MongoDB

Martin Johannes Brucker

Agenda

- Documents & document databases
- Company
- Licensing
- Product portfolio
- IT-architecture
- Data models
- Client libaries
- Use-cases
- Installation, configuration and programming

Documents

A document is a fundamental unit of data in a document-oriented database. It represents a self-contained piece of information that can be stored, retrieved, and manipulated.

Document databases

A document database, part of the NoSQL family, is designed to store, retrieve, and manage document-oriented information, usually in JSON format. It’s known for its flexible schema, ease of development, and ability to scale horizontally.

Source https://aws.amazon.com/de/nosql/document/

Company

- **Headquarters**: MongoDB's headquarters are in **New York City**, a major center for finance, culture, and technology1.
- **Customer Base**: MongoDB boasts a strong customer base with over **47,800** customers, highlighting its broad market acceptance2.
- **Usage**: The platform has been downloaded over **265 million times**, demonstrating its high demand and widespread use3.
- Revenue Stream: MongoDB has shown significant growth with a 31.07% **increase** in annual revenue for 2024, reaching **$1.683 billion**. A substantial portion of this revenue comes from their cloud services, MongoDB Atlas, which represents **65%** of their total Q4 revenue45.

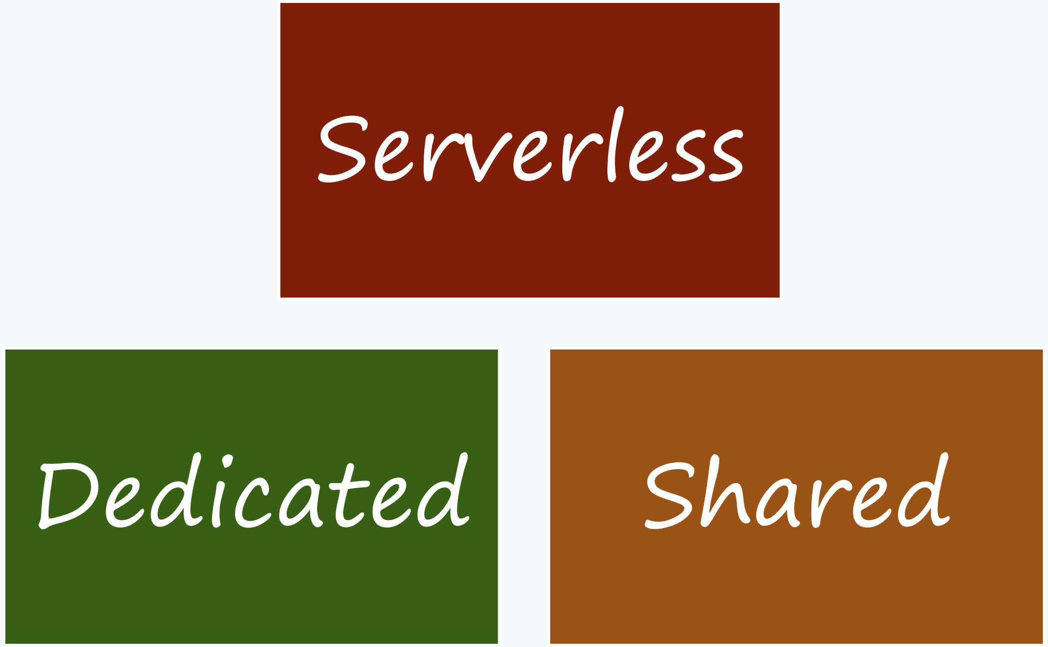
Source https://investors.mongodb.com/news-releases/news-release-details/mongodb-inc-announces-fourth-quarter-and-full-year-fiscal-2023

Licensing

- **Serverless**: Starting **at $0.10/million** reads, ideal for serverless applications with variable traffic. Offers up to **1TB storage**, scales resources seamlessly, and includes always-on security and backups.

- **Dedicated**: Starting at **$57/month** (estimated at **$0.08/hour**), suitable for production applications with complex workloads. Provides **10GB to 4TB storage, 2GB to 768GB RAM**, network isolation, and multi-region options.

- **Shared**: Available for **$0/month**, perfect for those new to MongoDB. It includes **512MB to 5GB storage**, shared RAM, and the option to upgrade to dedicated clusters, with no credit card needed to start.


