Application-Driven Analytics

Defining the Next Wave of Modern Apps

November 2022

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

Table of Content

Introduction

By building smarter apps and increasing the speed of business insight, application-driven analytics gives you the opportunity to out-innovate your competitors and improve efficiency. However, it does require a shift in the way we think about and deliver analytics.

Applications run the business. Analytics manages the business. Traditionally, these functions have existed in separate domains built by different teams, serving different audiences, with data duplicated and stored in different systems. That is how things have typically worked, and to be clear, this approach is not going away anytime soon. But today, it's not enough.

That's because the digital economy demands our applications become smarter, driving better customer experiences and bringing more efficiency to the business by autonomously taking intelligent actions on our behalf — and doing it all in real time. Along with smarter apps, businesses want insights faster so they know what is happening "in the moment."

To meet these demands, we can no longer rely only on copying data out of our operational systems into centralized analytics stores. Moving data takes time and creates too much separation between application events and analytical actions.

Instead, analytics processing must be "shifted left" to the source of the data, to the applications themselves. We call this shift **application-driven analytics.** And it's a shift that both developers and analytics teams need to be ready to embrace.

In this paper, we will define application-driven analytics along with its use cases and business drivers. We'll then discuss the foundational capabilities needed to deliver smarter apps and real-time business insights before wrapping up with examples of customers already on this journey.

What is Application-Driven Analytics?

Developers build applications. They don't *"do analytics."* At least, that is how the division of responsibility between applications and analytics has been defined in the past.

In many organizations today, analytics is usually the job of a *different* team — composed of data engineers, data analysts, and data scientists. Those analytics folks extract data from source operational systems and transform it into standardized structures to align neatly with all of the other operational data they are slurping up. Then they load it into centralized analytics data warehouses and data lakehouses. When the data finally lands in the analytics system, a set of standardized "run the business" questions are asked of it, and, in more advanced cases, machine learning models are built from it.

Analytics "Shifts Left"

Why is this process evolving? In the digital economy, applications are the gateway into how customers experience and perceive a business. Successful modern applications are defined by their ability to drive better customer experiences, surface insights, and take intelligent action directly within the application on live operational data — *in real-time*. And it's developers who are tasked with making this happen, not analysts.

To make this shift successful, developers need a data platform — designed *for developers* — that makes it easy to process their application data in all sorts of new and interesting ways.

Does all this mean that analytics teams no longer matter? Quite the opposite. The reality is that the most valuable, relevant, up-to-date data lives in applications. Analytics teams therefore need a pathway into the platform so they can perform analytics closer to the application — using the tools they are familiar with — working with fresher data to unlock insights and take action faster.

To meet the needs of both developer and analytics teams, our data platform must be able to manage **common data sets** powering the **different use cases** built by application developers and analytics teams.

Use Cases for Application-Driven Analytics

To serve both developers and analytics teams, the underlying data platform needs to enable two use cases for application-driven analytics:

1. In-app analytics
2. Real-time business visibility

In-App Analytics
Developers build applications that perform analytics on live data directly within the operational flow of the application, enhancing user experiences and driving immediate user or app actions.

Examples of in-app analytics include serving a bespoke offer to a customer on an ecommerce site, detecting and blocking a fraudulent transaction, dynamically adjusting pricing, triggering trades based on stock price movements, and many more.

Real-Time Business Visibility
Analytics teams generate insights directly against up-to-date application data (without disrupting the application itself) using the same tools they use every day with their existing centralized analytics systems.

A major difference that analytics teams need to contend with is that application data is by its nature “messy” and not transformed into the neatly curated structures stored in data warehouses.

The focus of real-time business visibility is to optimize business processes. Examples include:

- Measuring inventory, and pricing to determine when to run a flash sale or to restock
- Monitoring production line output to optimize resource utilization
- Identifying market sectors with the largest intra and inter-day gains and losses from stock trades
- Tracking live A/B test results to analyze and optimize user journeys in an app

The Analytics Spectrum

Application-driven analytics complements existing analytics processes built around centralized data warehouses and data lakehouses. In **no way** does it replace them.

In Table 1 below, we identify required capabilities across the spectrum of different classes of analytics. These are designed to help you match appropriate technologies and skill sets to each business use case you are building for.

