

# How Financial Services Achieve A Strategic Advantage With Data- Driven Disruption

Seven Industry Leaders Reveal Why They Choose  
MongoDB



# From Mainframe to Multi-Cloud

Why data flexibility and portability matter

# Why leaders choose MongoDB

**From crypto startups to tier 1 banks, the financial services industry trusts MongoDB to liberate data, empower developers, and embrace disruption.**

Whether its trading platforms and end-to-end digital loan origination, or AI/ML-driven fraud detection systems and financial 'super apps', MongoDB is replacing relational databases and mainframes as the data platform financial services use to modernize legacy workloads and build leading experiences for their customers.

We've moved [32 banking apps to AWS in 30 days](#) and helped [streamline 65 relational databases](#) at a global bank into one, proving

MongoDB as the new standard for financial services firms as they transition from legacy technologies to hybrid-cloud, cloud, and multi-cloud architectures.

One of North America's top 5 largest banks recently chose MongoDB as its database standard to fuel modernization, improve uptime, and power a highly available, always-on secure customer experience for tens of millions of retail customers.

The bank runs over 200 applications on MongoDB across digital, capital markets, consumer lending, risk, and payment divisions. Use cases span mainframe offload, operational data store, single view of the customer and many more.

At the other end of the spectrum, challenger banks like Current and illimity, and global payment solutions providers like Splitit rely on Atlas, MongoDB's multi-cloud developer data platform, to launch, scale, and quickly iterate their cloud-native, data-driven businesses.

From challengers to established players, mainframe to multi-cloud, MongoDB helps financial services unleash the power of software and data.

## Contents

1. ["A flick of the switch and the payments started flowing"](#)
2. ["This supports our multi-cloud future"](#)
3. [Powering a "Super App" for 5 million retail bank customers](#)
4. ["Ease of use, flexibility, and the scalability we need"](#)
5. ["The primary driver was development velocity"](#)
6. [Escaping the monolithic mainframe](#)
7. ["We've demonstrated to the finance industry a better way to use data."](#)



# The Developer Data Platform (Atlas)

A disruptive force for data and financial services

# What is a developer data platform?

Cloud is the new norm, and cloud-native data warehouses are now massively parallel-processed. Data pipelines can handle terabytes of data. Storage is cheap, AI/ML applications are everywhere, and data-processing frameworks like Spark can handle large volumes of data.

To master these changes, financial services firms need a consolidated way to work with data, even as they're building a wide range of applications in a variety of contexts.

Built to meet these challenges, a developer data platform introduces an integrated set of database and data services, removing much of the overhead of managing a data infrastructure and, in the process, boosting

developer productivity and innovation.

A developer data platform also eliminates the need for niche databases that serve only a small class of use cases, simplifying an organization's data infrastructure and doing away with a raft of ETL work.



“Every new technology added to the stack is an exponential increase in the number of system component interactions we need to design, test, monitor, and maintain. We keep our technology footprint small so our feature output can be large.”

Trevor Marshall, the CTO of Current, a leading U.S. challenger bank.



Unlike the sprawling, ‘spaghetti’ architectures found in many financial services firms, an integrated developer data platform empowers development teams with a single, unified experience. There is only one

operational and security model to master, and that model maximizes abstraction where possible.

## MongoDB's developer data platform (Atlas).

MongoDB provides a developer data platform ([Atlas](#)) designed to help development teams move fast and reduce their underlying data infrastructure complexity. Atlas combines transactional processing, relevance-based search, real-time analytics, and mobile to cloud data synchronization in an elegant and integrated data architecture.