Serverless

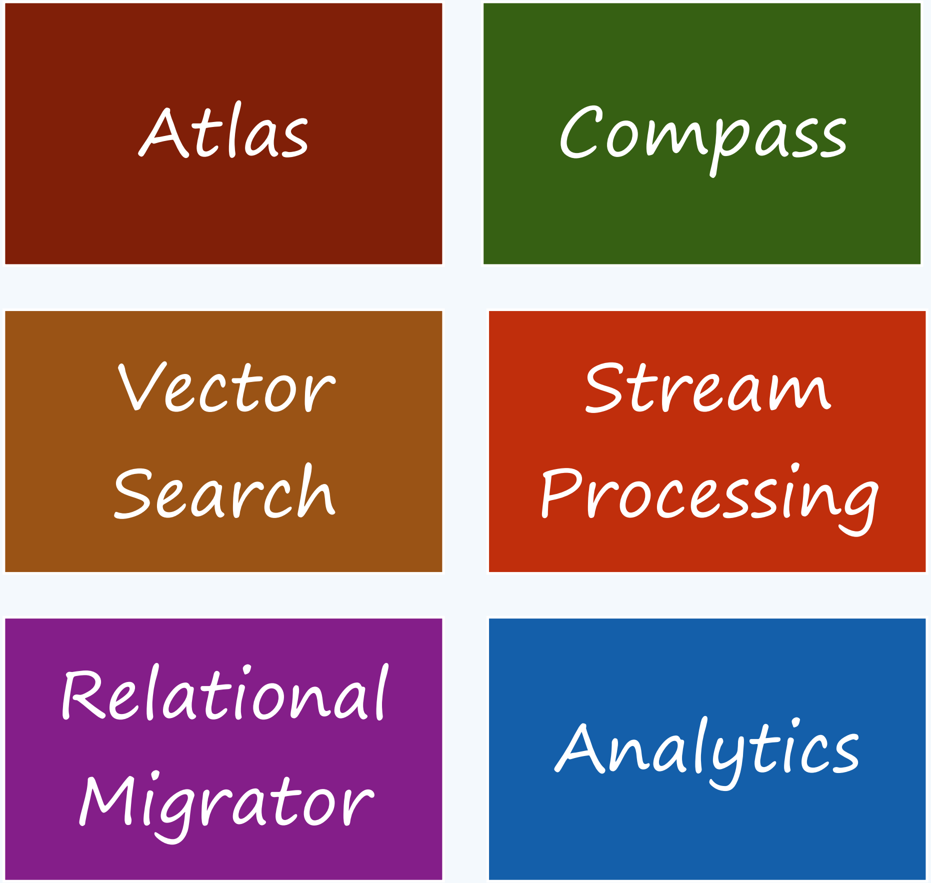
Dedicated

Shared

Source https://www.mongodb.com/pricing

Product portfolio

- **Atlas**: MongoDB's fully-managed cloud database service that automates complex infrastructure tasks.
- **Compass**: The intuitive GUI for MongoDB, enabling easy data exploration and manipulation.
- **Vector Search**: A powerful search engine that allows for rich text and geospatial queries within MongoDB.
- **Stream Processing**: Real-time data processing capabilities within MongoDB for immediate insights and actions.
- **Relational Migrator**: Simplifies the transition from relational databases to MongoDB, preserving data integrity.
- **Analytics**: Advanced analytics on operational data within MongoDB, eliminating the need for separate data warehouses.