	Application-Driven Analytics		Centralized Analytics (Data Warehouse / Lakehouse)
	In-App Analytics	Real-Time Business Visibility	
Built by	App developers	Data engineers	Data engineers
Consumed by	Application users (humans, algorithms, machines)	Business analysts, data scientists, and LoB decision makers	Business analysts, data scientists, and LoB decision makers
Decision scope	Per-operation / individual	Per-process	BU-wide
Avg. records processed per query	Hundreds to thousands	Thousands to tens of thousands	Hundreds of thousands to millions
Number of data sources	One	Several	Many
Query complexity	Low	Medium	High
Query frequency*	Continuous	Multiple times per hour	Multiple times per day
Query latency*	< 1 second	Seconds	≥ minutes
Query concurrency*	≥ Thousands	Tens to hundreds	Tens
Data update frequency	Continuous	Multiple times per minute	Multiple times per day
Data latency (freshness)	≤ Seconds	Seconds to hours	Hours to weeks

* Note that centralized analytics systems can achieve similar performance levels to app-driven analytics by pre-computing and caching query results in materialized views. However, this comes at the expense of reduced update frequency and data freshness.

<table>
<tr><th colspan="2">Application Driven Analytics</th></tr>
<tr><th>In-App Analytics</th><th>Real-Time Business Visibility</th></tr>
<tr><td colspan="2">Foundational Capabilities

1. Flexible data model to natively store, enrich, and analyze data of any structure.

2. Versatile query engine supporting almost any query shape — from point queries to sophisticated data processing pipelines and relevance-based search, recommendations, and discovery.

3. Distributed cloud-native architecture to scale out large data sets, parallelize complex queries across nodes and data partitions, isolate operational from analytical workloads, and land insights close to users.

4. Integration with industry-leading data visualization, streaming, data science, and machine learning tools.

5. End to end security and governance with access controls, audit logs, and encryption of data in-flight, at-rest, and in-use to protect sensitive user data and corporate IP.</td></tr>
<tr><td colspan="2">Unique Capabilities</td></tr>
<tr><td>Low-latency processing of individual records or to an aggregated subset of records (typically <1,000)</td><td>High-throughput processing of thousands of records</td></tr>
<tr><td>Continuously analyze live operational data, without impacting the application</td><td>Query and blend live with archived operational data, without impacting the application</td></tr>
<tr><td>Support data structures optimized for the ingest, storage, and analysis of time series data streams</td><td>Support data structures optimized for integration into downstream centralized analytics systems</td></tr>
</table>

In Table 2 above, we focus on application-driven analytics to identify both the foundational and unique requirements for how data needs to be managed.

How MongoDB Helps

As application-driven analytics becomes pervasive, the MongoDB Atlas developer data platform unifies the core data services needed to make smarter apps and improved business visibility a reality.

Atlas does this by seamlessly bridging the traditional divide between transactional and analytical workloads in an elegant and integrated data architecture. As shown in Figure 1, what you get with MongoDB Atlas is a single platform managing a common data set for both developers and for analysts.

With its flexible document data model and unified query interface, the Atlas platform minimizes data movement and duplication, eliminates data silos and architectural complexity while unlocking analytics faster and at lower cost on live operational data. It does this while meeting the most demanding requirements for resilience, scale, and data privacy.

