Use Case Guidance:

Selecting the best apps for MongoDB, and when to evaluate other options

February 2020
# Table of Contents

**Introduction**  
When Should I Use MongoDB?  
MongoDB Database & Atlas: OLTP & Online Analytics  
MongoDB Atlas Data Lake: Offline Analytics  
The MongoDB Data Platform  

**Key Strategic Initiatives Supported by MongoDB**  
Legacy Modernization  
Cloud Data Strategy  
Data as a Service (DaaS)  
Business Agility  

**Use Cases for MongoDB**  
Single View  
Customer Data Management and Personalization  
Internet of Things (IoT) and Time-Series Data  
Product Catalogs and Content Management  
Payment Processing  
Mobile Apps  
Mainframe Offload  
Online Analytics and AI  
Storage Tiering for Online Analytics  
Offline (Long Running) Analytics  
Enter MongoDB Atlas Data Lake  
Unlocking the Data Lake  
MongoDB Atlas Data Lake Use Cases  

**When Should I Not Use MongoDB?**  
Common Off-the-Shelf Software Built for Relational Databases  
Ad-Hoc Reporting in the MongoDB Server: Unindexed Queries  
Data Warehouse Replacements with Atlas Data Lake  

**Conclusion**  
We Can Help  
Resources
Introduction

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:

- Demands for higher developer productivity and faster time to market – where release cycles are compressed to days and weeks – are being held back by rigid relational data models and traditional waterfall development practices.
- An inability to manage and extract insights from massive increases in new and rapidly changing data types – structured, semi-structured, and polymorphic data – generated by modern web, mobile, social, AI, and IoT applications.
- Difficulty in exploiting the wholesale shift to distributed systems and cloud computing. These new platforms provide developers access to on-demand, elastically scalable compute and storage infrastructure and managed services, while meeting a whole new set of regulatory demands for data privacy.

For these reasons, non-tabular (sometimes called NoSQL or non-relational) databases have seen rapid adoption, with MongoDB firmly established as the most popular. Architecture and development teams often get started with MongoDB in tactical projects, using the experiences they gain as a proving ground for a new class of data platform. As they start to realize the value MongoDB offers, many ask where else they can apply the technology to other projects in the business.

This white paper guides you to applicable use cases, and those workloads where you should evaluate alternative solutions.

When Should I Use MongoDB?

The MongoDB data platform can be used across a range of OLTP and analytical apps. Through its design, MongoDB provides a technology foundation that enables development teams to meet the demands of modern apps, working with data wherever it lives: on device, in the application backend, and in the data lake.

Three core architectural principles underpin MongoDB:

1. The document data model – presenting you the best way to work with data.
2. A distributed systems design – allowing you to intelligently put data where you want it.
3. The freedom to run anywhere – allowing you to future-proof your apps and eliminate vendor lock-in.

With the MongoDB Database and MongoDB Atlas Data Lake you can address a wide range of application requirements.
MongoDB Database & Atlas: OLTP & Online Analytics
The MongoDB Database is a general purpose database designed to serve most OLTP and online analytics workload:

- Wherever you are thinking about using a relational database, you should consider MongoDB.
- Wherever you are thinking about using a NoSQL database, you should consider MongoDB.

Whether you plan to run your apps in your own facilities, as a serverless, cloud-native solution, or a hybrid deployment model in between, the MongoDB Database provides complete infrastructure agility. You can run MongoDB yourself on your own infrastructure, or use MongoDB Atlas, our fully managed and fully automated global cloud database service available on ~ 70 regions across AWS, Azure, and GCP.

MongoDB Atlas Data Lake: Offline Analytics
Beyond the MongoDB Database, the MongoDB Atlas Data Lake (beta) extends the power and productivity of MongoDB to offline analytics workloads. The Atlas Data Lake allows you to quickly and easily query data in any format on Amazon S3 (with other cloud providers coming) using the same MongoDB Query Language and tools used by the database.

With Atlas Data Lake you can realize the value of your data lake faster: 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.

The MongoDB Data Platform
Through the MongoDB Database and Atlas Data Lake, you have an integrated data platform that shares the same query language and tools, fully managed for you by the cloud-native Atlas service, enabling you to serve a broad range of operational and analytical applications:

1. MongoDB Atlas: designed for online OLTP applications requiring millisecond response times, provisioned with dedicated analytics nodes for real time analytics against live, operational data (a pattern called Translytics).
2. MongoDB Atlas: Online analytics with optimized secondary indexes to support known and predictable query patterns requiring latency in the range of subsecond to several-second response times.
3. MongoDB Atlas Data Lake: Designed for offline analytics against large volumes of data residing in cloud storage, with latency in the range of seconds (at the very least), but more typically minutes and more – dictated by how much data you are querying and how it is partitioned.

You can move data between each Atlas tier, exposing it to your operational and analytical tiers as needed.
Key Strategic Initiatives Supported by MongoDB

Finding new ways to compete in the digital economy involves much more than simply inserting new technology into your application stack. Fully harnessing the opportunities presented by digital requires bringing together **people, processes, and platform technologies** to support your organization’s strategic initiatives. Underpinned by MongoDB’s data platform, we have developed a suite of solution stacks that provide:

- **Advisory consulting** to understand your strategic objectives and build a roadmap to deliver them.
- **Program governance** for delivery controls throughout a project.
- **Application lifecycle expertise** supported by design patterns and reference architectures, implementation best practices, and technical training.

From projects to modernize legacy apps, to moving to the cloud, exposing enterprise data as a service, or enabling business agility, MongoDB solutions can help you address your organization’s most transformational strategic initiatives.

