



MongoDB®

A document-oriented database program

PRESENTED BY HANNES BÄHR

Content

1. Document Data and Databases
2. The Company MongoDB: Basic Facts
3. MongoDB Products and Licenses
4. IT Architecture Overview
5. Data Models in MongoDB
6. APIs and Client Libraries
7. Use Cases in Industry
8. Set Up and Application



1. Document Data and Databases

Document Data

Document data is a form of self-contained semi-structured data organized as key-value pairs, typically stored in formats like JSON, BSON, or XML.

Characteristics

- Self-contained: All related data stored in one document
- Supports nesting: Objects within objects, arrays, etc.
- Schema-flexible: Different documents in the same collection can have different fields
- Human-readable and easy to parse

Importance in Big Data

- Ideal for complex, hierarchical, dynamic data, e.g. nested structures in IoT or user profiles
- Enables fast development and agile iteration due to schema flexibility
- Reduces need for relational joins by embedding related data
- Supports horizontal scaling: suitable for large-scale, high-volume applications

1. Document Data and Databases

Document Databases

Document databases are NoSQL databases that can store and retrieve data as documents in different formats (e.g. JSON, BSON) grouped into collections.

Key Features

- Stores data as documents, not rows
- Each document is uniquely identified by an ID
- No enforced schema: Documents can evolve over time
- Documents are indexed for efficient querying

Advantages over Traditional Databases

- Better suited for agile development and frequent schema changes
- Fewer need for joins: Related data often embedded in one document
- Easier to scale horizontally and better performance for hierarchical or nested data
- Natural fit for modern applications: Web, mobile, microservices

2. The Company MongoDB: Basic Facts

Key Dates

- Founded in 2007 as 10gen, Inc. by Dwight Merriman, Eliot Horowitz, and Kevin Ryan
- Renamed to MongoDB, Inc. in 2013 to align with the product's name
- IPO on the NASDAQ in 2017 (ticker: MDB)
- Since foundation headquartered in New York City, USA

Mission and Focus

- Providing a developer-friendly database platform for modern applications
- Emphasizes schema flexibility, horizontal scalability, and cloud-native architectures
- Bridging traditional database reliability with NoSQL agility

Growth, Main Products and Community

- Flagship products: MongoDB Server, MongoDB Atlas, Atlas Search, Charts, Mobile Sync
- Trusted by over 40,000 customers, including Adobe, Bosch, and AT&T
- Strong open-source community and active ecosystem
- Hosts globslevents (MongoDB World, MongoDB.local) and offers free learning resources

3. MongoDB Products and Licenses

Product	Description	Key Features	License
MongoDB Community Server	Free, open-source database for local and self-managed use	<ul style="list-style-type: none">• Core NoSQL engine• Flexible, dynamic schema• JSON/BSON document storage	Server Side Public License (SSPL) ¹
MongoDB Atlas	Fully managed cloud DBaaS on AWS, Azure, GCP	<ul style="list-style-type: none">• Auto-scaling and monitoring• Automated backups• Built-in security and compliance	Commercial License
MongoDB Enterprise (Advanced)	Self-managed, on-prem or private cloud edition for businesses	<ul style="list-style-type: none">• LDAP/SAML integration• Encryption at rest and in transit• BI/Analytics connectors	Commercial License
Atlas Tools (e.g. Atlas Search, Charts, Realm)	Add-on tools for search, visualization, and mobile apps	<ul style="list-style-type: none">• Full-text search (Atlas search)• Data visualizations (Charts)• Mobile data sync (Realm)	Commercial License (part of Atlas plans)

¹Note: SSPL restricts third-party DBaaS offerings without open-sourcing the management code. It is not approved by the Open-Source Initiative (OSI).