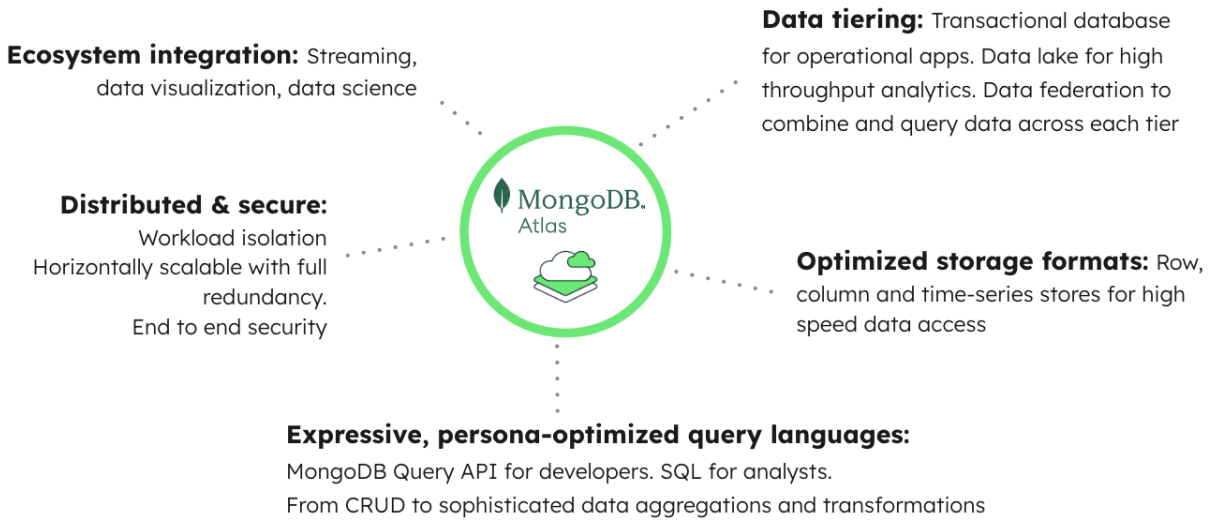


Figure 1: MongoDB Atlas combines transactional and analytical processing in a multi-cloud data platform.

In the following section of this paper, we will dig into how MongoDB Atlas meets the demands of application-driven analytics, further exploring the platform capabilities shown in Figure 2 below.

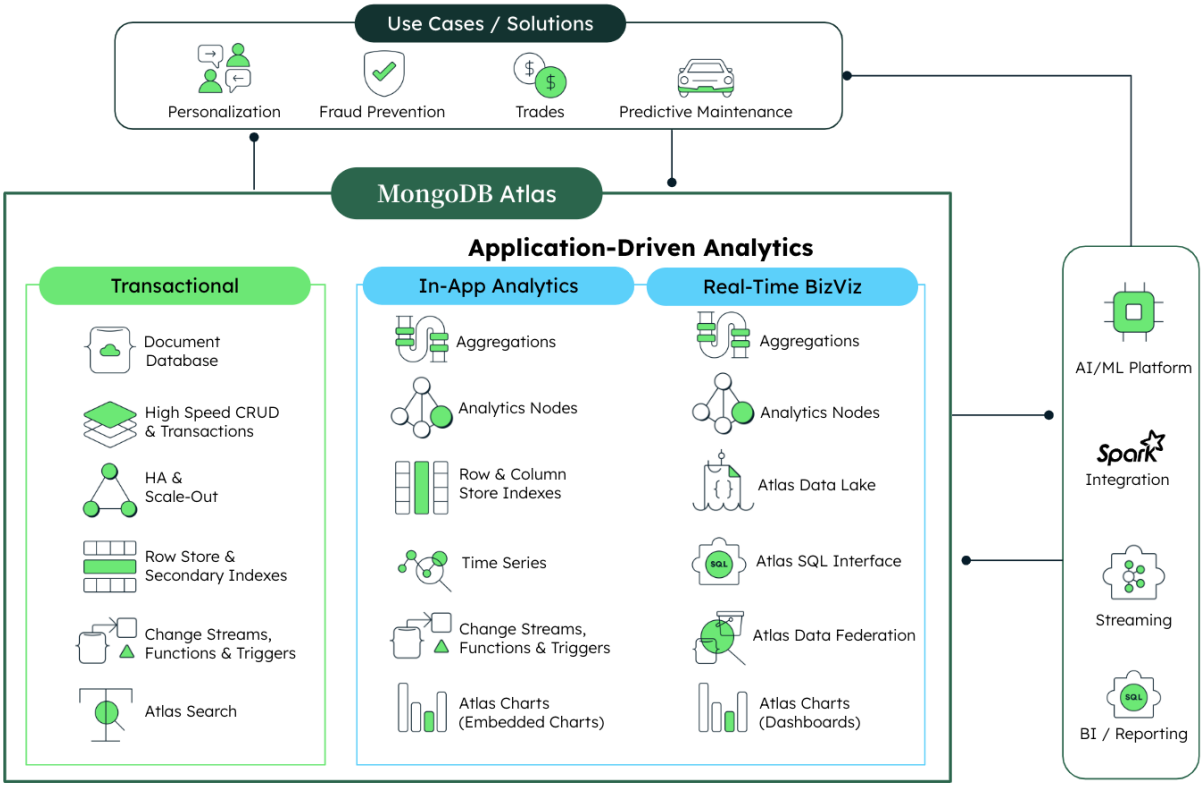


Figure 2: Mapping MongoDB Atlas platform capabilities to the transactional and analytical sides of a modern application.