**Figure 1: MongoDB Professional Services solutions stacks**

Legacy Modernization

Legacy Modernization enables you to apply the latest innovations in development methodologies, architectural patterns, and technologies to refresh your portfolio of legacy applications.
We work with you to build an Application Modernization Factory (AMF), cultivated from best practices developed with some of the world’s largest organizations. We partner with your teams to accelerate the assessment, prioritization, and redesign of legacy apps, quantifying the economic value of change and providing a roadmap for delivery. We work with you through the modernization efforts of redevelopment, consolidation, and optimization, harnessing patterns and technologies such as agile and DevOps, microservices, cloud computing, and MongoDB best practices.

Review our Legacy Modernization overview to learn more, and download the Relational Database Migration Guide for best practices in schema design, application development, and data migration when moving from relational databases to MongoDB.

**Cloud Data Strategy**

Companies have long realized the agility and cost benefits of running on cloud infrastructure instead of maintaining their own data centers. However, a successful Cloud Data Strategy is much more than just using someone else’s computer. To derive the agility and cost benefits that the cloud promises, you need a comprehensive approach that relies on using the right technologies and operational processes.

MongoDB is a cloud-native data platform that can meet you wherever you are on your cloud journey:

- **MongoDB Atlas** is our global, fully managed, on-demand cloud service for MongoDB, available on AWS, Microsoft Azure, and GCP.
- **MongoDB Ops Manager** provides all of the operational tooling you need to build your own private or hybrid cloud environment.
- **MongoDB Realm** is the serverless platform from MongoDB. It is a reliable and scalable backend that makes it simple to integrate with data and services. It streamlines development with a single syntax and data format from database to frontend app – freeing you to create better apps faster than ever.

Harnessing MongoDB’s cloud services, the MongoDB Cloud Factory helps your organization take a cloud-first stance on MongoDB application development. We collaborate with your teams to develop a Cloud Operating Model, serving as a foundation for both development of new applications in the cloud and migrating existing workloads. We analyze your application portfolio to rapidly and iteratively identify applications most suitable for running in the cloud and provide technical expertise and best practices throughout the application development lifecycle.

Review our Cloud Data Strategy overview to learn more, and download the MongoDB Atlas Best Practices white paper for guidance on capacity planning, security, and performance optimization in the cloud.
Data as a Service (DaaS)

DaaS is an investment in consolidating and organizing your enterprise data in one place, then making it available to serve new and existing digital initiatives. Data as a Service becomes a system of innovation, exposing data as a cross-enterprise asset. It unlocks data from legacy systems to drive new applications, without the need to disrupt existing backends. Typical use cases include customer single view, analytics and AI, and mainframe offloading.

The path to Data as a Service is to implement an Operational Data Layer (ODL). This data layer sits in front of legacy systems, enabling you to meet challenges that the existing platforms can’t handle – without a full “rip and replace” of those existing systems.

Figure 2: Reference architecture for an Operational Data Layer

MongoDB has developed a tried and tested approach to constructing an Operational Data Layer. The Data Layer Realization methodology helps you unlock the value of data stored in silos and legacy systems, driving rapid, iterative integration of data sources for new and consuming applications. Data Layer Realization offers the expert skills of MongoDB’s consulting engineers, but also helps develop your own in-house capabilities, building deep technical expertise and best practices.

Review our Data as a Service overview to learn more, and download the ODL Reference Architecture for best practices in implementing an Operational Data Layer.

Business Agility

Organizations that are equipped to respond quickly to new opportunities, threats, and changing regulatory conditions will solidify their competitive positions and gain market share.
You need to build new applications and functionality to address emerging and changing use cases better and faster than ever before.

MongoDB’s Innovation Accelerator helps achieve widespread transformation of your development organization and processes to achieve faster time to market across multiple business applications and use cases. Our Professional Services team works with you to re-equip your development teams, supporting agile application development and continuous delivery of new functionality. The Innovation Accelerator:

- Establishes a roadmap for achieving business agility.
- Ensures rapid prototyping of solutions.
- Builds deep technical expertise and best practices across your development teams.
- Supports agile and iterative development, testing, and deployment.

Review our Business Agility overview to learn more.

**Use Cases for MongoDB**

Through the solutions discussed above, MongoDB serves a broad range of operational and analytical applications. In each of the following use cases, we provide a definition of the application, along with required capabilities and how MongoDB can help. Review the MongoDB Architecture Guide for more detail on specific MongoDB features and capabilities applicable across all of these use cases.

**Single View**

A Single View, sometimes called Customer 360 or a data hub, aggregates data from multiple source systems into a central repository to create a single view of a business entity. By creating a single, real-time view, organizations enhance business visibility and enable new classes of analytics to better serve their customers and improve oversight of key resources.

While the most common use case is building a single view of your customers, the same approach can be applied to create a single view of supply chains, financial asset classes, products, and more.