Atlas

Compass

Vector Search

Stream Processing

Relational Migrator

Analytics

Source https://www.mongodb.com/products

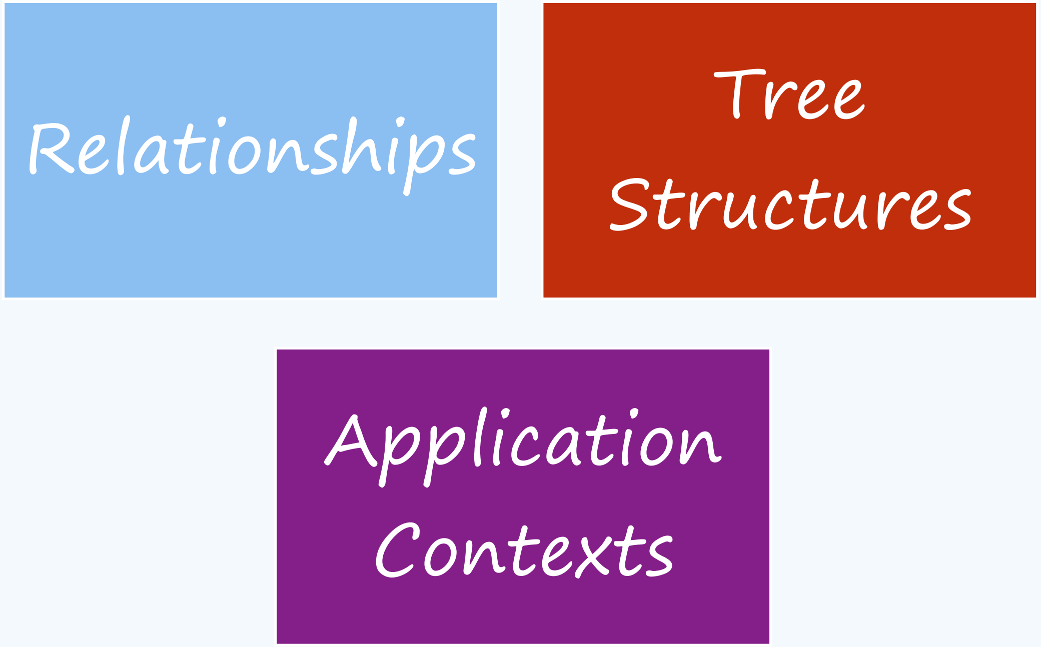
IT-architecture

- **Document Data Model**: MongoDB uses a document data model that allows for a natural and intuitive way to work with data. It’s flexible, enabling quick adaptation and changes, and supports a wide variety of data and queries.
- **Distributed Systems Design**: The architecture allows for intelligent data placement. You can scale horizontally by adding more machines to share the data set and load, which is essential for handling big data and intense workloads.
- **Unified Experience**: MongoDB provides a consistent experience across various platforms, from the cloud to the edge, supported by comprehensive and integrated developer services. This approach helps future-proof work and eliminates vendor lock-in.

Source
https://www.mongodb.com/resources/products/fundamentals/mongodb-architecture

Data Models

- Relationships:
  - Define how data entities associate with one another.
  - Essential for structuring databases and queries.
  - Include various types like one-to-one, one-to-many, and many-to-many.
- Tree Structures:
  - Represent hierarchical data models.
  - Efficient for organizing and retrieving nested data.
  - Common in applications like file systems and organizational charts.
- Application Contexts:
  - Dictate how data models apply within specific domains.
  - Ensure consistency and clarity within bounded contexts.
  - Facilitate integration across different systems and models.



Relationships

Tree Structures

Application Contexts

Source https://www.mongodb.com/docs/manual/applications/data-models/

Client libaries

- **PyMongo**: This is the official MongoDB driver for synchronous Python applications. It's well-documented and widely used in the community.
- **Motor**: If you need to access MongoDB in a non-blocking manner or from co-routines, Motor is the recommended asynchronous driver.
- **MongoEngine**: An Object-Document Mapper (ODM) for MongoDB and Python. It's a higher-level abstraction over PyMongo.

Source https://www.mongodb.com/docs/drivers/pymongo/

Use-cases

- **Big Data**: MongoDB is well-suited for Big Data applications due to its scalability and performance with large datasets.
- **Content Management Systems (CMS):** Its flexible schema allows for easy storage and retrieval of various types of content.
- **Mobile Applications**: MongoDB's features support the rapid development and scaling of mobile apps.
- **Real-Time Analytics**: The database can handle real-time data processing, making it ideal for analytics applications.
- **Internet of Things (IoT):** MongoDB can store and analyze the vast amount of data generated by IoT devices.

Source https://hevodata.com/learn/mongodb-use-case/

Installation, configuration and programming

Installation and configuration using docker

```
docker run
–d --name mongo-on-docker
-p 27017:27017
-e MONGO_INITDB_ROOT_USERNAME=mongoadmin
-e MONGO_INITDB_ROOT_PASSWORD=secret mongo
```

Programming with pymongo

```
!pip install pymongo

from pymongo import MongoClient

# Connect to the MongoDB server
client = MongoClient("localhost", 27017)
```