MongoDB Architecture Guide: Overview

October 2020

Table of Contents

Industry Context

Data and software are today at the heart of every business, but for many organizations, realizing the full potential of the digital economy remains a significant challenge. Since the inception of MongoDB, we have believed the biggest challenge developers face comes from working with data:

- Demands for higher productivity and faster time to market are being held back by rigid relational data models that are mismatched to modern code and impose complex interdependencies between engineering teams.
- An inability to work with, and extract insights from, massive increases in the new and rapidly changing structured, semi-structured, and polymorphic data generated by today's applications.
- Monolithic and fragile legacy databases inhibiting the wholesale shift to distributed systems and the cloud computing that deliver the resilience and scale demanded by digital business, while also meeting a whole new set of regulatory demands for data privacy.
- Previously separate transactional, analytical, search, and mobile workloads are converging to create rich, data-driven applications and customer experiences. However each workload has traditionally been powered by its own database, creating duplicated data silos stitched together with fragile ETL pipelines, accessed by different developer APIs.

To address some of these challenges, non-tabular (sometimes called NoSQL or non-relational) databases have seen rapid adoption over the past decade. But many of these NoSQL databases are simply bandaids, offering a niche set of functionality.

The problem is that typical NoSQL databases do one or two things well – they might offer more flexibility in the data model than traditional databases, or scale-out easily. But to do this they discard the most valuable features of relational databases. They often sacrifice data integrity and the ability to work with data in the ways needed to build rich and valuable applications – whether these are new digital touchpoints with an organizations' customers, or modernized core backend business processes.

MongoDB was launched in 2009 as a completely new class of general-purpose platform, and quickly established itself as the database most wanted by developers. This is

because MongoDB takes the best of relational and NoSQL databases, providing a technology foundation that enables customers to meet the demands of almost any class of modern application.

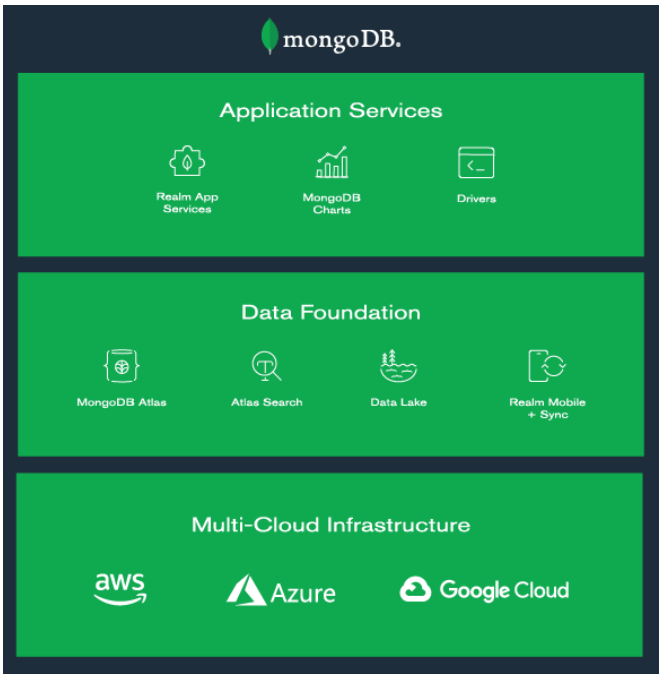

Figure 1: MongoDB Cloud Platform

Three core architectural principles underpin MongoDB and drive our R&D investments:

1. The **document data model and MongoDB Query Language**, giving developers the fastest way to innovate in building transactional, operational and analytical applications.
2. A **multi-cloud, global database**, giving developers the freedom to run their applications anywhere with the flexibility to move across private and public clouds as requirements evolve – without having to change a single line of code.
3. The **MongoDB Cloud**, providing a unified developer experience for modern applications than span cloud to edge, in database, search, and the data lake, backed by comprehensive & integrated application services.

In this Guide, we provide an overview of MongoDB's underlying architecture.

Documents and MongoDB Query Language: The Fastest Way to Innovate

Built around JSON-like documents, document databases are both intuitive and flexible for developers to work with. They promise higher developer productivity, and faster evolution with application needs. As developers have experienced these benefits, the document data model has become the most popular alternative to the tabular model used by traditional relational databases.

There are four primary advantages of the document data model.

1. Intuitive: Faster and Easier for Developers

Documents in the database directly map to the objects in your code, so they are much more natural to work with.