**Customer example:** As the UK’s fastest growing electronics retailer, AO.com was challenged to maintain business expansion without losing touch with its customers. It turned to MongoDB, to build a single customer view, delivering the project in just 3 months. With its single view of the customer, the business has been able to reduce call handling times by 40%, cut fraud processing from hours to seconds, and enabled its legal and marketing teams to better support new GDPR requirements.
<table>
<thead>
<tr>
<th>Required Capabilities</th>
<th>Why MongoDB?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data model</strong></td>
<td><strong>Data model</strong></td>
</tr>
</tbody>
</table>
| ● Ingest data of any structure from source systems  
  ● Dynamically adapt as a source systems schema changes | ● Document data model to store rich, multi-structured data  
  ● Flexible schema with no schema migrations  
  ● MongoDB Connector for Apache Kafka to stream data changes from source systems to the single view |
| **Query model**       | **Query model** |
| ● Multiple access patterns and query types  
  ● Simple lookups through to sophisticated analytics for business insight and personalization | ● Expressive query language, secondary indexing, aggregation pipeline, full-text search, and on-demand materialized views  
  ● Data visualization: MongoDB Charts and BI Connector  
  ● AI: Spark Connector, idiomatic R and Python drivers  
  ● MongoDB Data Lake for offline analytics of your data in S3, using the same query language and tools as your OLTP database |
| **Grow and protect**  | **Grow and protect** |
| ● Scale as new data sources are on-boarded  
  ● Robust security and data sovereignty controls | ● Distributed systems architecture with native sharding for scale out  
  ● Comprehensive access controls, encryption down to the level of individual fields, and auditing  
  ● Global clusters to control data residence |
| **Design and implement** | **Design and implement** |
| ● Governance processes to build and maintain the single view | ● 10-Step methodology to creating a single view |

**Table 1: Required capabilities for single view use cases**

Beyond technology, organizations need to implement the business processes needed to deliver and maintain a single view. Built on experiences gained over many years working with organizations of all sizes and industries, we have developed a 10-Step Single View Methodology to share best practices.
The MongoDB 10-step methodology provides a step-by-step guide to each stage of the single view project – from discovery to data loading and reconciliation, through to deployment, along with the governance and tools essential to successfully delivering a single view app with MongoDB.

**Customer Data Management and Personalization**

The single view use case is an example of customer data management. In this context, customer data also applies to data held on employees, citizens, patients, students, partners, through to complete corporate entities. This data is used by both backend Line of Business apps such as CRM, HR, billing, and order processing, as well as front-end apps such as eCommerce user profiles and customer self-service portals.

The same set of required capabilities defined earlier for single view applies also to customer data management, especially in personalizing a user’s experience as they interact with your apps and digital properties. To stand out from competitors, you need to offer contextual and engaging experiences that are personalized for each user – reflecting their interests and providing recommendations to them, in real time and on any device.

**Customer example:** Expedia is a virtual concierge – it knows if you want family-friendly holidays, business travel, or a once-in-a-lifetime break. Using MongoDB, it pushes special travel offers to users in real time by analyzing their searches and comparisons across its site.

Managing any type of personally identifiable information makes privacy a key requirement for your customer data management apps. The European Union’s GDPR (General Data Protection Regulation) is ushering in a new wave of global privacy regulations that govern how organizations collect, store, process, retain, and share the personal data of citizens. You can learn more about how MongoDB can support your compliance initiatives by downloading our GDPR and data privacy guide.
Internet of Things (IoT) and Time-Series Data

Today, the IoT is enabling companies to blend the physical with the digital worlds. The business value of connecting all of these “things” is realized through the creation of new revenue models, improved productivity, and the ability to generate new insights that drive operational efficiencies. The IoT already connects billions of devices worldwide, and that number is growing every day. Many market analysts predict that only by adopting IoT can organizations fully unlock the revenue opportunities promised by digital transformation.

IoT is one example of an application with time-series data at its core. Other time-series use cases include financial trading systems, clickstreams, monitoring and event logging, and asset tracking. Download our white paper to learn more about time-series schema design best practices.

<table>
<thead>
<tr>
<th>Required Capabilities</th>
<th>Why MongoDB?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data model</strong></td>
<td>Data model</td>
</tr>
</tbody>
</table>
| - Accommodate complex and quickly changing time-series data generated by sources such as heterogeneous sensors, connected devices, log files, and financial stock tickers | - Document data model to store rich, multi-structured data  
- Flexible schema makes it easy to ingest new data without changes to the database |
| **Scale & Resilience**| Scale & Resilience  |
| - Ingest high volumes of sensor and event data from geographically distributed assets  
- Always-on | - Distributed systems architecture with native sharding  
- Replica sets and Global Clusters to distribute data across multiple regions |
| **Analytics**         | Analytics  |
| - Concurrent real-time analysis against live, operational data, without the latency of moving it to specialized storage  
- Discover valuable insights with analytics and machine learning. | - Workload isolation to separate data ingest from analytics processes running on the same database cluster  
- Expressive query language, secondary indexing, aggregation pipeline, and on-demand materialized views  
- Visualization: MongoDB Charts and BI Connector  
- AI: Spark Connector, idiomatic R and Python drivers  
- MongoDB Data Lake for offline analytics of your data in S3, using the same query language and tools as your OLTP database |

**End to End Platform**
Seamlessly manage data at the edge of the network and in the core backend
Push anomalous events to consuming systems

- MongoDB Realm to synchronize local data at the edge with backend MongoDB Atlas database
- MongoDB Connector for Apache Kafka, change streams and Atlas triggers for reactive event driven pipelines

<table>
<thead>
<tr>
<th>Table 2: Required capabilities for IoT and time-series use cases</th>
</tr>
</thead>
</table>

IoT customer example: Bosch has built its Internet of Things suite on MongoDB, bringing the power of big data to a new range of industrial and consumer IoT applications including manufacturing, automotive, retail, energy and many others. Learn more from the case study, and review our IoT Reference Architecture.

