Use Case Guidance:

Selecting the Best Apps for MongoDB, and when to evaluate other options

March 2019

Table of Contents

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?

Unlike NoSQL databases that serve only a niche set of use cases, the MongoDB server is a general purpose database designed to serve any OLTP / operational and real-time 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, MongoDB provides complete infrastructure agility.

Through its design, MongoDB provides a technology foundation that enables development teams to meet the demands of modern apps with:

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.

Through this design, developers are more productive, building apps faster and running them securely at scale. To learn more about the three core pillars of MongoDB's architecture and how they enable you to serve a broad set of use cases, review the MongoDB Architecture Guide.

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 opportunities available 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.

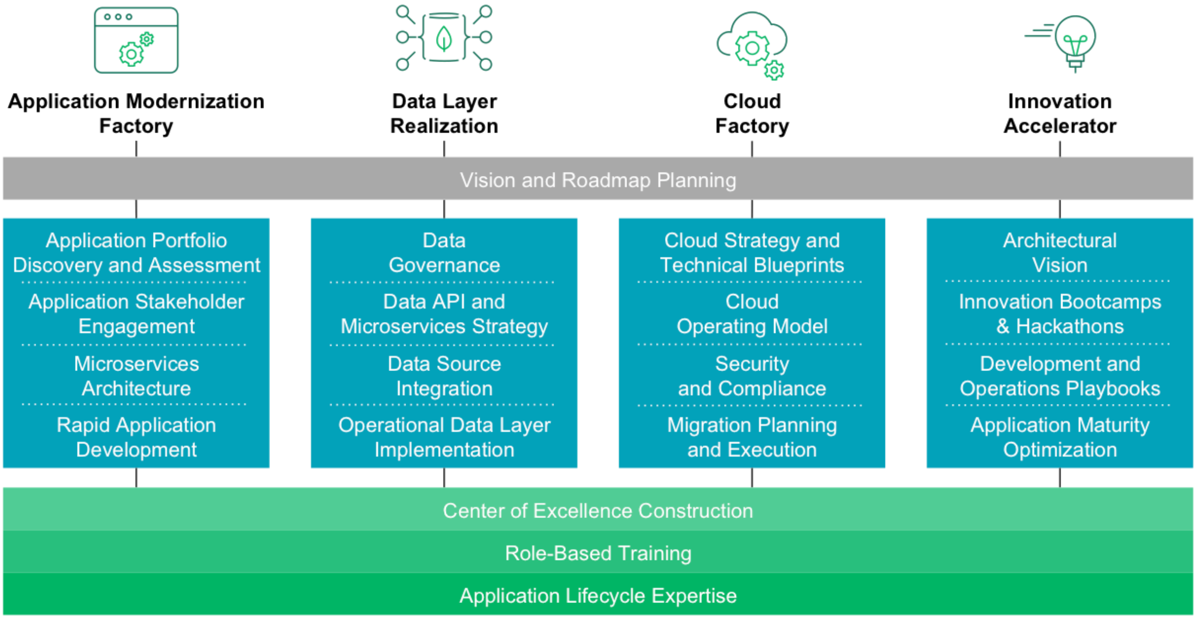
Application Modernization Factory
Data Layer Realization
Cloud Factory
Innovation Accelerator
Vision and Roadmap Planning
Application Portfolio Discovery and Assessment
Application Stakeholder Engagement
Microservices Architecture
Rapid Application Development
Data Governance
Data API and Microservices Strategy
Data Source Integration
Operational Data Layer Implementation
Cloud Strategy and Technical Blueprints
Cloud Operating Model
Security and Compliance
Migration Planning and Execution
Architectural Vision
Innovation Bootcamps & Hackathons
Development and Operations Playbooks
Application Maturity Optimization
Center of Excellence Construction
Role-Based Training
Application Lifecycle Expertise

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 a 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 Stitch 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.