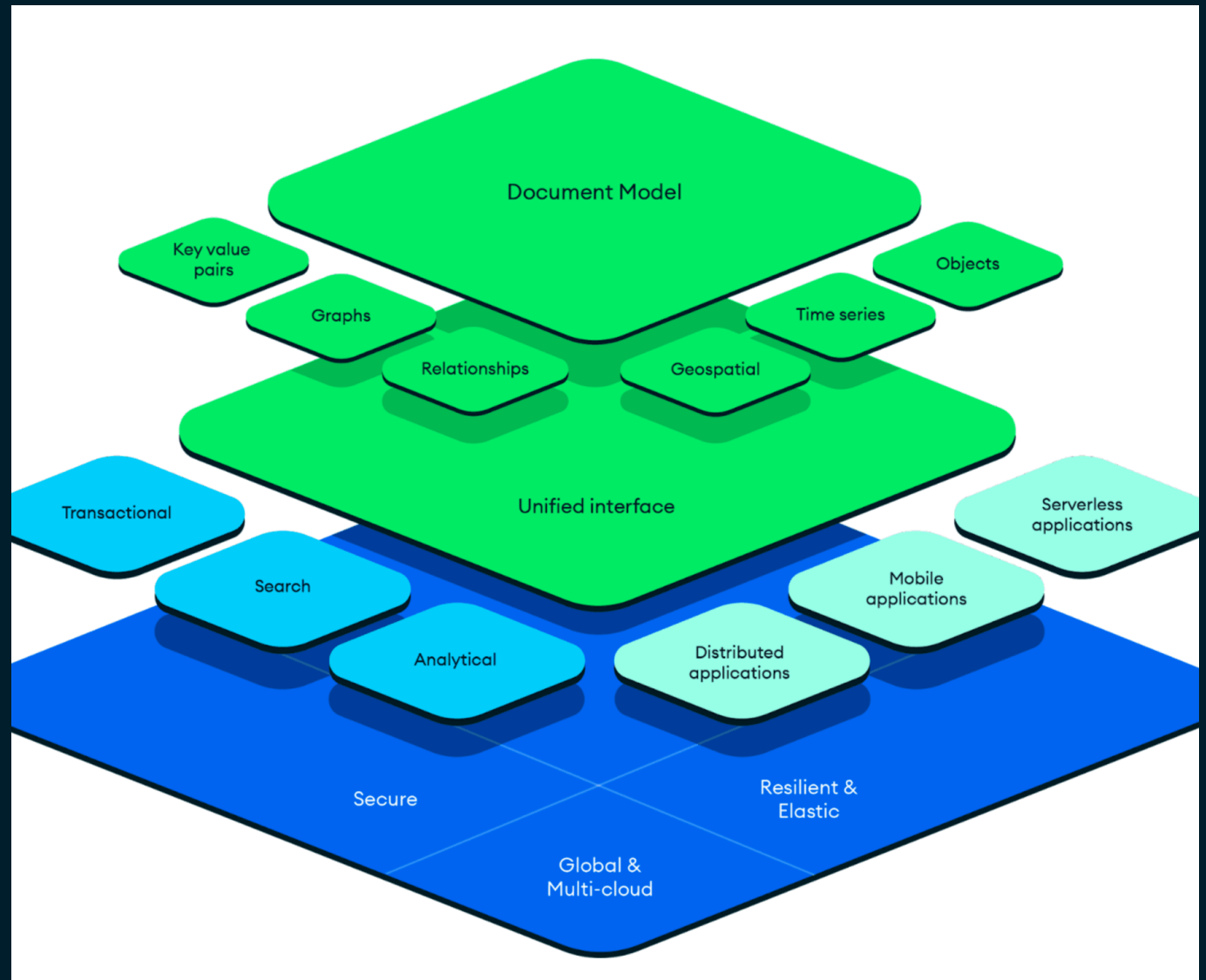
At the heart of Atlas is a database built on a [flexible document data model](#) that makes it easier to model and remodel financial data as regulation or application requirements change. Crucially, MongoDB has the

transactional and data integrity capabilities, like ACID compliance, needed to actually replace your relational databases, not just supplement them.

Atlas, the developer data platform integrates the capabilities of multiple specialist databases and data services, including:

- Full-text search
- Local and edge data persistence
- Built-in mobile data synchronization
- Real-time analytics
- Data lake
- Data visualization

Finally, at the core of our developer data platform is the most advanced cloud database service on the market, Atlas, which is built on a foundation of resilience, nearly infinite scalability, and best-in-class security. Gain true [multi-cloud deployment flexibility](#) and enjoy the services of multiple providers, including AWS, Google Cloud, and Microsoft Azure, with no vendor lock-in.



# "A flick of the switch and payments started flowing"

A new payments platform at Macquarie Bank





**In its native Australia, Macquarie's retail business is known not just as one of the largest in the country, but as a digital pioneer, picking up multiple awards for mobile and online banking services.**

When it was time to bring the bank onto Australia's New Payments Platform (NPP), a new national system for real-time payments, Macquarie's engineering team was laser-focused on delivering the best possible customer experience.

"The real driver of the project is the customer wanting to have access 24/7, 365 days a year," says Chris Clark, principal engineer and lead on the NPP project.

"Going offline for any length of time to perform upgrades and maintenance isn't an option, as we want to deliver a truly always-on experience for our clients."

The NPP project had to follow Macquarie's existing development and architecture best practices, which included an agile approach, with multiple MVP product releases, on a microservices architecture.