Time-series customer example: Man AHL’s Arctic application uses MongoDB to store high frequency financial services market data, handling 250M ticks per second. The hedge fund manager’s quantitative researchers (“quants”) use MongoDB to research, construct, and deploy new trading models in order to understand how markets behave. With MongoDB, Man AHL realized a 40x cost saving when compared to an existing proprietary database. In addition to cost savings, they were able to increase processing performance by 25x over the previous solution.

Product Catalogs and Content Management

With online and mobile sales volumes growing north of 25% year-on-year, it is vital for companies to scale their eCommerce product catalogs to meet explosive demand and a tougher competitive market. At the same time, the increasing ubiquity of high speed internet connectivity and smart mobile devices is changing the content management landscape. Long gone are the days of static websites and text-based publications. They are being replaced by engaging and immersive experiences enlivened with rich media assets and user generated content, all of which have to be delivered with low latency to any device, anywhere on the planet.

While most catalog use cases are centered on ecommerce product catalogs, other examples include asset catalogs used for internal inventory management and trade catalogs used in financial services.

Content management use cases include websites, online publications, research and training materials, document management, and open data repositories. Many content management systems are custom-built, but there are also packaged platforms such as Adobe Experience Manager and Sitecore, both of which use MongoDB for data management.
## Required Capabilities

<table>
<thead>
<tr>
<th>Data model</th>
<th>Why MongoDB?</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Handle massive variability in content and catalog attributes, metadata, media assets, and UGC</td>
<td>● Document data model to store rich, multi-structured data</td>
</tr>
<tr>
<td>● Quickly update schema as new products and content are offered</td>
<td>● Flexible schema with no schema migrations</td>
</tr>
<tr>
<td>● GridFS to store binary assets in the database</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User experience</th>
<th>User experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Rich queries and search as users browse the product catalog and CMS</td>
<td>● Expressive MongoDB Query Language, Atlas Search based on Lucene, graph traversals</td>
</tr>
<tr>
<td>● Never slow down under the peak loads generated by promotions, cyclical peaks, or new publications</td>
<td>● Native sharding for horizontal and elastic scale in response to demand</td>
</tr>
<tr>
<td>● Never go down, always available</td>
<td>● Global Clusters to colocate data close to users, and replication for resilience</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Real time analytics</th>
<th>Real time analytics</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Serve up personalized recommendations</td>
<td>● Aggregation pipeline</td>
</tr>
<tr>
<td>● Monitor sales performance and content consumption in real time</td>
<td>● AI: Spark Connector, idiomatic R and Python drivers</td>
</tr>
<tr>
<td></td>
<td>● Dashboards: MongoDB Charts and BI Connector</td>
</tr>
</tbody>
</table>

**Table 3:** *Required capabilities for product catalogs and content management*

**Product catalog customer example:** As a top 10 global retail brand with 170+ million active buyers and 1 billion live listings across 190 markets around the world, eBay cannot afford systems downtime. This is why the company relies on MongoDB as one of its core enterprise data platform standards, powering multiple, customer-facing applications that run ebay.com. The company’s product catalog is distributed and scaled on a 50-node MongoDB replica set, spread across multiple data centers.

**Content management customer example:** Around $4 trillion is invested globally every year in medical and scientific research. Elsevier publishes 17% of the content and discoveries generated from that research. MongoDB is at the core of the Elsevier cloud-based platform, enabling the company to apply software and analytics that turns content into actionable knowledge and new insights for its customers.

## Payment Processing

With payment processing moving to web and mobile channels, organizations are seeking to modernize existing backend transactional systems to deliver the availability and scale needed to reliably serve more customers. They have to do this without giving up the strong data integrity guarantees they have come to expect from traditional relational databases.
### Required Capabilities

<table>
<thead>
<tr>
<th>Required Capabilities</th>
<th>Why MongoDB?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data integrity</strong></td>
<td></td>
</tr>
<tr>
<td>● Handle payment and order processing with data correctness guarantees</td>
<td>● Multi-document ACID transactions with snapshot isolation</td>
</tr>
<tr>
<td><strong>Data model</strong></td>
<td></td>
</tr>
<tr>
<td>● Support variability in payment data and financial instruments</td>
<td>● Flexible document data model to store rich, multi-structured data</td>
</tr>
<tr>
<td>● High precision number, temporal, and geospatial data types for lossless processing, sorting and comparisons</td>
<td>● Bank to Bank interchange standards (SWIFT, Maestro) now define JSON implementations</td>
</tr>
<tr>
<td><strong>Analytics</strong></td>
<td></td>
</tr>
<tr>
<td>● Fraud detection, risk profile and liquidity monitoring</td>
<td>● Aggregation pipeline with Workload Isolation, enabling translytics on the payment platform</td>
</tr>
<tr>
<td>● Monitor sales performance in real time</td>
<td>● AI: Spark Connector, R &amp; Python drivers</td>
</tr>
<tr>
<td><strong>Resilience &amp; Latency</strong></td>
<td>● MongoDB replica sets and Global Clusters to distribute data across multiple regions</td>
</tr>
<tr>
<td>● Maintain availability in the face of outages and planned maintenance</td>
<td></td>
</tr>
<tr>
<td><strong>Data Integration</strong></td>
<td></td>
</tr>
<tr>
<td>● Integrate eCommerce apps with payment providers</td>
<td>● MongoDB Atlas triggers and functions to call 3rd party payment provides</td>
</tr>
<tr>
<td>● Push payment and fraud alerts to consuming systems</td>
<td>● MongoDB Connector for Apache Kafka, change streams and Atlas triggers for event-driven pipelines</td>
</tr>
</tbody>
</table>

**Table 4: Required capabilities for payment processing**

Payment processing use cases include eCommerce backends, financial trading systems, billing engines, and mobile payment gateways.