The following example JSON document in MongoDB demonstrates how a customer object is modeled in a single document structure with related data embedded as sub-documents and arrays. This approach collapses what would otherwise be seven separate parent-child tables linked by foreign keys in a relational database.

```
{
  "_id":
      ObjectId("5ad88534e3632e1a35a58d00"),
  "name": {
    "first": "John",
    "last": "Doe" },
  "address": [
    { "location": "work",
      "address": {
        "street": "16 Hatfields",
        "city": "London",
        "postal_code": "SE1 8DJ"},
      "geo": { "type": "Point", "coord": [
         51.5065752,-0.109081]}},
  ],
  "phone": [
    { "location": "work",
      "number": "+44-1234567890"},
     ],
  "dob": ISODate("1977-04-01T05:00:00Z"),
  "retirement_fund":
        NumberDecimal("1292815.75")
}
```

With the document data model, there is no need to decompose data across tables, run expensive JOINs, or integrate a separate ORM layer. Data that is accessed together is stored together, so you have less code to write and your users get higher performance.

2. Flexible Schema: Dynamically Adapt to Change

A document's schema is dynamic and self-describing, so you don't need to first pre-define it in the database. Fields can vary from document to document and you modify the structure at any time, allowing you to continuously integrate new application functionality, without wrestling with disruptive schema migrations.

If a new field needs to be added, it can be created without affecting all other documents in the collection, without updating a central system catalog and without taking the database offline.

When you need to make changes to the data model, the document database continues to store the updated objects without the need to perform costly `ALTER TABLE` operations, update a separate ORM middleware layer, and coordinate all of these changes across multiple developer, DBA, and Ops teams. Documents allow multiple versions of the same schema to exist in the same table space. Old and new applications can co-exist.

MongoDB also offers Schema Validation so you can enforce standards over each document's structure. With schema validation, you have control to apply data governance to a document schema when your application enters production, while maintaining the benefits of a flexible data model in development.

3. Universal: JSON Documents are Everywhere

Lightweight, language-independent, and human readable, JSON has become an established standard for data communication and storage. Documents are a superset of all other data models so you can structure data any way your application needs – rich objects, key-value pairs, tables, geospatial and time-series data, and the nodes and edges of a graph.

As a result of these properties, you can serve many more classes of application with a single database. You can work with documents using a single query language, giving you a consistent development experience however you've chosen to model your data.

MongoDB stores data as JSON (JavaScript Object Notation) documents in a binary representation called BSON (Binary JSON). Unlike most databases that store JSON data as primitive strings and numbers, the BSON encoding extends the JSON representation to include additional types such as int, long, date, floating point, and decimal128. This makes it much easier for applications using MongoDB to reliably process, sort, and compare data.

4. Powerful: Serve any Workload

An important difference between databases is the expressivity of the query language, richness of indexing, and data integrity controls.

The MongoDB Query Language is comprehensive and expressive. Ad hoc queries, indexing, and real time aggregations provide powerful ways to access, group, transform, and analyze your data. You can federate queries across your databases supporting transactional workloads and archived data in your data lake using the same MongoDB Query Language and drivers, all using a single connection string.

With ACID transactions you maintain the same all-or-nothing and snapshot isolation guarantees you're used to in relational databases, whether manipulating data in a single document, or with MongoDB's scale-out architecture, across multiple documents geographically distributed in multiple shards.

With strong data consistency, MongoDB eliminates the application complexity imposed by eventually consistent NoSQL systems. MongoDB's consistency guarantees are fully tunable, allowing you to balance data freshness against performance.