4. IT Architecture Overview

Application and Driver Layer

- Applications connect to database through MongoDB driver
- Driver manages communication and load balancing

Query router (mongos)

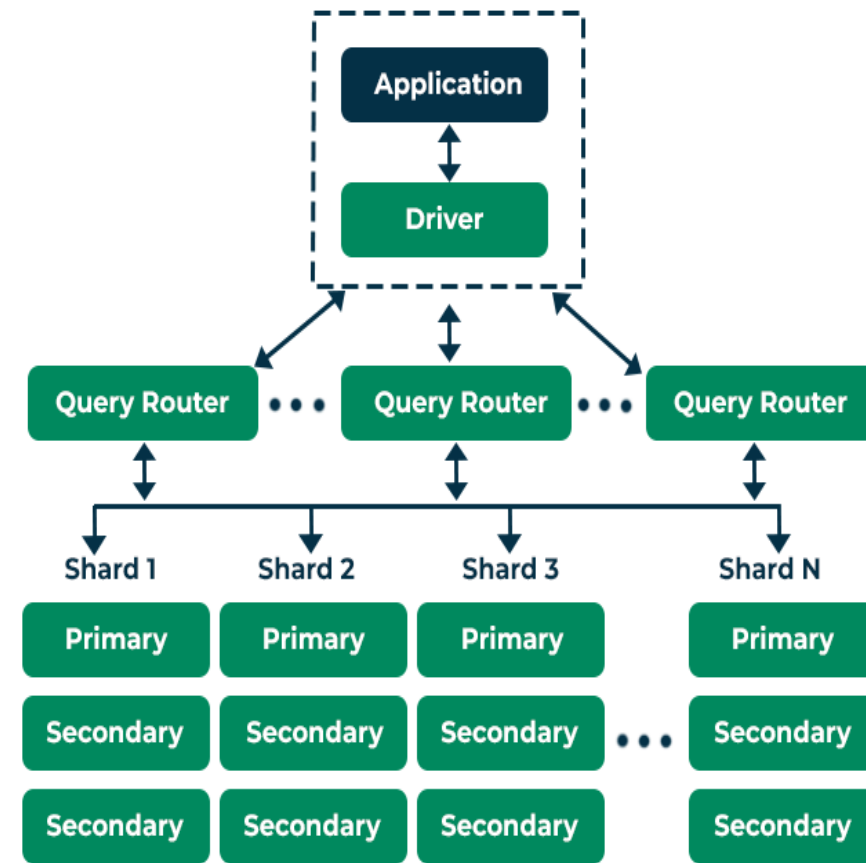
- Distribute client requests to the correct shards
- Handle query targeting and aggregation
- Merge results from multiple shards
- Multiple routers for scalability and redundancy

Shards

- Data is split across shards: each stores a subset of the data based on the shard key
- Each shard contains a replica set: group of one primary and multiple secondaries node(s)

Replica Sets

- Ensure high availability: if a primary fails, a secondary is promoted automatically
- Data is replicated from the primary to secondaries



4. IT Architecture Overview

Horizontal Scaling

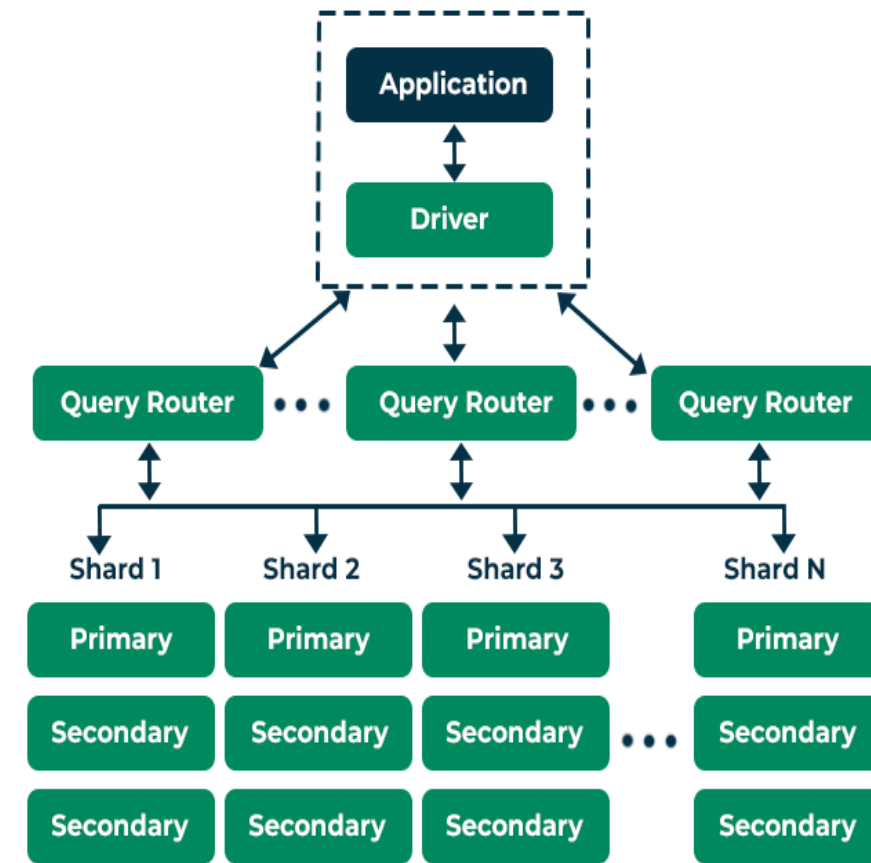
- Architecture enables scaling out by adding more shards
- Supports read and write operations across cluster