**Payment processing customer example:** Cisco migrated its eCommerce platform, handling over $40bn of revenue per annum, from a legacy relational database to MongoDB.
As a result, customer experience is improved by reducing latency 5x and improving availability by 100x. The migration has driven developer productivity gains, eliminating 25+ business critical backlog features. Developers can build new applications faster, while the company’s eCommerce platform can tap into the business agility enabled by cloud computing.

Unlike most modern, distributed databases, MongoDB has support for multi-document transactions. Through snapshot isolation, transactions provide a consistent view of data, and enforce all-or-nothing execution to maintain data integrity. Transactions in MongoDB feel just like transactions developers are familiar with from relational databases, making them simple to add to any application that needs them. Unlike relational databases, MongoDB’s transactions can operate against highly scaled-out sharded clusters.

The addition of multi-document transactions makes it easier than ever for developers to address a complete range of use-cases with MongoDB, while for many, simply knowing that they are available provides critical peace of mind. You can learn more by reading our Multi-Document ACID Transactions whitepaper.

Mobile Apps

From a “nice to have” ten years ago, mobile is now central to almost every customer engagement strategy. From smarter devices to ubiquitous sensors embedded in every-day objects, through to high performance WiFi and cellular networks, organizations embracing “mobile-first” development are opening up new opportunities.

Many organizations start their mobile journeys with mobile-optimized websites and refactored web apps such as customer portals and banking – enabling them to deliver services to customers across their preferred channels. Now new waves of mobile apps have emerged, including immersize, AI-driven lifestyle and retail experiences, mobile payments, Augmented Reality and gaming, personalized healthcare and fitness tracking, streaming services, and many more.

The MongoDB data platform underpins your mobile apps, from the device through to the backend:

- **MongoDB Realm database** – Radically simplifies development and improves user experience. Store data where you need it, on your iOS and Android devices to your backend in the cloud – all using a rich data model.
- **MongoDB Realm** and Stitch – The best way for your applications and users to access the data and services they need. Realm sits at the front end and middle layer of MongoDB’s data platform, providing secure access to data hosted in MongoDB Atlas. With Realm and Stitch, you can quickly create rich, secure apps and services without app servers, web hosts, or gateways, and keep documents in sync between your mobile devices and Atlas.
- **MongoDB Atlas** – on-demand, elastic, and fully managed global cloud database backend, with baked-in best practices, leaving developers free to concentrate on their apps.
### Required Capabilities

<table>
<thead>
<tr>
<th>Required Capabilities</th>
<th>Why MongoDB?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Developer velocity</strong></td>
<td><strong>Developer velocity</strong></td>
</tr>
<tr>
<td>- Rapidly build and evolve mobile apps</td>
<td>- Same data model on mobile devices and backend</td>
</tr>
<tr>
<td>- Create rich app experiences</td>
<td>- Object and document data structures match objects and JSON syntax used in code</td>
</tr>
<tr>
<td>- Feature toggles to incrementally deploy new app functionality</td>
<td>- Simple SDKs to access data stored on-device and in the backend</td>
</tr>
<tr>
<td><strong>Data portability and security</strong></td>
<td><strong>Data portability and security</strong></td>
</tr>
<tr>
<td>- Sync data between device and backend</td>
<td>- Expressive query language to work on data on the device and in the backend</td>
</tr>
<tr>
<td>- Access controls to data</td>
<td>- Flexible data model to support multiple app feature sets</td>
</tr>
<tr>
<td>- Same mobile database for Android, iOS and IoT</td>
<td><strong>Resilience and scale</strong></td>
</tr>
<tr>
<td>- Never go down, always available</td>
<td><strong>Resilience and scale</strong></td>
</tr>
<tr>
<td>- Never slow down under the peak loads generated by new app launches</td>
<td>- Global Clusters to colocate data close to users and replication for resilience</td>
</tr>
<tr>
<td><strong>Resilience and scale</strong></td>
<td>- Realm serverless platform: auto-scale</td>
</tr>
<tr>
<td><strong>Table 5: Required capabilities for mobile apps</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Mobile customer example:** 7-Eleven used MongoDB to expand customer engagement into m-commerce. Customers can browse a local store’s inventory, and then click and collect product they want to buy, all via their mobile app. As well as handling the transactional needs of the app, MongoDB also layers the analytics capabilities that allow 7-Eleven to further improve customer experience.

**Mainframe Offload**

Despite its long predicted demise, the mainframe remains a critical IT asset in many large enterprises. But ongoing reliance on the mainframe does not come without challenges. Web, mobile, social, Artificial Intelligence, and Internet of Things applications are driving a deluge
of new data. The volume, speed, and diversity of this data is overwhelming mainframe environments. Coupled with pressures to meet new regulatory demands and cut costs, CIOs are challenged in how quickly they can remake the business for digital, while trying to innovate on top of legacy technologies.

Mainframe offloading is the process of replicating commonly accessed mainframe data to an Operational Data Layer (ODL) built on MongoDB, against which operations are redirected from consuming applications. The existing mainframe is left untouched. By offloading mainframe operations to MongoDB, organizations can drive faster innovation, improved customer experience, and reduced costs.