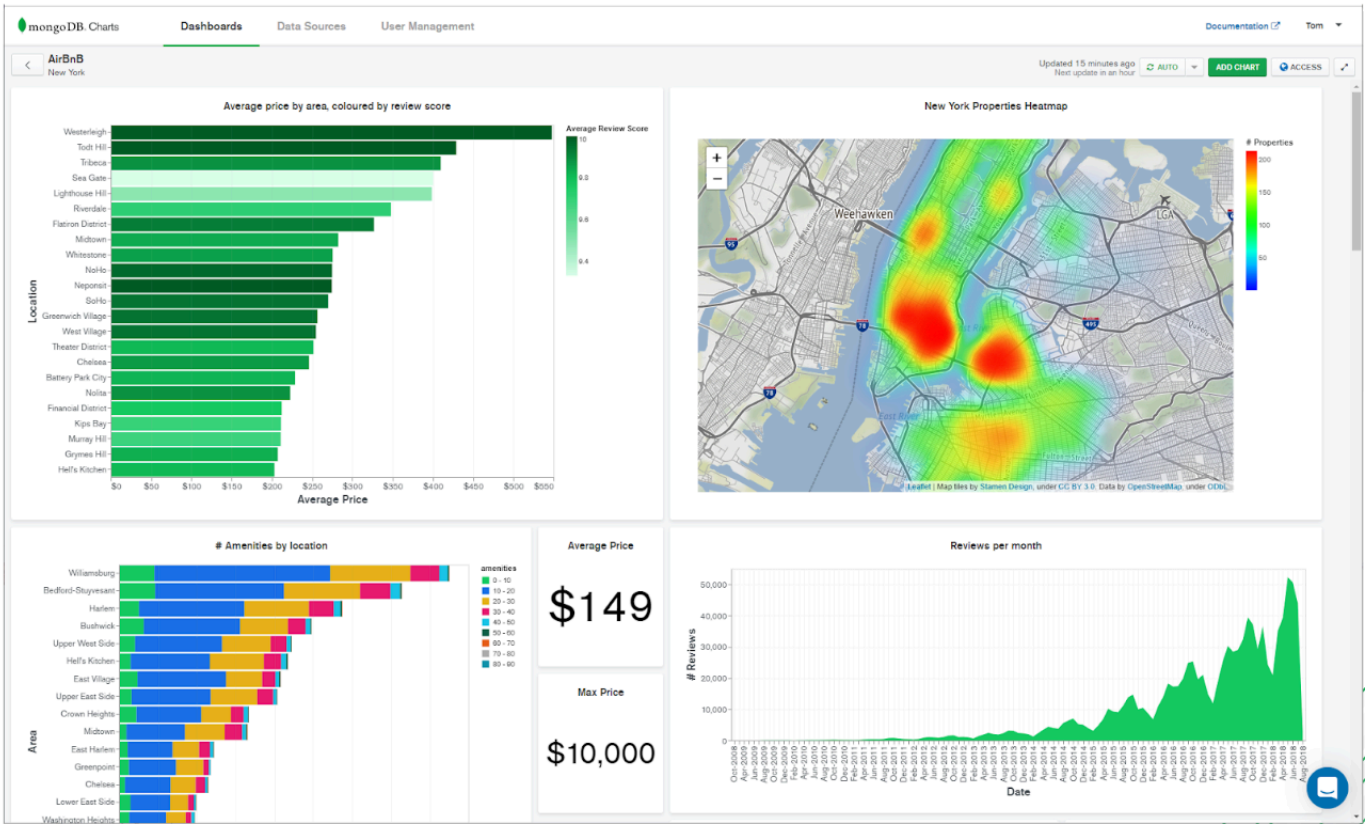

Figure 2: Creating rich visualizations of your data with MongoDB Charts

Working with Document Data

To accelerate developer productivity, MongoDB provides native drivers for all popular programming languages and frameworks. Supported drivers include Java, Javascript, C#/.NET, Go, Python, PHP, Scala, Rust and more. All supported MongoDB drivers are designed to be idiomatic for the given programming language. This makes it much more natural for developers to work with data than string-based languages like SQL, and eliminates the need for cumbersome and fragile ORM abstraction layers.

You can also interact with MongoDB graphically using MongoDB Compass, the GUI for MongoDB. Through Compass you can explore and manipulate your data, visually create queries and aggregation pipelines from the GUI and then export them as code to your app; view and create indexes; build schema validation rules; and more.

Beyond working with documents through the MongoDB drivers, you need to make data accessible to multiple consumers across the organization so they can extract insights and hidden value in your data. MongoDB provides a range of visualization tools and connectors to make this straightforward:

- MongoDB Charts is the fastest and easiest way to create visualizations of richly-structured JSON data. You can create graphs and build dashboards, sharing them with other users for collaboration, and embed them directly into your web apps to create engaging user experiences.
- The MongoDB Connector for BI lets you use MongoDB as a data source for your existing SQL-based BI and analytics platforms such as Tableau, Microstrategy, Looker, and more.
- The MongoDB Connector for Apache Spark exposes all of Spark's libraries, including Scala, Java, Python and R. MongoDB data is materialized as DataFrames and Datasets for analysis with machine learning, graph, streaming, and SQL APIs.

To make it easy for businesses to act on data in real time, many developers are building fully reactive, event-driven data pipelines. MongoDB goes beyond many other databases with features like Change Streams that

automatically detect and notify consuming applications of any data modifications in the database, while MongoDB Atlas Triggers allow you to execute server-side logic in response to database events. With the MongoDB Connector for Apache Kafka, you can build robust data pipelines that move events between systems in real time, using MongoDB as both a source and sink for Kafka. The connector is supported by MongoDB and verified by Confluent.

In summary, documents are the best way for developers to work with data, and MongoDB gives you the most productive and fully-featured implementation of a document database anywhere.

Multi-Cloud Global Database: Freedom and Flexibility

To harness the tremendous rate of innovation in the cloud and reduce the risk of lock-in, project teams should build their applications on data platforms that deliver a consistent experience across any environment. MongoDB can be run anywhere – from developer laptops to mainframes, from private clouds to the public cloud.