Foundational Capabilities for Application-Driven Analytics

As shown in Table 2 previously, in-app analytics and real-time business visibility share a common set of requirements that MongoDB Atlas is designed to address. These are discussed below.

1. Flexible Document Data Model: Store and Analyze Data of Any Structure

Unlike the tabular data model of relational databases and data warehouses, MongoDB is built around the document data model. Documents offer productivity, agility and performance advantages for developers and data engineers. This is because:

- Documents map directly to objects in code and so are much more natural to work with.

- You can persist and immediately start working with data in its native form without lengthy upfront schema design and without having to maintain burdensome and opaque ORM layers.
- Documents support massive diversity in data types and data structures, enabling you to handle almost any use case. Fields can vary from document to document and each can be independently indexed, queried, and modified. This flexibility allows you to efficiently access and analyze any single piece of data, as well as enrich and backfill data in documents as your applications evolve.
- You can modify your documents' structure at any time allowing you to continuously integrate new application functionality without having to coordinate complex schema changes across databases, data warehouses, and data marts.
- Taking advantage of data locality, documents store data that is accessed together, reducing both read and write latency.

2. Versatile Query Engine: Operational and Analytical Processing

The MongoDB Query API and aggregation framework is implemented in the methods and functions of all the leading programming languages. For developers, the query API therefore feels like an extension of those languages, enabling them to work productively with data as code.

For analysts, the Atlas SQL Interface, discussed later, means they continue to work with MongoDB using their familiar SQL skills and tools.

With the versatility of the MongoDB query engine you can run simple point queries for lightning fast lookups through to building modular, multi-stage aggregation pipelines. With the aggregation pipeline you can perform powerful real-time analytics and transformations over your data, as well as run ad hoc, exploratory queries. Aggregations are used for many different data processing and manipulation tasks. For example, you can:

- Filter, group, join, and sort data
- Search data sets and surface recommendations
- Calculate moving averages and cumulative sums over rolling time windows
- Traverse graph structures to find hidden relationships in your data
- Process geospatial coordinates

- Cleanse and transform data within the database
- Mask and obfuscate sensitive data
- Support machine learning frameworks for efficient model building and serving
- Write out result sets to populate materialized views, power reporting dashboards and visualizations, integrate with data engineering pipelines, and more.

MongoDB Atlas does not constrain you to working only with data in hot database storage. With Atlas Data Federation, also discussed later, your queries can span multiple databases — even if they live on different cloud providers — and cloud object stores, enabling you to create refined data sets combining live and archived data.

The aggregation framework is one of MongoDB's most powerful capabilities. In addition to product documentation, the Practical MongoDB Aggregations ebook is a great resource to help developers and data engineers get started.

3. Distributed, Cloud Native Architecture: Isolate and Scale any Workload, Land Insights Close to Users

MongoDB is built on an elastic and distributed systems design. This means that out of the box you can replicate and scale-out large data sets and workloads across multiple nodes, providing fault tolerance along with horizontal scalability.

Intelligent data placement policies enable you to control precisely where data is stored so that you can comply with data residency regulations and land analytics physically close to your users. It also allows you to isolate transactional and analytical workloads from one another within Atlas:

- Atlas Analytics Nodes are co-located as part of your regular Atlas database cluster but are physically isolated from nodes supporting the operational workload. This means that analytic queries work on identical replicas of fresh application data but do not compete for system resources with nodes powering the application. Analytics nodes can also be sized and scaled independently from the rest of the cluster, allowing you to optimize price/performance for different workload classes.
- Cluster-to-Cluster Sync provides even higher levels of isolation by continuously replicating live operational data from the transactional Atlas database cluster to an entirely separate cluster dedicated to analytics processing.

- Atlas Data Lake (currently in technology preview and discussed in more detail later), is another option for workload isolation — more typically used for real-time business visibility. Atlas Data Lake is best suited for analytics with long-running queries against large data sets where you need to balance price/performance against data freshness. Atlas Data Lake ingests snapshots from your live database, transforms it into an analytics-optimized columnar format and persists it onto low-cost cloud object storage. From there, you can then query the data in an environment that is completely isolated from your live application.