<table>
<thead>
<tr>
<th>Required Capabilities</th>
<th>Why MongoDB?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data model</strong></td>
<td></td>
</tr>
<tr>
<td>● Ingest system of record data from the mainframe, then enrich it with data of any structure from source systems</td>
<td></td>
</tr>
<tr>
<td>● Dynamically adapt as a source system’s schema changes</td>
<td></td>
</tr>
<tr>
<td>● Document data model to store rich, multi-structured data</td>
<td></td>
</tr>
<tr>
<td>● Flexible schema with no schema migrations</td>
<td></td>
</tr>
<tr>
<td><strong>Query model</strong></td>
<td></td>
</tr>
<tr>
<td>● Multiple access patterns and query types over multiple channels</td>
<td></td>
</tr>
<tr>
<td>● Simple lookups through to sophisticated analytics for business insight</td>
<td></td>
</tr>
<tr>
<td>● Expressive query language, secondary indexing, aggregation pipeline, and on-demand materialized views</td>
<td></td>
</tr>
<tr>
<td>● MongoDB Realm provides simple and secure API access from clients and microservices to your data</td>
<td></td>
</tr>
<tr>
<td>● Data visualization: MongoDB Charts and BI Connector</td>
<td></td>
</tr>
<tr>
<td>● AI: Spark Connector, idiomatic R and Python drivers</td>
<td></td>
</tr>
<tr>
<td><strong>Synchronize</strong></td>
<td></td>
</tr>
<tr>
<td>● Ensure data is available and consistent across mainframe and operational apps</td>
<td></td>
</tr>
<tr>
<td>● Any data changes can be automatically propagated between the mainframe and ODL as they happen, using the MongoDB Connector for Apache Kafka</td>
<td></td>
</tr>
<tr>
<td><strong>Grow and protect</strong></td>
<td></td>
</tr>
<tr>
<td>● Scale as new data sources are on-boarded</td>
<td></td>
</tr>
<tr>
<td>● Robust security and data sovereignty controls</td>
<td></td>
</tr>
<tr>
<td>● Resilient, always-on</td>
<td></td>
</tr>
<tr>
<td>● Distributed systems architecture with native sharding for scale out</td>
<td></td>
</tr>
<tr>
<td>● Comprehensive access controls, encryption down to the level of individual fields, and auditing</td>
<td></td>
</tr>
<tr>
<td>● Global clusters to control data residence</td>
<td></td>
</tr>
</tbody>
</table>

**Table 6:** Required capabilities for mainframe offload use cases
**Customer example:** Alight Solutions (formerly AON Hewitt) offloaded its Human Capital Services from the mainframe to MongoDB, improving application performance by 250x, reducing costs, and creating a platform for innovation. Review the Mainframe Offload Reference Architecture to learn more about why to offload, common approaches and patterns, enabling technologies, program execution, and governance.

**Online Analytics and AI**

Every organization is striving to be data-driven. But it isn’t only the data itself that is valuable – it is the insight it generates. That insight can help designers better predict new products that customers will love. It can help manufacturing companies model failures in critical components, before costly recalls. It can help financial institutions detect fraud and retailers better forecast supply and demand, while eCommerce businesses can build recommendation engines that make more informed and relevant suggestions to customers. The list goes on.

How quickly an organization can unlock and act on that insight is becoming a major source of competitive advantage. Collecting data in operational systems and then relying on nightly batch ETL (Extract Transform Load) processes to update the Enterprise Data Warehouse is no longer sufficient. Speed-to-insight is critical, and so analytics against live operational data to drive real-time action is fast becoming a necessity.

**Figure 4:** MongoDB’s distributed systems architecture enables the isolation of operational and real-time analytical workloads on a single cluster, providing faster and fresher insights against your data
<table>
<thead>
<tr>
<th>Required Capabilities</th>
<th>Why MongoDB?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data model</strong></td>
<td><strong>Data model</strong></td>
</tr>
<tr>
<td>● Handle data of any structure from operational systems</td>
<td>● Document data model to store rich, multi-structured data</td>
</tr>
<tr>
<td>● Dynamically adapt the schema during machine model training</td>
<td>● Flexible schema with no schema migrations</td>
</tr>
<tr>
<td><strong>Online analytics</strong></td>
<td><strong>Online analytics toolkit</strong></td>
</tr>
<tr>
<td>● Multiple access patterns and query types</td>
<td>● Expressive query language, secondary indexing, Lucene-based Atlas Search, aggregation pipeline, and on-demand materialized views</td>
</tr>
<tr>
<td>● Simple lookups through to sophisticated data transformations and aggregations</td>
<td>● Distributed architecture to enable workload isolation: separating data ingest from analytics processes running on the same database cluster</td>
</tr>
<tr>
<td>● Eliminate fragile and lengthy ETL</td>
<td><strong>Integrated with data analytics ecosystem</strong></td>
</tr>
<tr>
<td></td>
<td>● Data visualization: MongoDB Charts and BI Connector</td>
</tr>
<tr>
<td></td>
<td>● AI: Spark Connector, idiomatic R and Python drivers</td>
</tr>
<tr>
<td></td>
<td>● MongoDB Atlas triggers and functions to call cloud-based machine learning frameworks</td>
</tr>
<tr>
<td></td>
<td>● MongoDB Connector for Apache Kafka, change streams and Atlas triggers for reactive event driven pipelines</td>
</tr>
<tr>
<td></td>
<td>● MongoDB Data Lake for offline analytics of your data in S3, using the same query language and tools as your OLTP database</td>
</tr>
<tr>
<td><strong>Integrated with data analytics ecosystem</strong></td>
<td><strong>Grow and protect</strong></td>
</tr>
<tr>
<td>● Visualize data of any structure</td>
<td>● Distributed systems architecture with native sharding for scale out</td>
</tr>
<tr>
<td>● Leverage investments in existing reporting and AI frameworks</td>
<td>● Comprehensive access controls, encryption, and auditing</td>
</tr>
<tr>
<td>● Push analytics updates to consuming systems</td>
<td>● Global clusters to control data residence</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Required capabilities for real-time analytics and AI use cases

You can get started with MongoDB Charts, our native visualization tool for creating insights from multi-structured document data, by trying it out on MongoDB Atlas. Download our Apache Spark and MongoDB white paper to learn more about turning analytics into real-time actions.
Storage Tiering for Online Analytics

Analytical systems can grow to substantial sizes, often storing hundreds of Terabytes of data. Historically these systems were served by an offline Enterprise Data Warehouse where data was processed in nightly or weekly batches. However changes in customer expectations and in business requirements demand online environments that can serve subsets of the most important data in real time.