Broadest Reach: Private, Hybrid, Public Clouds

With MongoDB, the developer experience is entirely unaffected by your chosen deployment model. This enables you to take advantage of unique capabilities in each platform without changing a single line of application code and without the heavy lift and risk of complex database migrations.

MongoDB Atlas: Fully Automated Database as a Service

MongoDB Atlas is the global cloud database service for modern applications. You can deploy fully managed MongoDB across AWS, Azure, or Google Cloud with best-in-class automation and proven practices that guarantee availability, scalability, and compliance with security standards.

MongoDB Atlas is available through a pay-as-you-go model and billed on an hourly basis. It's easy to get started – use a simple GUI or programmatic API calls to select the public cloud provider, region, instance size, and features you need. MongoDB Atlas provides:

- Automated database and infrastructure provisioning along with auto-scaling so teams can get the database resources they need, when they need them, and can elastically scale in response to application demands.
- Always-on security features to protect your data, including network isolation, fine-grained access controls, auditing, and end-to-end encryption down to the level of individual fields.
- Certifications with global standards to help you achieve your compliance requirements, including ISO 27001, SOC 2, and more. Atlas can be used for workloads subject to HIPAA, PCI-DSS, or the GDPR.
- Built-in replication both within and across regions for always-on availability, even in the face of complete regional outages.
- Global Clusters for fully managed, globally distributed databases that provide low latency, responsive reads and writes to users anywhere, with strong data sovereignty controls for regulatory compliance.
- Fully managed backups with point-in-time recovery to protect against data corruption, and the ability to query backups in-place without full restores.
- Fine-grained monitoring, real-time metrics, query profiler, and customizable alerts for comprehensive performance visibility.
- Intelligent schema and index recommendations with the Performance Advisor, which analyzes slow query logs of your database collections and ranks suggestions by impact to your database performance.
- Automated patching and single-click upgrades for new major versions of the database, enabling you to take advantage of the latest MongoDB features.
- Auto-archiving of aged data from your live database clusters to fully-managed cloud object storage with Online Archive. Federated query enables you to analyze your live MongoDB Atlas data and historical data on object storage together and in-place, with a single query, for faster insights. Review the Atlas Data Lake

section later in this guide to learn more.

- Live migration to move your self-managed MongoDB database into the Atlas service or to move Atlas databases between cloud providers.

MongoDB Atlas is serving a vast range of workloads for startups, Fortune 500 companies, and government agencies, including mission-critical applications handling highly sensitive data in regulated industries. Built and run by the same team that develops the database, MongoDB Atlas is the best way to run your MongoDB applications.

MongoDB Run by You, with Tools from Us

If you need to run the database on your own self-managed infrastructure for business or regulatory requirements, MongoDB offers on-premises management tools. These tools can be used to power a MongoDB database behind a single application, or to build your own private database service and expose it to your development teams.

MongoDB Ops Manager is the simplest way to run MongoDB on premises or in a private cloud, making it easy for operations teams to automate deployment, monitoring, backup, and scaling of MongoDB. Through Ops Manager, you can manage the complete lifecycle of your MongoDB databases via a powerful GUI, or programmatically with APIs to enable integration with your Infrastructure as a Code tools.

Kubernetes is the industry leading container orchestration platform. It provides you with a consistent automation and management experience anywhere from on-premises infrastructure to the public cloud. Kubernetes users can use the MongoDB Enterprise Operator for Kubernetes that integrates with MongoDB Ops Manager to automate and manage MongoDB clusters. You have full control over your MongoDB deployment from a single Kubernetes control plane. You can use the operator with upstream Kubernetes, or with any popular distribution such as Red Hat OpenShift and Pivotal Container Service (PKS).

With the MongoDB Atlas Service Broker, you can spin up and manage Atlas clusters directly from Kubernetes, controlling infrastructure provisioning, database setup, global distribution, and more. Compatible with the Open Service Broker API, the Atlas Service Broker makes MongoDB Atlas part of your Kubernetes service catalog, allowing cluster creation and management directly from Kubernetes.

Distributed Architecture: Scalable, Resilient and Mission Critical

Through replica sets and native sharding, MongoDB enables you to scale out your applications with always-on availability. You can distribute data for low latency user access, while enforcing data sovereignty controls for data privacy regulations such as the GDPR.

Availability and Data Protection with Replica Sets

MongoDB replica sets enable you to create up to 50 copies of your data, which can be provisioned across separate nodes, data centers, and geographic regions.

Replica sets are predominantly designed for resilience: if a primary node suffers an outage or is taken down for maintenance, the MongoDB cluster will automatically elect a replacement in a few seconds, switching over client connections and retrying any failed operations for you. In MongoDB Atlas, uptime is backed by a 99.995% Service Level Agreement.