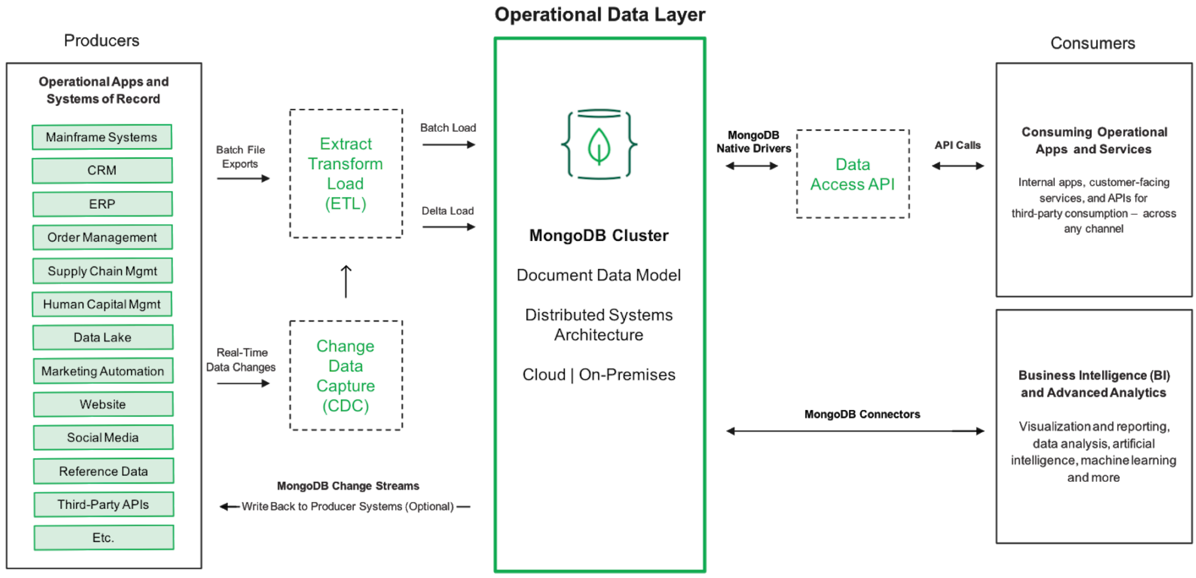
Operational Data Layer
Producers
Operational Apps and Systems of Record
Mainframe Systems
CRM
ERP
Order Management
Supply Chain Mgmt
Human Capital Mgmt
Data Lake
Marketing Automation
Website
Social Media
Reference Data
Third-Party APIs
Etc.
Batch File Exports
Extract Transform Load (ETL)
Batch Load
Delta Load
Real-Time Data Changes
Change Data Capture (CDC)
MongoDB Change Streams
Write Back to Producer Systems (Optional)
MongoDB Cluster
Document Data Model
Distributed Systems Architecture
Cloud | On-Premises
MongoDB Native Drivers
Data Access API
API Calls
MongoDB Connectors
Consumers
Consuming Operational Apps and Services
Internal apps, customer-facing services, and APIs for third-party consumption – across any channel
Business Intelligence (BI) and Advanced Analytics
Visualization and reporting, data analysis, artificial intelligence, machine learning and more

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.

Required Capabilities	Why MongoDB?
Data model ● Ingest data of any structure from source systems ● Dynamically adapt as a source systems schema changes	**Data model** ● Document data model to store rich, multi-structured data ● Flexible schema with no schema migrations
Query model ● Multiple access patterns and query types ● Simple lookups through to sophisticated analytics for business insight and personalization	**Query model** ● Expressive query language, secondary indexing, and aggregation pipeline ● Data visualization: MongoDB Charts and BI Connector ● AI: Spark Connector, idiomatic R and Python drivers
Grow and protect ● Scale as new data sources are on-boarded ● Robust security and data sovereignty controls	**Grow and protect** ● Distributed systems architecture with auto-sharding for scale out ● Comprehensive access controls, encryption, and auditing ● Global clusters to control data residence
Design and implement ● Governance processes to build and maintain the single view	**Design and implement** ● 10-Step methodology to creating a single view

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

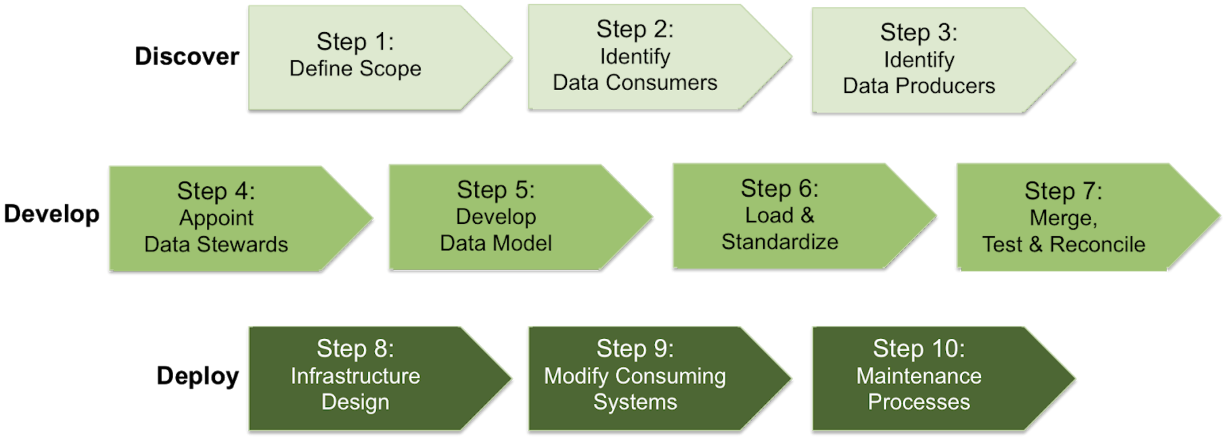
Discover
Step 1: Define Scope
Step 2: Identify Data Consumers
Step 3: Identify Data Producers
Develop
Step 4: Appoint Data Stewards
Step 5: Develop Data Model
Step 6: Load & Standardize
Step 7: Merge, Test & Reconcile
Deploy
Step 8: Infrastructure Design
Step 9: Modify Consuming Systems
Step 10: Maintenance Processes