Document Storage Model

- Data is stored in collections
- Each containing multiple BSON documents
- Each document is self-contained and schema-flexible
- Logical separation, as collections live inside databases

Storage Engine

- Default: WiredTiger (concurrency, compression, journaling)
- Older engine: MMAPv1 (deprecated)
- In Atlas, storage is fully managed and abstracted.



5. Data Models in MongoDB

Field

- A key-value pair: the smallest unit of data
- Values can be strings, numbers, arrays, objects, or other documents

Document

- A self-contained data structure represented in BSON
- Stores fields and values: Supports nesting and complex hierarchies
- Each document includes a unique ‘_id’ field
- Dynamic schema: schema is applied at the document level, not globally

Collection

- A logical grouping of documents (like tables in SQL)
- No enforced schema: documents in the same collection can have different structures

Database

- Groups collections under a common namespace
- MongoDB can host multiple databases within the same deployment

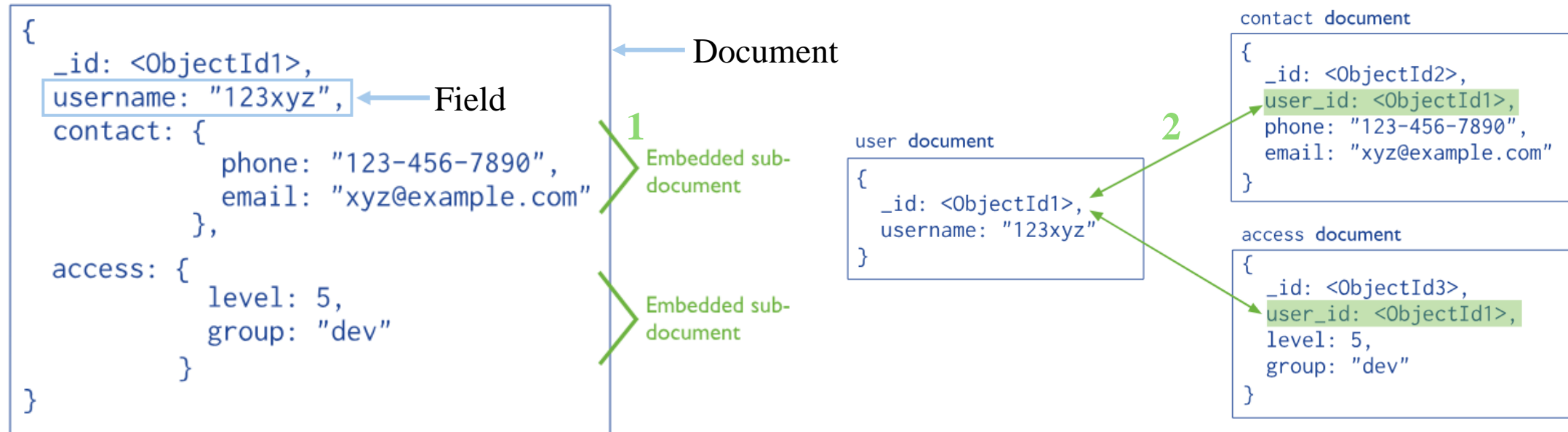
5. Data Models in MongoDB

Indexes

- Created on fields to speed up queries (single-field, compound, text, geospatial, etc.)
- MongoDB automatically indexes the ‘_id’ field by default