The replica set election process is controlled by sophisticated algorithms based on an extended implementation of the Raft consensus protocol. Before a secondary replica is promoted, the election algorithms evaluate a range of parameters including:

- Analysis of election identifiers, timestamps, and journal persistence to identify those replica set members that have applied the most recent updates from the primary replica.
- Heartbeat and connectivity status with the majority of other replica set members.
- User-defined priorities assigned to replica set members.

Extending data protection, developers can configure replica sets to provide tunable, multi-node durability, and geographic awareness. Through MongoDB's write concern, you can ensure write operations propagate to a majority of replicas in a cluster. This minimizes the risk of data being rolled back following the election of a new primary replica. You can also create custom write concerns that target

specific members of a replica set, deployed locally and in remote regions. This ensures writes are only acknowledged once custom policies have been fulfilled, such as writing to at least a primary and replica in one region and at least one replica in a second region. This reduces the risk of data loss in the event of a complete regional failure.

Beyond resilience, replica sets can also be used to scale read operations, intelligently routing queries to a copy of the data that is physically closest to the user. With sophisticated policies such as hedged reads, the cluster will automatically route queries to the two closest nodes (measured by ping distance), returning results from the fastest replica. This helps minimize queries waiting on a node that might otherwise be busy, reducing 95th and 99th percentile read latency. Note that hedged reads are available in shared clusters only.

Scale-Up, Scale-Out, Scale Across Storage Tiers

Like most databases, you can scale MongoDB vertically by moving up to larger instance sizes. As a distributed system, MongoDB can perform a rolling restart of the replica set to enable you to move between different instances without application downtime. In MongoDB Atlas, cluster auto-scale can adjust both compute and storage in response to application load, avoiding you having to monitor utilization and react to scaling needs.

Through native sharding, MongoDB can also scale your database out across multiple nodes to handle write-intensive workloads and growing data sizes. Sharding with MongoDB allows you to seamlessly scale the database as your applications grow beyond the hardware limits of a single server, and it does so without adding complexity to the application.

To respond to evolving workload demands, you can add and remove shards at any time. You also have the flexibility to refine your shard key on-demand, without impacting system availability. As your shard key is modified or as you change the cluster topology, MongoDB will automatically rebalance data across shards as needed without manual intervention.

By simply hashing a primary key value, many distributed databases randomly spray data across a cluster of nodes, imposing performance penalties when data is queried, or adding application complexity when data needs to be localized to a specific region. By exposing multiple sharding policies to developers, MongoDB offers a better approach. Data can be distributed according to query patterns or data placement requirements, giving you much higher scalability across a more diverse set of workloads:

- **Ranged Sharding**. Documents are partitioned across shards according to the shard key value. Documents with shard key values close to one another are likely to be co-located on the same shard. This approach is well suited for applications that need to optimize range based queries, such as co-locating data for all customers in a specific region on a specific set of shards.
- **Hashed Sharding**. Documents are distributed according to an MD5 hash of the shard key value. This approach guarantees a uniform distribution of writes across shards, which is often optimal for ingesting streams of time-series and event data.
- **Zoned Sharding**. Provides the ability for developers to define specific rules governing data placement in a sharded cluster.

Global Clusters in MongoDB Atlas allows you to quickly implement zoned sharding using a visual UI or the Atlas API. Easily create distributed databases to support geographically distributed apps, with policies enforced for data sovereignty. Each zone is part of the same, single cluster and can be queried globally, but data is pinned to shards in specific regions based on data locality policies.

Beyond vertical and horizontal scaling, MongoDB also offers tiered-scaling. When working in the cloud, MongoDB Atlas Online Archive will automatically tier aged data out of the database onto cloud object storage in the Atlas Data Lake. Archived data remains fully accessible with federated queries that span both object and database storage in a single connection string. This approach enables you to more economically scale data storage by moving it to a lower cost storage tier without losing access to the data, and without grappling with slow and complex ETL pipelines.