A good example for this usage pattern is the Telecom Call Data Record (CDR) store. Traditionally the CDR was used offline by internal batch business processes to generate monthly billing information and customer churn reports. Today the expectation is that customers have direct and real time access to the CDR to provide visibility into their usage patterns, calls made and data consumption, while the business wants access to the freshest data to present relevant offers, detect fraud, and analyze usage patterns.

Assuming several years of data, CDRs can quickly reach hundreds of TBs in data size. In this scenario, tiering data can provide efficient and cost-effective data access for both the telecom provider and their customers:

- The most recent 45-days worth of CDRs represent “critical hot data”, stored in memory and on SSDs to provide low-latency, real time access.
- The next three months worth of data is considered “warm” and is stored on disk. While access will not be as fast as the hot data tier, it is still delivered at an acceptable latency for user experience.
- Older data is considered cold, and can be archived to the MongoDB Data Lake running on cloud object storage where it is stored for the required seven year regulatory retention period.

Other examples for the tiered data pattern can be found in payment archives or payment Instruction management systems which store information on the planned and executed scheduled transactions in a bank. While the last 45 days are of immediate interest to the client base, older data can be kept available in the database and in the data lake.

The key formula for system sizing in a tiered data architecture can be represented as:

\[
\text{Total Data Amount} \times \left( \frac{\text{Hot time}}{\text{total time}} \right) \times \text{compression ratio} = \text{CRITICAL DATA VOLUME for hot data}
\]

Based on client design preference, database nodes can be created with 32GB - 64GB memory and the appropriate CPU resource. A regular 64GB node with 16 cores will hold around 5TB of critical hot data.
To illustrate system sizing, we will use an example where this approach was applied at a large European payments provider who was seeking to migrate away from their existing legacy relational data warehouse.

The customer stores 150TB of payment data for each 6-month period. Their tiering model is structured so that:

1. The most recent 30-days represents hot data that has to be queried with the lowest possible latency.
2. The next 30-days represents warm data. Latency is important, but not as critical.
3. The remaining 120-days is cold data.

Testing showed they could compress the data by 50%, so applying the formula we documented earlier, both their hot and warm data was sized at 12.5TB each:

\[
150TB \times \left( \frac{30}{180 \text{ days}} \right) \times 50\% \text{ compression} = 12.5TB \text{ hot data.}
\]

With each node handling 5TB of data, they were able to provision a 3-shard cluster for their hot data, and another 3-shard cluster for their warm data. Using MongoDB’s Time to Live (TTL) indexes, they are able to offload all cold data into the Atlas Data Lake.

**Offline (Long Running) Analytics**

While still an incredibly powerful part of many organizations analytics and BI infrastructure, traditional data warehousing, ETL, and analytics tools have for some time been unable to keep up with the data avalanche generated by the digital economy. The Hadoop framework – comprising dozens of independently developed and maintained projects – emerged about a decade ago as the panacea to address the data volumes organizations were grappling with. But Hadoop quickly became more of a problem than a solution – requiring massive cost and complexity to get value from data. While it was easy to dump massive volumes of data into Hadoop, it quickly became clear that getting anything useful out of it was beyond what most companies were capable of delivering.

**Enter MongoDB Atlas Data Lake**

MongoDB Atlas Data Lake allows you to quickly and easily query data in any format on Amazon S3 using the MongoDB Query Language (MQL) and tools.

With Atlas Data Lake you can realize the value of your data lake faster: 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.
Unlocking the Data Lake

Here is how the MongoDB Atlas Data Lake helps you unlock your data lake:

- **No schema definition required.** Any richly structured data stored in JSON, BSON, CSV, TSV, Avro, and Parquet formats can be analyzed in place without incurring the complexity, cost, and time-sink of ingestion, transformation, and metadata management.

- **Work with data at any scale, anywhere.** Decoupled compute and storage allows you to seamlessly expand and contract each tier of your data lake independently. Parallelize complex queries to take advantage of existing data partitioning across multiple S3 buckets, delivering performance – even for global analytics that access data in multiple AWS regions.

- **Accelerate productivity of your developer and data teams.** Spin up your data lake right alongside your Atlas OLTP clusters from a common UI. Then query and visualize your data using the same rich and expressive MongoDB Query Language.
and tools. You can query data any way you want – from simple lookups and ad-hoc queries through to sophisticated aggregation and data processing pipelines.

- **On-demand and serverless.** Simply provide access to your existing S3 buckets with a few clicks from the Atlas UI and start running queries immediately. Atlas Data Lake is completely serverless so there is no infrastructure to set up and manage, and you pay only for the queries run and only when actively working with the data.