Embedded Documents¹ vs. References²

- Supports embedding related data inside documents for performance
- Alternatively, data can be referenced between documents if needed.



6. APIs and Client Libraries

MongoDB API

- MongoDB uses a binary protocol called MongoDB Wire Protocol
- Most access is through driver-based libraries, not direct HTTP
- MongoDB Query API defines the command syntax for CRUD operations, filtering, indexing, and aggregation
- Supports transactions, schema validation, and change streams

Client Libraries (Drivers)

- Official drivers available for: Python, Java, JavaScript/Node.js, and many more
- Drivers provide abstraction for:
 - Connecting to replica sets/shards
 - Executing queries and updates
 - Running aggregations and transactions
 - Handling retries and connection pooling

6. APIs and Client Libraries

Atlas APIs (for Cloud Users)

- MongoDB Atlas offers REST APIs and CLI tools for managing and automating tasks
- GraphQL API via MongoDB Realm allows querying with GraphQL syntax

Integration and Ecosystem

- Integration with BI tools via MongoDB Connector for BI (JDBC/ODBC)
- Integration with Kafka, Spark, Tableau, Power BI, and Grafana
- Integration with REST and GraphQL apps via Realm or custom APIs

7. Use Cases in Industry

Enterprise and Financial Systems

- Supports customer profiles, transaction histories, and audit trails
- ACID transactions in MongoDB 4.0+ allow safe multi-document operations
- Notable users: Wells Fargo, Citizens Bank, Coinbase

E-commerce and Product Catalogs

- Products often have varied attributes (e.g., size, color, brand)
- MongoDB allows storing diverse product data without rigid schemas
- Combines product info, pricing, inventory, and reviews in a single document
- Notable users: OTTO, Expedia, Albertsos

IoT and Sensor Data

- Manages time-stamped data from diverse device types
- Supports large-scale ingestion and flexible schema
- Notable users: Bosch, Enpal, Vaillant

... AND MANY MORE!

8. Set Up and Application

Installation Options

- MongoDB Community Server:
 - Available for Windows, macOS, Linux (RPM/DEB/Tarball)
 - Easy install via Homebrew, Chocolatey, or Docker images
- MongoDB Atlas:
 - Fully managed cloud service on AWS, Azure, and GCP
 - No installation required: Start with a few clicks via web UI or CLI

Configuration

- Local Deployment:
 - Configured via the 'mongod.conf' file or command-line flags
 - Includes storage path, network interface, security, and replication settings
- Atlas Deployment:
 - Configuration handled via UI, CLI, or Terraform provider
 - Supports backup policies, cluster sizing, regional deployment, etc.

8. Set Up and Application

Programming Interfaces

- Drivers and SDKs:
 - Connect using official drivers (Python, JavaScript, Java, etc.)
 - Perform CRUD operations, aggregations, indexing, and transactions
- MongoDB Shell (mongosh):
 - Command-line interface for interactive data exploration and admin tasks
 - Built-in access to `db.collection.find()`, aggregation, and scripting
- Atlas App Services / Realm:
 - Backend-as-a-Service for mobile/web apps
 - Offers GraphQL API, authentication, triggers, and serverless functions

Dev Tools and Integrations

- Compass: GUI for schema analysis, performance metrics, and ad hoc queries
- VS Code Extension, MongoDB CLI, and MongoDB Atlas UI for full developer workflows
- Easily integrates with CI/CD pipelines, BI tools, and cloud-native stacks

Thanks for your attention!