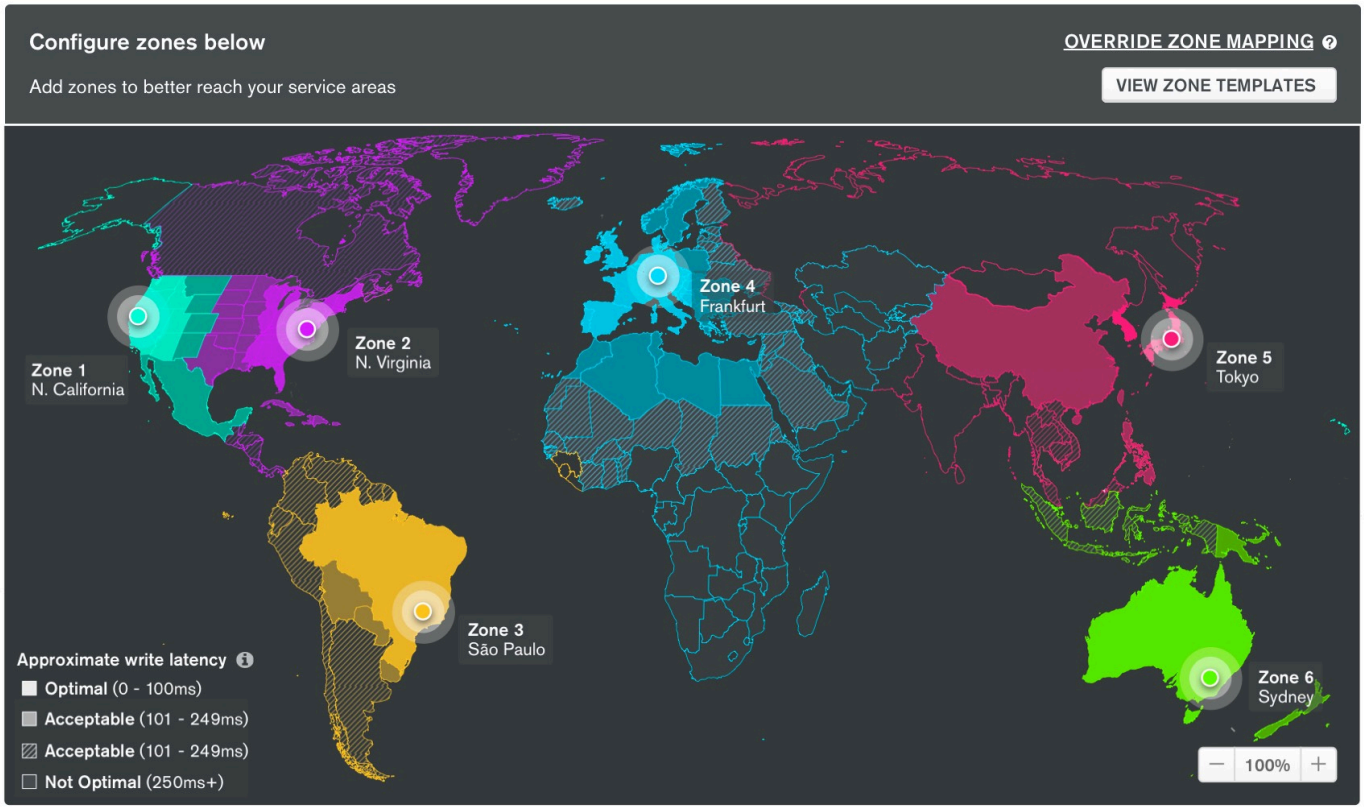

Figure 3: Serving always-on, globally distributed, write-everywhere apps with MongoDB Atlas Global Clusters

Privacy and Security

With the digital economy becoming so essential for economic prosperity, it's no surprise that governments and enterprises around the world are responding to growing public concern for the safety of personal data.

MongoDB features extensive capabilities to defend, detect, and control access to data:

- **Authentication**. Simplifying access control to the database, MongoDB offers a strong Challenge-Response mechanism based on SCRAM-256, along with integration to enterprise security infrastructure including LDAP, Windows Active Directory, Kerberos, x.509 certificates, and AWS IAM.
- **Authorization**. Role-Based Access Control (RBAC) enables you to configure granular permissions for a user or an application based on the privileges they need to do their job.
- **Auditing**. For regulatory compliance, security administrators can use MongoDB's native audit log to record all database activity and changes.
- **Network Isolation**. MongoDB Atlas users' data and underlying systems are fully isolated from other users. Database resources are associated with a user group, which is contained in its own Virtual Private Cloud (VPC). Access must be granted by IP whitelisting or VPC Peering.
- **Encryption Everywhere**. MongoDB data can be encrypted while in motion across the network, while in use in database memory, and while at rest, whether on disk or in backups.

MongoDB's Client-Side Field Level Encryption (FLE) provides amongst the strongest levels of data privacy and security for regulated workloads.

With Client-Side FLE your most sensitive data is automatically encrypted before leaving the application, and so the database only ever works with it as ciphertext. Whether resident in memory, in system logs, at-rest in storage, and in backups – all protected data remains unreadable.

Client-Side FLE complements existing network and storage encryption to protect the most highly classified, sensitive fields of your records **without**:

- Developers needing to write additional, highly complex encryption logic
- Compromising your users' ability to query encrypted data
- Significantly impacting database performance

By securing data with Client-Side FLE you can move to managed services in the cloud with greater confidence. This is because the database only works with encrypted fields, and you control the encryption keys, rather than having the database provider manage the keys for you. This additional layer of security enforces an even finer-grained separation of duties between those who use the database and those who administer and manage the database.