**MongoDB Atlas Data Lake Use Cases**

All users across your organization can now benefit from the data in your S3 data lake, working together to answer questions and build data-rich applications. Common use cases for MongoDB Atlas Data Lake include:

- **Data Lake Analytics.** Explore all of your rich data naturally, democratizing access across business units and users.
- **Data Products and Services.** Create data snapshots, market research, and insight-as-a-service products.
- **Machine Learning and AI.** Accelerate data exploration and feature engineering, exposing models to build more intelligence into your apps and services.
- **Active Archives.** Run historical analysis against data assets retained in long-term cold storage.

You can learn more from the [MongoDB Atlas Data Lake product page](#).

**When Should I Not Use MongoDB?**

As illustrated above, MongoDB serves most operational / OLTP and analytics use cases. There are, however, some workloads that are better suited to alternative technologies, which we discuss below.

**Common Off-the-Shelf Software Built for Relational Databases**

While many organizations have successfully migrated from relational databases to MongoDB, you cannot drop-in MongoDB as a replacement for ISV packaged applications built around the relational data model and SQL. In these scenarios, it is better to work directly with the ISV to encourage them to offer MongoDB as the data persistence layer with their application.

To learn how MongoDB stacks up against the most popular relational databases, check out our resources pages:

- [MongoDB and Oracle compared](#).
- [MongoDB and Postgres compared](#).
- [MongoDB and MySQL compared](#).
Ad-Hoc Reporting in the MongoDB Server: Unindexed Queries

When query patterns are predictable and can be optimized with secondary indexes, then the MongoDB Server will efficiently power online analytics-heavy workloads. Users can easily visualize results of analytics queries with MongoDB Charts or with the BI Connector, and feed the data into AI frameworks with the MongoDB Spark Connector and R or Python drivers.

The MongoDB Server does not optimize for ad-hoc reporting use cases where analysts are constantly executing full table scans across unindexed data sets. In these scenarios, you can use the MongoDB Atlas Data Lake, or pair MongoDB with your existing data warehouse.

Data Warehouse Replacements with Atlas Data Lake

While the MongoDB Atlas Data Lake is an alternative to Hadoop, it is not a replacement of a data warehouse. Rather, it is a powerful complement to that technology. Data warehouses offer comprehensive controls for ETL, data lineage, metadata management and cataloging, governance, security, and workload management, backed by an extensive ecosystem of skills and tools.

The following table identifies workload characteristics to demonstrate the different properties of traditional data warehouses and the Atlas Data Lake.

<table>
<thead>
<tr>
<th></th>
<th>Data Warehouse</th>
<th>MongoDB Atlas Data Lake</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data set</strong></td>
<td>Highly curated, cleansed, filtered, aggregated data, ingested via ETL processes from operational databases and applications</td>
<td>Vast pools of raw data, stored in its native form, ingested from logs, sensors, devices, streams, APIs and operational databases</td>
</tr>
<tr>
<td><strong>Data structures</strong></td>
<td>Pre-defined and fixed, tabular schema, with well defined constraints and relationships</td>
<td>Dynamic and flexible schema, rich data structures of any shape</td>
</tr>
<tr>
<td><strong>Query patterns</strong></td>
<td>Highly optimized for specific reporting and BI purposes including dashboards, statistics and predictive modeling, regressions, and decision trees</td>
<td>Ad-hoc, data exploration and discovery, machine learning workloads, dashboards</td>
</tr>
<tr>
<td><strong>Data storage</strong></td>
<td>Specialized hardware, dedicated clusters of optimized compute and storage</td>
<td>General purpose cloud storage accessed on-demand by serverless compute instances</td>
</tr>
<tr>
<td><strong>Consumers</strong></td>
<td>Business analysts, data scientists</td>
<td>Business analysts, developers, data scientists, data engineers</td>
</tr>
</tbody>
</table>
The Atlas Data Lake is an ideal option for those organizations exploring modern, cloud-native approaches to harnessing value from new sources of data flowing into their business.

**Conclusion**

MongoDB is the most popular and widely used modern data platform in the market. Developers working with data touch more than the database, and need the best way to work with data across all their systems. Through its natural and flexible data model, expressive query language, distributed design, and platform agnostic architecture, MongoDB is able to serve the most demanding operational, real-time analytics and data lake use cases your teams are building today.

MongoDB supports its platform-based approach through an array of enabling technologies, training, and professional services. Get in touch to learn more about MongoDB for modern apps, review use cases, and more

**We Can Help**

We are the company that builds and runs MongoDB. Over 14,200 organizations rely on our commercial products. We offer software and services to make your life easier:

- **MongoDB Atlas** is the database as a service for MongoDB, available on AWS, Azure, and GCP. It lets you focus on apps instead of ops. With MongoDB Atlas, you only pay for what you use with a convenient hourly billing model. Atlas auto-scales in response to application demand with no downtime, offering full security, resilience, and high performance.
- **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 visualizations of MongoDB data anywhere. 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.
- The **MongoDB Realm and Stitch Platform** helps you build better fullstack apps faster. It offers easily configurable rules for accessing data and services directly from your application frontend, along with serverless functions to execute application logic. You can automatically sync data between the client and backend data layer. Through
integrations with your code repositories you can develop locally and then seamlessly deploy version-controlled application updates to test and production.

- **MongoDB Realm Database** has been installed over 2 billion times, offering a fast, easy-to-use, alternative to SQLite and Core Data. With support for complex queries, safe threading, a reactive architecture to create responsive and fluent UIs, encryption, and cross-platform support, developers can simplify their code and build powerful and engaging experiences on more devices.

- **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
- Presentations
- White Papers
- Free Online Training
- Webinars and Events
- Documentation