Figure 3: *10-Step methodology to building a single view with MongoDB*

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. It provides a step-by-step guide to each stage of the 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.

Required Capabilities	Why MongoDB?
Data model ● Accommodate complex and quickly changing time-series data generated by sources such as heterogeneous sensors, connected devices, log files, and financial stock tickers	**Data model** ● 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 ● Ingest high volumes of sensor and event data from geographically distributed assets ● Always-on	**Scale & Resilience** ● Distributed systems architecture with auto-sharding ● Replica sets and Global Clusters to distribute data across multiple regions
Real time 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.	**Real time analytics** ● Workload isolation to separate data ingest from analytics processes running on the same database cluster ● Expressive MongoDB Query Language, secondary indexing, and aggregation pipeline ● Visualization: MongoDB Charts and BI Connector ● AI: Spark Connector, idiomatic R and Python drivers
End to End Platform ● Seamlessly manage data at the edge of the network and in the core backend ● Push anomalous events to consuming systems	**End to End Platform** ● MongoDB Mobile embedded into edge gateways for local storage and processing ● MongoDB Stitch to synchronize local data with backend MongoDB Atlas database ● MongoDB change streams and Atlas triggers for reactive event driven pipelines

Table 2: *Required capabilities for IoT and time-series use cases*

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.

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.

Real-Time 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.

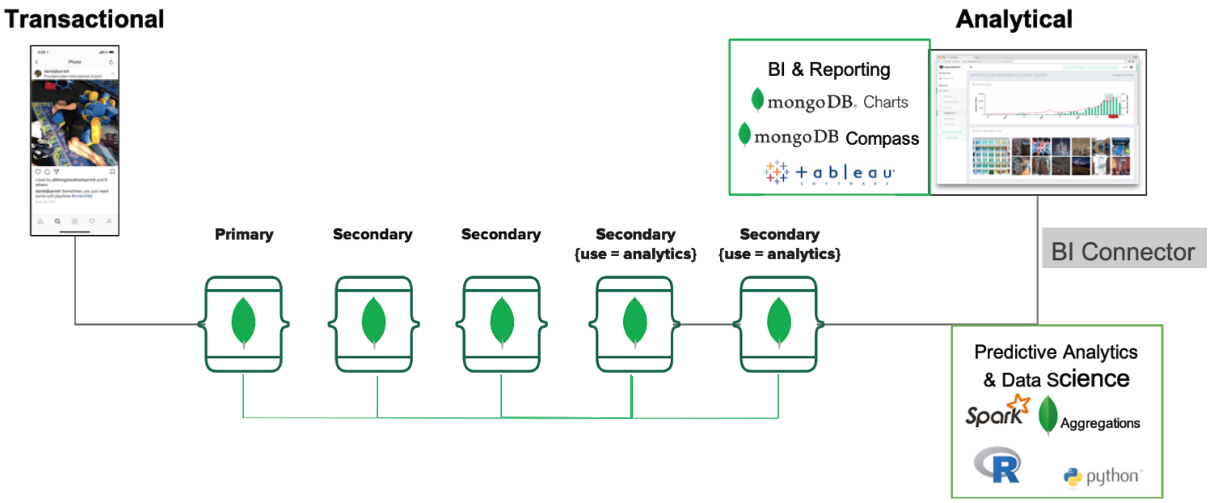

Figure 4: *MongoDB's distributed systems architecture enables the isolation of operational and analytical workloads on a single cluster, providing faster and fresher insights against your data*