Client-Side FLE also enables you to more easily comply with "right to erasure" mandates in modern privacy legislation. When a user invokes their right to erasure, you simply destroy the associated encryption key and the user's Personally Identifiable Information (PII) is rendered **unreadable** and **irrecoverable** to anyone.

You can learn more by downloading our Guide to Client-Side Field Level Encryption.

MongoDB Cloud

Modern data architectures are not limited to the transactional database. Many applications also require analytics and search functionality, which often requires teams to learn, deploy, and manage additional systems. If you're building mobile apps, you'll need to deal with data on the device and syncing it to the backend. You may also find yourself building data visualizations, writing a lot of glue code to move data between data services, or creating and operating custom data access APIs.

Only through a consistent, unified experience – for both developers and the operations teams supporting them – can companies avoid a sprawl of disparate data silos, each with their own set of APIs, operational models, and security requirements.

This is the issue the MongoDB Cloud is uniquely designed to solve.

Why Use MongoDB Cloud?

In MongoDB Cloud, the database is fully integrated with other data services with **automatic syncing, data tiering, and federated query**. Search indexes run alongside the database and are automatically kept in sync. Aged data can be auto-archived to cloud storage, providing fully managed data tiering while retaining access. Queries are automatically routed to the appropriate data tier without requiring you to think about data movement, replication, or ETL. MongoDB Cloud can even automatically sync backend data to an embedded database on mobile devices.

Across these services, MongoDB provides a **consistent and elegant developer experience**. From database to search to analytics on your data lake, there's a common way of working with data that simplifies development. Unlike other cloud data platforms that require you to learn entirely different technologies and APIs, MongoDB Cloud presents all its services in a single system and with a consistent interface.

MongoDB Cloud runs on **multi-cloud infrastructure**. You can choose underlying infra from AWS, GCP, or Azure and get a consistent experience, deploying and controlling clusters in different clouds from a single UI. Avoid lock-in or take advantage of different cloud vendor's services with simple data portability across clouds.

MongoDB Cloud is **fully managed for operational simplicity**. From the basics of deployment automation and monitoring to advanced features like auto-scale and intelligent performance advice, MongoDB Cloud gets operations out of your way. Clusters are fully managed, data synchronization between services is automated, and making adjustments is as easy as a button click or API call.

MongoDB Cloud **works with your ecosystem**. Manage your infrastructure as code with Kubernetes and Terraform integrations, plug in your monitoring and alerting tools, integrate with your security environment, connect to data

tools like Kafka and Spark, and work with MongoDB integrations in your usual IDEs.

What's in MongoDB Cloud?

Central to MongoDB Cloud is **MongoDB Atlas**. MongoDB Cloud extends Atlas with other data services that work with it seamlessly, giving you more ways of working with data.

MongoDB Atlas Search

Atlas Search makes it easy to create fast, relevant, full-text search capabilities on top of your data in the cloud, and is built on top of Apache Lucene, the industry standard library.

The service is fully managed, operationally invisible, and integrated within the Atlas cloud database, removing the need for teams to deploy and manage a separate search platform. Search functionality is also exposed via the MongoDB query language so developers do not need to learn a new API. Simply create search indexes directly in Atlas and use the MongoDB query language to build sophisticated search queries.

Supported search capabilities include fuzzy search, autocomplete, facets and filters, custom scoring, analyzers for 30+ languages, and more. Atlas Search is available for all Atlas database clusters running MongoDB 4.2 or higher.

MongoDB Atlas Data Lake

Atlas Data Lake brings a serverless, scalable data lake to the cloud platform with an on-demand query service that enables you to analyze data in cloud object storage (Amazon S3) in-place using the MongoDB Query Language (MQL).

There is no infrastructure to set up or manage and no need to capacity plan as Atlas Data Lake automatically parallelizes operations by breaking queries down and then dividing the work across multiple compute nodes. Atlas Data Lake can also automatically optimize your workloads by utilizing compute in the region closest to your data. This is useful for data sovereignty-related needs, granting you the ability to specify which region your data needs to be processed in, and giving you a global view of data with greater security.

Support for federated queries allows you to combine and analyze data across S3 and your Atlas database clusters together with a single query. In addition, you can easily persist the results of your aggregations to either object storage or your cloud database. Supported data formats include JSON, BSON, CSV, TSV, Avro, ORC and Parquet.

Atlas Data Lake is also the technology underpinning the Atlas Online Archive, which allows you to automatically move historical data out of your database to cloud object storage while retaining query access through the same connection string.

MongoDB Realm for Data at the Edge

The Realm Mobile Database extends your data foundation out to the edge of the network, and is fully integrated with the MongoDB Cloud. Realm is a lightweight database embedded directly on the client device. Realm helps solve the unique challenges of building for mobile, making it simple to store data on-device and enabling data access even when offline.