4. Platform Integrations: Infusing Analytics across Your Application Estate

Any new part of your technology stack needs to integrate with your existing data flows, analytics tooling, and deployment platforms. MongoDB maintains connectors for ease of integration and interoperability within your environment:

- The MongoDB Connector for Apache Kafka configures MongoDB as both a source and sink within your data pipelines — whether for building reactive, event-driven microservices or for streaming data from MongoDB to centralized analytics systems downstream from your applications.
- The MongoDB Connector for Apache Spark allows Spark jobs to read from and write to MongoDB as part of your data science and data engineering platform.

SQL Integration and Native Visualization of Live Application Data

Many analytics teams have standardized on a set of SQL-based BI tools for reporting and analytics. Through the Atlas SQL Interface (preview), analytics teams can connect their preferred tools (e.g., Tableau, PowerBI) directly to MongoDB. This functionality allows them to query live application data from MongoDB for real-time business visibility and to blend operational data from MongoDB with data from other databases used in the organization.

Although the BI tools used by analytics teams are incredibly powerful, they can also incur cost and complexity for users who want to visualize MongoDB data as part of a regular operational application. For those users, we developed MongoDB Atlas Charts.

Atlas Charts is a data visualization service that natively supports richly structured JSON data. You can easily create charts, graphs, and dashboards in a drag-and-drop interface and share them with other users for collaboration. You can also embed them directly into your applications to create engaging user experiences with in-app analytics — for example, creating leaderboards that are refreshed in real time.

Atlas Charts can be configured to read from analytics nodes, ensuring no impact to operational workloads, and from data archived in the Atlas Data Lake. By blending live application data with historical data, you can easily provide business users with analysis of business trends over time, with charts refreshed in real time as data changes in the application.

Figure 3: Atlas Charts in action — visualizing real-time product usage metrics for Meltwater. [Read the full case study.]

5. End-to-End Security: Protecting Your Users and Your Business

Thousands of organizations — from startups to industry leaders in the Fortune 100 — entrust MongoDB with sensitive business and user data.

MongoDB takes this responsibility seriously and is dedicated to making every effort to protect customer data, including continually improving security processes and controls, as well as upholding transparency with regard to data usage. MongoDB is also committed to delivering the highest levels of standards conformance and regulatory compliance as part of our ongoing mission to address the most demanding security and privacy requirements of our customers. You can learn more from the MongoDB Trust Center.

Meeting the Demands of In-App Analytics

Beyond the foundational capabilities described above, MongoDB Atlas also offers row and column store indexing, along with specialized optimizations for the processing and analysis of time series and multi-dimensional panel data. Collectively, all of these capabilities enable developers to deliver analytics on live data directly within the operational flow of the application.

Low Latency Data Access: Row and Column Store Indexes

Whether retrieving a single document in a collection (e.g., a specific customer by their unique ID) or a subset of documents (e.g., all customers within a specific region) indexes provide efficient access to your data, avoiding the need to scan every record.

With MongoDB, you can declare indexes on any field or compound of fields within your documents, including fields nested within arrays. Specialized indexes are available for geospatial processing, clustered indexes for time series data analysis, and relevance-based full-text search. Atlas Search includes features such as “more like this,” allowing developers to quickly and easily surface personalized recommendations that increase user engagement to drive increased click-throughs and conversions.

The column store index is a new index type coming soon to MongoDB. Column store indexes speed up large-scale, ad hoc analytics queries that aggregate specific fields across most or all documents in a collection. Examples include computing counts, averages, and min/max values (e.g., maintaining a running sales total and average sales price over the duration of a product promotion).

The actual performance improvement compared to using regular row-based indexes will vary based on workload, but internal testing on data sets that exceed physical RAM in an Atlas node shows a 15x speed-up. This improvement was achieved without having to modify the structure of our documents or adapt our queries to use column store indexes.

Time Series Analytics

Time series data is becoming increasingly common in modern applications. Internet of Things (IoT) sensor data, financial trades, clickstreams, and logs are all valuable sources of insights and analytics for every business.

However regular databases struggle to meet the unique ingest, storage, and processing demands of time series data. For this reason, MongoDB developed the highly optimized time series collection type and clustered indexes. Built on a highly compressible columnar storage format, time series collections can reduce storage and I/O overhead by as much as 70%.