Required Capabilities	Why MongoDB?
Data model ● Handle data of any structure from operational systems ● Dynamically adapt the schema during machine model training	**Data model** ● Document data model to store rich, multi-structured data ● Flexible schema with no schema migrations
Analytics toolkit ● Multiple access patterns and query types ● Simple lookups through to sophisticated data transformations and aggregations ● Eliminate fragile and lengthy ETL	**Analytics toolkit** ● Expressive query language, secondary indexing, and aggregation pipeline ● Distributed architecture to enable workload isolation: separating data ingest from analytics processes running on the same database cluster
Integrated with data analytics ecosystem ● Visualize data of any structure ● Leverage investments in existing reporting and AI frameworks ● Push analytics updates to consuming systems	**Integrated with data analytics ecosystem** ● Data visualization: MongoDB Charts and BI Connector ● AI: Spark Connector, idiomatic R and Python drivers ● MongoDB Stitch functions to call cloud-based machine learning frameworks ● MongoDB change streams and Atlas triggers for reactive event driven pipelines
Grow and protect ● Scale as new data sources are analyzed ● Robust security and data sovereignty controls	**Grow and protect** ● Distributed systems architecture with auto-sharding for scale out ● Comprehensive access controls, encryption, and auditing ● Global clusters to control data residence

Table 3: *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.

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	Why MongoDB?
Data model ● Handle massive variability in content and catalog attributes, metadata, media assets, and UGC ● Quickly update schema as new products and content are offered	**Data model** ● Document data model to store rich, multi-structured data ● Flexible schema with no schema migrations ● GridFS to store binary assets in the database
User experience ● Rich queries and search as users browse the product catalog and CMS ● Never slow down under the peak loads generated by promotions, cyclical peaks, or new publications ● Never go down, always available	**User experience** ● Expressive MongoDB Query Language, faceted search, graph traversals ● Auto-sharding for horizontal and elastic scale in response to demand ● Global Clusters to colocate data close to users, and replication for resilience
Real time analytics ● Serve up personalized recommendations ● Monitor sales performance and content consumption in real time	**Real time analytics** ● Aggregation pipeline ● AI: Spark Connector, idiomatic R and Python drivers ● Dashboards: MongoDB Charts and BI Connector

Table 4: *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	Why MongoDB?
Data integrity ● Handle payment and order processing with data correctness guarantees	**Data integrity** ● Multi-document ACID transactions with snapshot isolation
Data model ● Support variability in payment data and financial instruments ● High precision number, temporal, and geospatial data types for lossless processing, sorting and comparisons	**Data model** ● Flexible document data model to store rich, multi-structured data ● Advanced BSON data types including decimal, datetime, and GeoJSON data
Real time analytics ● Fraud detection, risk profile and liquidity monitoring ● Monitor sales performance in real time	**Real time analytics** ● Aggregation pipeline ● AI: Spark Connector, R & Python drivers ● Dashboards: MongoDB Charts and BI Connector
Resilience & Latency ● Maintain availability in the face of outages and planned maintenance	**Resilience & Latency** ● MongoDB replica sets and Global Clusters to distribute data across multiple regions
Data Integration ● Integrate eCommerce apps with payment providers ● Push payment and fraud alerts to consuming systems	**Data Integration** ● MongoDB Stitch functions to call 3rd party payment provides ● MongoDB change streams and Atlas triggers for event-driven pipelines

Table 5: *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.

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 Mobile – Radically simplifies development and improves user experience. Store data where you need it, from iOS, Android, and IoT devices to your backend in the cloud – all using a single database.
- MongoDB Stitch – The best way for your applications and users to access the data and services they need. Stitch sits at the heart MongoDB's data platform, providing secure access to data hosted in MongoDB Atlas. With Stitch, you can quickly create rich, secure apps and services without app servers, web hosts, or gateways, and keep documents in sync between MongoDB Mobile 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	Why MongoDB?
Developer velocity ● Rapidly build and evolve mobile apps ● Create rich app experiences ● Feature toggles to incrementally deploy new app functionality	**Developer velocity** ● Same database on mobile devices and backend ● Document data model matches objects and JSON syntax used in code ● Single SDK to access data stored on device and in the backend ● Expressive MongoDB Query Language, secondary indexing, and aggregation pipeline on device and in the backend ● Flexible data model to support multiple app feature sets
Data portability and security ● Sync data between device and backend ● Access controls to data ● Same mobile database for Android, iOS and IoT	**Data portability and security** ● Stitch Mobile Sync to automatically propagate data between MongoDB Mobile and Atlas ● MongoDB change streams and Atlas triggers for reactive event driven pipelines ● Stitch QueryAnywhere access rules to control data visibility
Resilience and scale ● Never go down, always available ● Never slow down under the peak loads generated by new app launches	**Resilience and scale** ● Global Clusters to colocate data close to users and replication for resilience ● Stitch serverless platform: auto-scale ● Sharding for horizontal and elastic scale