Realm Sync is seamlessly integrated and keeps data up-to-date across devices and users by automatically, bi-directionally syncing data between the client and a backend Atlas database cluster.

Realm Application Services go further, simplifying the code required to stand up both mobile and web applications. Realm's SDKs give developers the tools needed to access data stored in MongoDB Atlas directly from the client, and interact with the platform's application services.

Conclusion and Next Steps

Every industry is being transformed by data and digital technologies. As you build or remake your company for a digital world, speed matters – measured by how fast you build applications, how fast you scale them, and how fast you can gain insights from the data they generate. These are the keys to applications that provide better customer experiences, enable deeper, data-driven insights, and make new products or business models possible.

MongoDB enables you to meet the demands of modern apps with a data platform built on three core architectural foundations:

1. The **document data model and MongoDB Query Language**, giving developers the fastest way to innovate in building transactional, operational and analytical applications.
2. A **multi-cloud, global database**, giving developers the freedom to run their applications anywhere with the flexibility to move across private and public clouds as requirements evolve – without having to change a single line of code.
3. The **MongoDB Cloud**, providing a unified developer experience for modern applications than span cloud to edge, in database, search, and the data lake, backed by comprehensive & integrated application services.

In this guide we have explored the foundational concepts that underpin the architecture of MongoDB. Other guides on topics such as performance, operations, and security best practices can be found at mongodb.com.

You can get started now with MongoDB by:

1. Reviewing the Use Case Guidance White Paper to identify applicable use cases for MongoDB.
2. Spinning up a fully managed MongoDB cluster on the Atlas free tier or downloading MongoDB for local development
3. Reviewing the MongoDB manuals and tutorials in our documentation

Safe Harbor

The development, release, and timing of any features or functionality described for our products remains at our sole discretion. This information is merely intended to outline our general product direction and it should not be relied on in making a purchasing decision nor is this a commitment, promise or legal obligation to deliver any material, code, or functionality.

We Can Help

We are the company that builds and runs MongoDB. Over 20,200 organizations rely on our commercial products. We offer cloud services and software to make your life easier:

MongoDB Atlas is the global cloud database service for modern applications. Deploy fully managed MongoDB across AWS, Azure, or Google Cloud with best-in-class automation and proven practices that guarantee availability, scalability, and compliance with security standards.

MongoDB Enterprise Advanced is the best way to run MongoDB on your own infrastructure. It's a finely-tuned package of advanced software, support, certifications, and other services designed for the way you do business.

MongoDB Atlas Data Lake allows you to quickly and easily query data in any format on Amazon S3 using the MongoDB Query Language and tools. You don't have to move data anywhere, you can work with complex data immediately in its native form, and with its fully-managed, serverless architecture, you control costs and remove the operational burden.

MongoDB Charts is the best way to create, share and embed visualizations of MongoDB data. Build visualizations quickly and easily to analyze complex, nested data. Embed individual charts into any web application or assemble them into live dashboards for sharing.

Realm Mobile Database allows developers to store data locally on iOS and Android devices using a rich data model that's intuitive to them. Combined with the MongoDB Realm sync-to-Atlas, Realm makes it simple to build reactive, reliable apps that work even when users are offline.

MongoDB Realm allows developers to validate and build key features quickly. Application development services like Realm Sync for mobile and Realm's GraphQL service, can be used with Realm Functions, Triggers, and Data Access Rules – simplifying the code required to build secure and performant apps.

MongoDB Cloud Manager is a cloud-based tool that helps you manage MongoDB on your own infrastructure. With automated provisioning, fine-grained monitoring, and

continuous backups, you get a full management suite that reduces operational overhead, while maintaining full control over your databases.

MongoDB Consulting packages get you to production faster, help you tune performance in production, help you scale, and free you up to focus on your next release.

MongoDB Training helps you become a MongoDB expert, from design to operating mission-critical systems at scale. Whether you're a developer, DBA, or architect, we can make you better at MongoDB.

Resources

For more information, please visit mongodb.com or contact us at sales@mongodb.com.

Case Studies (mongodb.com/customers)
Presentations (mongodb.com/presentations)
Free Online Training (university.mongodb.com)
Webinars and Events (mongodb.com/events)
Documentation (docs.mongodb.com)
MongoDB Atlas database as a service for MongoDB (mongodb.com/cloud)
MongoDB Enterprise Download (mongodb.com/download)
MongoDB Realm (mongodb.com/realm)

mongoDB

US 866-237-8815 ▪ INTL +1-650-440-4474 ▪ info@mongodb.com
© 2020 MongoDB, Inc. All rights reserved.