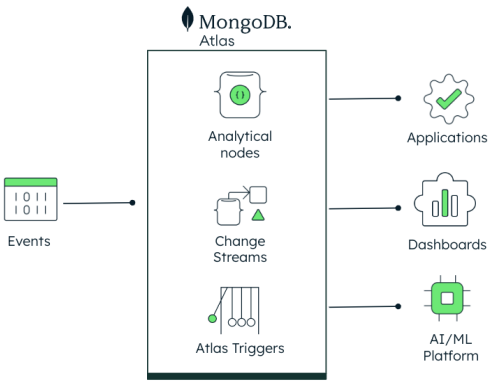
We combine time series collections with the power of the MongoDB Query API, introducing features like densification and gap filling to automatically populate missing data points. Window functions allow you to run analytical queries like moving averages and cumulative sums to uncover hidden patterns in voluminous time series data sets.

With Atlas Online Archive, aged data can automatically be tiered out of hot time series collections to low cost object storage, while preserving the ability to query the data at any time. This ability to blend and analyze newly ingested time series measurements with cooler data helps you unlock the potential of new data-intensive applications such as the IoT in ways not possible with regular databases.

Event-Driven Analytics

Increasingly applications must be able to continuously analyze data in real time as they react to live events. Dynamic pricing in a ride-hailing service, recalculating delivery times in a logistics app due to changing traffic conditions, triggering a service call when a factory machine component starts to fail, or initiating a trade when stock markets move. These are just a few examples of in-app analytics that require continuous, real-time data analysis.

MongoDB Atlas has a host of capabilities to support these requirements:



- With change streams, all database changes are published to an API, notifying subscribing applications when an event matches a predefined criteria.
- Atlas triggers and functions can then automatically execute application code in response to the event, allowing you to build reactive, real-time, in-app analytics.

Figure 4: Continuous data analysis in response to incoming application events.

Meeting the Demands of Real-Time Business Visibility

Real-time business visibility builds on the platform foundations described previously adding the tools that help analytics teams unlock insights and take action on application data in place, without having to move it all into centralized analytics stores.

Atlas SQL Interface

The Atlas SQL Interface (preview) enables analysts to leverage their existing SQL skills and tools to query data in MongoDB Atlas. The Atlas SQL Interface means analysts do not have to learn the more developer-centric MongoDB Query API. It also means they do not have to sacrifice real-time business visibility by ETL-ing data from MongoDB into a SQL-based analytics data warehouse or data lakehouse.

Analysts can work with rich, multi-structured documents without first having to define a schema or flattening their data. The SQL Interface leverages `mongosql`, a powerful, SQL-92 compatible dialect that's optimized for the document data model.

Using a familiar language and syntax, analysts can build SQL queries that are as intuitive to them as working with a traditional relational database, but that are compatible with — and translated to — the underlying MongoDB aggregation framework. With the Atlas SQL

Interface, they can quickly surface insights and create new data streams without time-consuming data manipulation.

Through our connectors, the Atlas SQL Interface enables tools such as Tableau, Looker, and Power BI to access and visualize data directly from MongoDB Atlas. JDBC and ODBC drivers are also available for analysts to connect their own SQL-based applications. SQL queries can be federated across live Atlas databases and Atlas Data Lake to help analysts gain complete, 360-degree visibility of operational data distributed across the business's application.

Atlas Data Lake

Atlas Data Lake is a fully managed storage solution that is optimized for complex analytics over large data sets while delivering the low-cost economics of cloud storage.

As data is ingested directly from Atlas databases' backup snapshots and customer-owned cloud storage buckets (coming soon), Atlas Data Lake automatically reformats it into an analytics-optimized columnar storage format, then partitions and indexes it. This creates an analytics environment that is completely isolated from your production clusters. Data is queried via Atlas Data Federation, discussed below.

Atlas Data Federation

Atlas Data Federation is a distributed query engine that allows you to natively query, transform, and move data across multiple sources inside and outside of MongoDB Atlas. Data sources can include one or more MongoDB Atlas databases (even if they are hosted on different cloud providers), Atlas Data Lake, Amazon Simple Storage Service (S3) buckets (with additional cloud providers coming in the future), and HTTP endpoints.

Data movement between different sources and storage tiers is completely transparent to your applications, avoiding the need to make any code changes.

Supporting the MongoDB Query API and Atlas SQL Interface, both developers and data engineers can make use of Atlas Data Federation. You can use Atlas Data Federation to:

- Transform Atlas cluster data into Parquet, JSON, BSON, or CSV files written to Amazon S3 buckets. This functionality is often used to integrate MongoDB application data into data engineering pipelines.