Table 6: *Required capabilities for mobile apps*

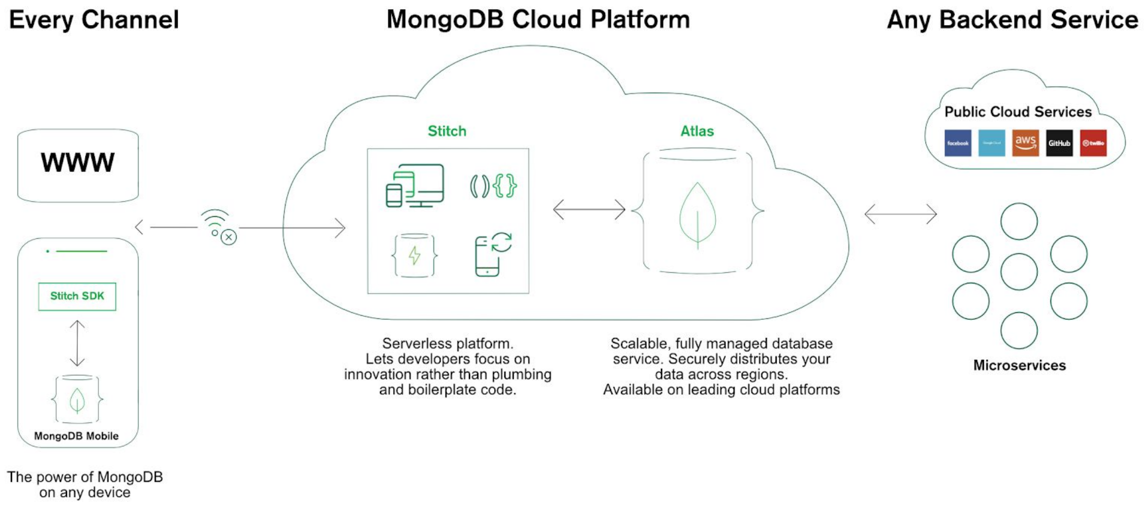
Every Channel
MongoDB Cloud Platform
Any Backend Service
WWW
Stitch SDK
MongoDB Mobile
The power of MongoDB on any device
Stitch
Atlas
Serverless platform. Lets developers focus on innovation rather than plumbing and boilerplate code.
Scalable, fully managed database service. Securely distributes your data across regions. Available on leading cloud platforms
Public Cloud Services
aws
GitHub
Microservices

Figure 5: *MongoDB data platform for mobile apps*

Mobile customer example: The Weather Channel uses MongoDB to handle 2 million requests per minute and provide real-time weather alerts for 40 million users. With MongoDB, the company's developers can get killer features out to users quickly. Changes that used to take weeks can now be pushed out in hours.

Download the MongoDB Stitch white paper to learn more about building mobile apps with a serverless application framework.

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.

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

Table 7: *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.

When Should I Not Use MongoDB?

As illustrated above, MongoDB serves most operational / OLTP and real-time 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: Unindexed Queries

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, MongoDB is often paired with a data warehouse or data lake, serving the results of those long running ad-hoc queries back to operational applications that demand real-time access to the data.

When query patterns are predictable and can be optimized with secondary indexes, then MongoDB will efficiently power analytics-heavy workloads.

In both cases, users can easily visualize results of analytics queries with 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.

Conclusion

MongoDB is the most popular and widely used modern data platform in the market. Through its natural and flexible data model, expressive query language, distributed design, and platform agnostic architecture, it is able to serve the most demanding operational and real-time analytics 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 MongoDB experts. Over 13,400 organizations rely on our commercial products. We offer software and services to make your life easier:

- MongoDB Atlas is a database as a service for MongoDB, letting you focus on apps instead of ops. With MongoDB Atlas, you only pay for what you use with a convenient hourly billing model. With the click of a button, you can scale up and down when you need to, with no downtime, full security, and high performance.
- MongoDB Enterprise Advanced is the best way to run MongoDB in your data center. It's a finely-tuned package of advanced software, support, certifications, and other services designed for the way you do business.
- MongoDB Stitch is a serverless platform which accelerates application development with simple, secure access to data and services from the client – getting your apps to market faster while reducing operational costs and effort.
- MongoDB Mobile lets you store data where you need it, from IoT, iOS, and Android mobile devices to your backend – using a single database and query language.
- 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

US 866-237-8815 • INTL +1-650-440-4474 • info@mongodb.com