a distinct segment in the long run and who the winners will be are still unknown. Hence, CTOs need to accept the risks of early-stage technology markets and mitigate vendor viability risks by evaluating exit cost risks and limiting long-term vendor contracts.

- **Lack of Investment in Appropriate Education and Exploration Cycles** — Although rapid innovation in generative AI is creating fear of missing out (FOMO) in organizations. Being unprepared for the learning curve that business and technology teams need can be detrimental to the ability to make the right decisions. Mitigate this risk by organizing informational sessions with analysts and subject matter experts, as well as with ideation and experimentation work that can be done iteratively.

- **Not Leveraging Technology Vendors Appropriately** — Sometimes being a follower will mitigate the risk of discovery and trial-and-error that early adopters go through. Use the best practices that vendors have learned from past implementations, and make use of them via actual features in offerings or services. There are multiple cases in which vector databases are used as a back-end managed service, without the need for buyer organizations to manage them directly.

- **Overengineering the Minimum Viable Product (MVP)** — If the decision to build is made, fast ideation and iteration for MVPs are important to show the potential and the use case. By trying to overengineer the initial MVP, you can risk missing the mark on timing, illustration of use case and the benefits necessary to secure buy-in and investment into architecting in a more-robust fashion. There are options for approachable vector management for vector libraries, as well as open-source variations of vector databases.

- **Not Understanding the Long-Term Costs of Managing and Scaling Applications** — This risk can materialize in two ways. It can be the result of overestimating the ability to own and operate the technology, or not fully understanding the vendor's commercial offerings and cost structure as the application scales beyond MVP.

Adoption Rate

Vector databases are still nascent, and we believe that fewer than 2% of businesses have deployed them in production. However, this is a fast-moving field, and the adoption rate is likely to accelerate in the future. Early adoption is happening in technology companies. Technology vendors in the B2C and B2B spaces are investing heavily, not only in scaling vector search for their offerings, but also ramping up the use cases in which data retrieval is augmenting the generative process (i.e., RAG implementations).

Enterprises are likely to adopt this technology as their AI maturity curve moves further to support fully customized applications where the support and management of the applications' back end will be owned by the technology teams.

Alternatives

An alternative option for vector management is vector libraries. Both options have been used to improve semantic search in support of generative AI applications. Vector libraries tend to be more appropriate for exploration and experimentation when the domain of the search is fixed or slow-moving, or the application lacks the need for operational scale and process. They can be a great choice for ideating and experimenting with minimum viable products (MVPs). They can be effective for working with knowledge bases that are relatively static or change slowly, and applications that do not require high availability and fast updates for querying (ability to query during import for large knowledge base updates). Some examples of available vector libraries and vector stores include Faiss from Facebook, Google ScaNN, Spotify Annoy, NMSLIB and HNSWLIB.

Text search vendors such as Elastic have added support for vector embedding. Both SQL database projects/vendors (e.g., PostgreSQL and Singlestore) and NoSQL database vendors (e.g., MongoDB and Redis) have also added vector capabilities.

Recommendations

EA/TI leaders responsible for their enterprises' digital futures should:

- Assess the extent to which they are developing their own generative AI applications to judge the relevance of vector databases to their organizations.
- Identify projects where adding a vector database could improve scalability, performance, accuracy and lower risks of their AI use cases, if they have decided to move ahead with a build approach.

- Invest in training and education about vector database capabilities to avoid expensive misuse and misalignment.
- Start with cloud-based managed services to reduce operational complexity and optimize the total cost of ownership in the short term.

Representative Providers

Growing interest in vector databases from investors has given rise to a vibrant ecosystem of start-ups, many of which are open-source. Incumbent database vendors have also been adding support for vector embeddings, thereby increasing choices for IT buyers. Pure-play start-ups, such as Pinecone, Weaviate, Zilliz, Qdrant and Chroma, have emerged during the past three years. AWS, Microsoft and GCP have also added managed vector search options in their platform as a service (PaaS) portfolios. The key vendors in the vector database are summarized in Table 1.

Table 1: Key Vendors in Vector Database

(Enlarged table in Appendix)

Vendor Name	Headquarters	Deployment Model	Open or Closed
Activeloop	San Francisco, CA	Self-managed and cloud managed	Open-source
Amazon Web Services	Seattle, WA	Cloud-managed	Closed-source
Chroma	San Francisco, CA	Self-managed	Open-source
Elastic	San Francisco, CA	Self-managed and cloud-managed	Open-source
Google Cloud	Mountainview, CA	Cloud-managed	Closed-source
Microsoft Azure	Redmond, WA	Cloud managed	Closed-source
MongoDB	New York City, NY	Self-managed and cloud-managed	Open-source
Pinecone	NYC, NY	Cloud-managed	Closed-source
Qdrant	Berlin, Germany	Self-managed and cloud-managed	Open-source
Redis	Mountainview, CA	Self-managed and cloud-managed	Open-source
SingleStore	San Francisco, CA	Self-managed and cloud-managed	Closed-source
Weaviate	Amsterdam, Netherlands	Self-managed and cloud-managed	Open-source
Zilliz	Redwood Shores, CA	Self-managed and cloud-managed	Open-source

Source: Gartner (August 2023)

In addition to the vendors cited above, there are community-supported, open-source projects, such as Faiss, PostgreSQL, Vald and Vespa.

Evidence

Detailed interviews and written surveys were conducted with most vendors profiled in this research. In addition, the authors have received more than 25 inquiries on this nascent topic.

Recommended by the Authors

Some documents may not be available as part of your current Gartner subscription.

Quick Answer: How Will Prompt Engineering Impact the Work of Data Scientists?

How to Choose an Approach for Deploying Generative AI

Glossary of Terms for Generative AI and Large Language Models

Hype Cycle for Artificial Intelligence, 2023

Table 1: Key Vendors in Vector Database

Vendor Name	*Headquarters*	*Deployment Model*	*Open or Closed*
Activeloop	San Francisco, CA	Self-managed and cloud managed	Open-source
Amazon Web Services	Seattle, WA	Cloud-managed	Closed-source
Chroma	San Francisco, CA	Self-managed	Open-source
Elastic	San Francisco, CA	Self-managed and cloud-managed	Open-source
Google Cloud	Mountainview, CA	Cloud-managed	Closed-source
Microsoft Azure	Redmond, WA	Cloud managed	Closed-source
MongoDB	New York City, NY	Self-managed and cloud-managed	Open-source
Pinecone	NYC, NY	Cloud-managed	Closed-source
Qdrant	Berlin, Germany	Self-managed and cloud-managed	Open-source
Redis	Mountainview, CA	Self-managed and cloud-managed	Open-source
SingleStore	San Francisco, CA	Self-managed and cloud-managed	Closed-source
Weaviate	Amsterdam, Netherlands	Self-managed and cloud-managed	Open-source
Zilliz	Redwood Shores, CA	Self-managed and cloud-managed	Open-source

Source: Gartner (August 2023)