- Query and aggregate data across multiple Atlas clusters, Atlas data lakes, and Amazon S3 buckets to get a holistic view of your data (for example, building a 360-degree view of your customers)..
- Read and import data from your Amazon S3 buckets into an Atlas cluster.

There are multiple ways you can put Atlas Data Federation to work. For example, consider building a model to forecast future product demand:

- You have closed sales, order backlog, and sales commits for the current fiscal year with each stored in separate MongoDB Atlas databases.
- Historic sales data from previous fiscal years is archived into Atlas Data Lake.
- Independent market forecasts are sitting as Parquet files in an S3 bucket.

With Atlas Data Federation, you can access all of these different data sources in a single query to build your forecasting model, all without the need to first move or transform data, or change the query as data moves between sources.

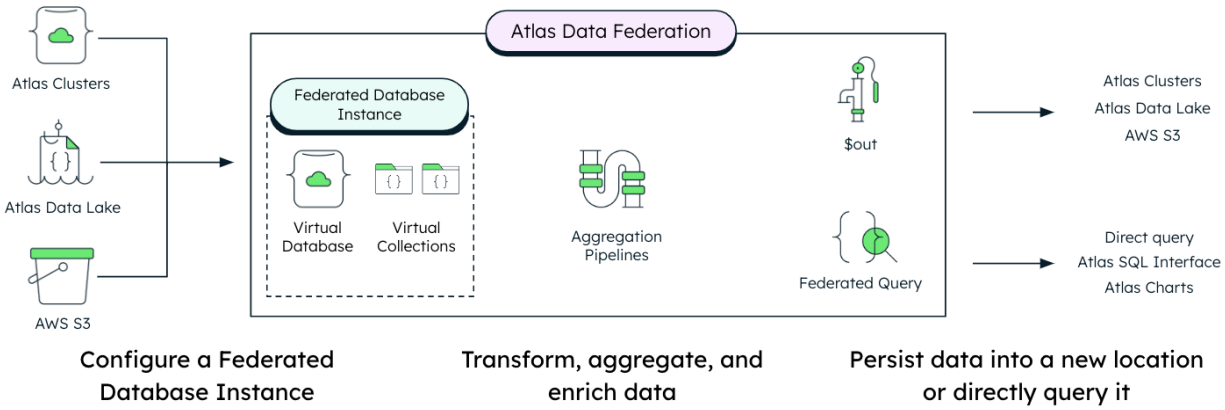


Figure 5: Atlas Data Federation combines and enriches data from multiple sources so you can directly query it or persist the results to another data store, eliminating fragile ETL processes.

Continuing with our forecasting example, a second Atlas Data Federation use case is for ETL-free data engineering. Query results can be persisted back into your live database cluster for real-time user consumption or written out into S3 in an analytics optimized format, such as Parquet, ready for model training in an AI platform.

Customer Case Studies

MongoDB counts 37,000+ customers globally across all industry sectors and sizes, including more than 50% of the Fortune 100. Many customers started out using MongoDB as an operational database for both new cloud-native applications as well as modernized legacy apps. More and more of these customers are now enriching their applications with new analytics functionality powered by MongoDB Atlas, blending in-app analytics with real-time business visibility.

Preventative Maintenance and Smart Factory

As one of the world's largest engineering companies, Bosch has been at the forefront of IoT adoption, using sensors to bridge the divide between the physical and digital worlds. The Bosch IoT Suite is the platform used by Bosch business units and its partners for the storage and analysis of IoT sensor data.

Projects range from predictive analytics used for preventative maintenance in connected vehicles through to smart factories with analysis of equipment usage in assembly lines. MongoDB sits at the heart of the Bosch IoT Suite, ingesting, storing, and processing streams of sensor and device data, and readying it for analytics processes. Alerts are triggered when anomalous events or defects are detected, and users can generate visualizations that show telemetry trends over time windows. Learn more.

Information Discovery and Intelligence

Iron Mountain is used by almost all of the Fortune 1000 to store and protect some of their most valuable information and assets. Over the past decade, the company has evolved from a traditional physical asset storage and shredding company to offering intelligent document processing (IDP) AI/ML-based solutions.

The Iron Mountain InSight platform digitizes, ingests, and processes millions of records per day to classify, enrich, and extract metadata, storing it in MongoDB Atlas. This enables Iron Mountain's customers to unlock the data they're storing to solve business problems and aid business processes. MongoDB's application-driven analytics and time series capabilities are central to the solution, allowing rapid insights and reporting to be

generated from the underlying data, without the latency and complexity of having to ETL data from operational to analytical systems. Learn more.

Relevance-Based Property Search and Sales Dashboarding
With hundreds of billions of dollars in property sales every year, Keller Williams (KW) is one of the world's largest real estate companies. It relies on a sophisticated cloud-based technology platform to power more than 180,000 web sites personalized for each of its agents, with MongoDB Atlas backing its microservices architecture.

As new property details are ingested into the platform, the MongoDB aggregation pipeline cleanses, transforms, and then persists them into a materialized view. From there, the KW search platform and Atlas Search indexes the property details, making them discoverable to buyers through sophisticated, geospatial searches against the property catalog. MongoDB also persists clickstreams from user sessions so that the business can visually analyze interest in each property along with performance of its agents. Learn more.

Fleet Management and Geospatial Processing
MongoDB Atlas powers the Volvo Connect customer portal, used by fleet managers to monitor vehicle status, generate reports, and plan routes – all providing real time visibility for the business.

Telemetry is collected from one million Volvo trucks around the world generating 65 million events per day, scaling to 2 billion events by 2024.

Volvo Connect was originally built on top of Oracle. As part of Volvo's platform modernization strategy with a move to microservices, MongoDB was selected for data model flexibility, scalability, and advanced analytics functionality. Learn more.

Single, 360-Degree View of the Customer at Sýn Vodafone: ROI in 21 Days.
As part of its initiative to improve customer experience and accelerate the delivery of new services, Sýn embarked on a complete overhaul of its backend systems. MongoDB Atlas was selected to power both transactional and business visibility workloads.

A key objective of Sýn's project has been to build a single, 360-degree customer view across all company touchpoints. To do this, they ingest customer data from their internal

marketing, CRM, web, and ecommerce systems into Amazon S3. Atlas Data Federation is then used to merge this disparate data into a single view. This allows the company's marketing team to get real time answers to questions on their customers such as "How do they use our websites? How do they use the marketing suite? Do they respond to emails or calls? What is the correct approach for them?." These answers help shape marketing strategy, with the marketing team using Atlas Charts to track campaign performance.

Sýn estimates that MongoDB Atlas delivered a Return On Investment (ROI) in just 21 days. Learn more.

Deep Learning for Autonomous Driving

Continental AG has been at the forefront of vehicle braking and tire systems for decades. The company is now leveraging that knowledge in building autonomous systems in a project it calls "Vision Zero," which strives for zero fatalities or accidents on our roads.

MongoDB is the underlying data layer powering the company's AI and deep learning systems. To collect, store, and process massive volumes of complex data, Continental needed a data platform that is flexible, developer friendly, and seamlessly scalable.

Data generated by sensors, radars, and cameras is complex and multi-structured, and it changes rapidly based on new configurations or prototypes. Machine learning frameworks rely on iterative feature engineering based on this evolving data to train and tune new models. With its flexible document data model and powerful query engine to preprocess the data ready for deep learning, MongoDB is the perfect fit. Learn more.

Getting Started

Application-driven analytics is defining the next wave of modern applications. Developers will need to work with data in ways that were previously the domain of dedicated analytics teams. At the same time, those same analytics teams will need direct access to source operational data in order to create fresher, real-time business visibility.

The MongoDB Atlas developer data platform is engineered to help both teams ride this new wave – leading to smarter apps and increased business visibility.

The best way to get started is to sign up for an account on MongoDB Atlas. From there, you can create a free database cluster, load your own data or our sample data sets, and explore what's possible within the platform. The MongoDB Developer Center hosts an array of resources including tutorials, sample code, videos, and documentation organized by programming language and product.

MongoDB also offers a range of instructor-led and self-paced training programs for developers and data engineers, with each module taking around 1-day to complete:

- The recommended path for instructor-led training from MongoDB Professional Services takes attendees through MongoDB fundamentals into data storage and retrieval, advanced query and data processing, and Atlas Data Federation.
- MongoDB University provides online, self-paced training that gets attendees started with MongoDB, the document data model, query API, performance tuning, and more.

In addition to training, MongoDB also provides a range of consulting services that can work with your teams at any stage of your project – from initial design and architecture through to optimizing applications already running in production.

Collectively, these resources and services will help you better meet user expectations with smarter apps while driving more opportunity with real-time business visibility.

Safe Harbor

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

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