"A flick of the switch and payments started flowing. We've had over 3 million transactions now. No issues. It just simply worked."

Chris Clark, principal engineer Macquarie bank



To free its teams for higher-value work, Macquarie prefers to run its infrastructure as a managed cloud service wherever possible. For the NPP project, they chose MongoDB Atlas, MongoDB's global cloud database service. "We appreciated the freedom it gave us," says Clark. "We no longer had to ask for a new instance of a database to be created. We could do that ourselves. Disk space, RAM, CPU ... our team had control of all those features."

The flexibility of the document database allowed the Macquarie team to iterate on its original vision for the NPP platform, as the schema could accommodate changes.

The freedom to deploy across multiple cloud vendors was also key, as was the ability to avoid vendor lock-in. "We didn't want to walk through any one-way doors," says Clark.

"With this solution we can run on any cloud provider we want." Macquarie is now running workloads across both AWS and Google Cloud.

**Watch our Future of Payments webinar**



# "This supports our multi-cloud future"

Bendigo Bank Launch 32 Apps On AWS In 30 Days



## Bendigo and Adelaide Bank, one of Australia's largest banks, are all in on digital transformation.

Key to the bank's transformation is the need to reduce complexity – for both its developers and its customers. This means simplifying and consolidating a sprawling tech stack.

“As we work to accelerate the transformation of our business, we believe the benefits of cloud will help our business systems by reducing disruption, improving velocity and consistency, and enhancing our risk and vulnerability management position,” said Ash Austin, Bendigo and Adelaide Bank's cloud platforms service owner.

With the ability to run across all three major hyperscalers [Google Cloud, AWS, and Azure] MongoDB Atlas gave Bendigo the flexibility they needed to embrace the cloud without being locked in to a particular vendor.



"It made it really easy for us to choose MongoDB because we didn't have to then hedge our bets on a particular cloud provider or a particular process; we could be flexible."

Dan Corboy, Cloud engineer Bendigo and Adelaide Bank



After steadily ramping up its digital transformation work, in December 2020, the bank kicked the project into high gear, successfully completing the migration of 32 of its banking applications to AWS in just 30 days.

Three-quarters of the “32 for 30” apps the bank needed to migrate to AWS were running on MongoDB. Using Mirror, a native MongoDB migration tool, the team was able to move those apps seamlessly to AWS.

### How to build a multi-cloud strategy:

[Video] [Cross Cloud Redundancy on MongoDB Atlas Explained](#)

[Blog] [Solve Cloud Concentration Risk with Multi-Cloud](#)

[White paper] [Why The World is Going Multi-Cloud](#)

[Documentation] [Create a multi-cloud cluster with MongoDB Atlas](#)

# Powering a "Super App" for 5 million customers

A financial powerhouse builds their mobile future





## A global leader in retail banking wanted to use its digital channels to build more personal relationships with its customers.

The bank's primary tool for relationship-building was a new mobile app, able to handle everything from complex transactions to online consultations.

The app is designed to both improve and coordinate the tools used jointly by banking managers and customers and, importantly, to simplify customer interactions with the bank.

The new mobile app was created in just a few months. It interacts with an [ODL](#) built on MongoDB, which sits in front of the core banking platform.

The app has several awards for user experience, and now has nearly 5 million users.

The bank chose an ODL for its immediate benefits, and also because an ODL is an easily scalable solution that can meet future demands for expansion. The bank plans to use the ODL to support additional services at a later stage.

De-risk your move away from the mainframe with an ODL in [The Five Phases of Banking Modernization](#)

A wide range of new functionality, using MongoDB's scalability, low latency, flexible data model, and built-in data analytics services, places the new app squarely in the lead in mobile banking.

But more important, it allowed the bank to assemble a group of applications and quietly merge them into a super app. Here's a sampling of what the new application can do:

**Financial planning assistance:** A financial timeline helps customers take control of their finances by showing upcoming transactions such as bills and subscriptions, mortgage payments, and taxes due.



**Global search engine:** A global search engine makes it easier to call up completed transactions and other historical information. Using artificial intelligence, the app helps customers monitor savings and expenses, and plan future payments. It also makes personalized recommendations for each client.

**Credit card management:** Customers can call up the geographic location of recent purchases.

**New budgeting tools:** The app helps customers set a budget for monthly expenses and performs a comparative analysis of the current and previous months' expenditures. Customers can choose an advanced analysis of their finances and receive money-saving tips ranging from food-shopping advice to strategies for saving money while on vacation.

**Subscription management:** Customers can manage their subscriptions to third-party services such as Netflix and Spotify,

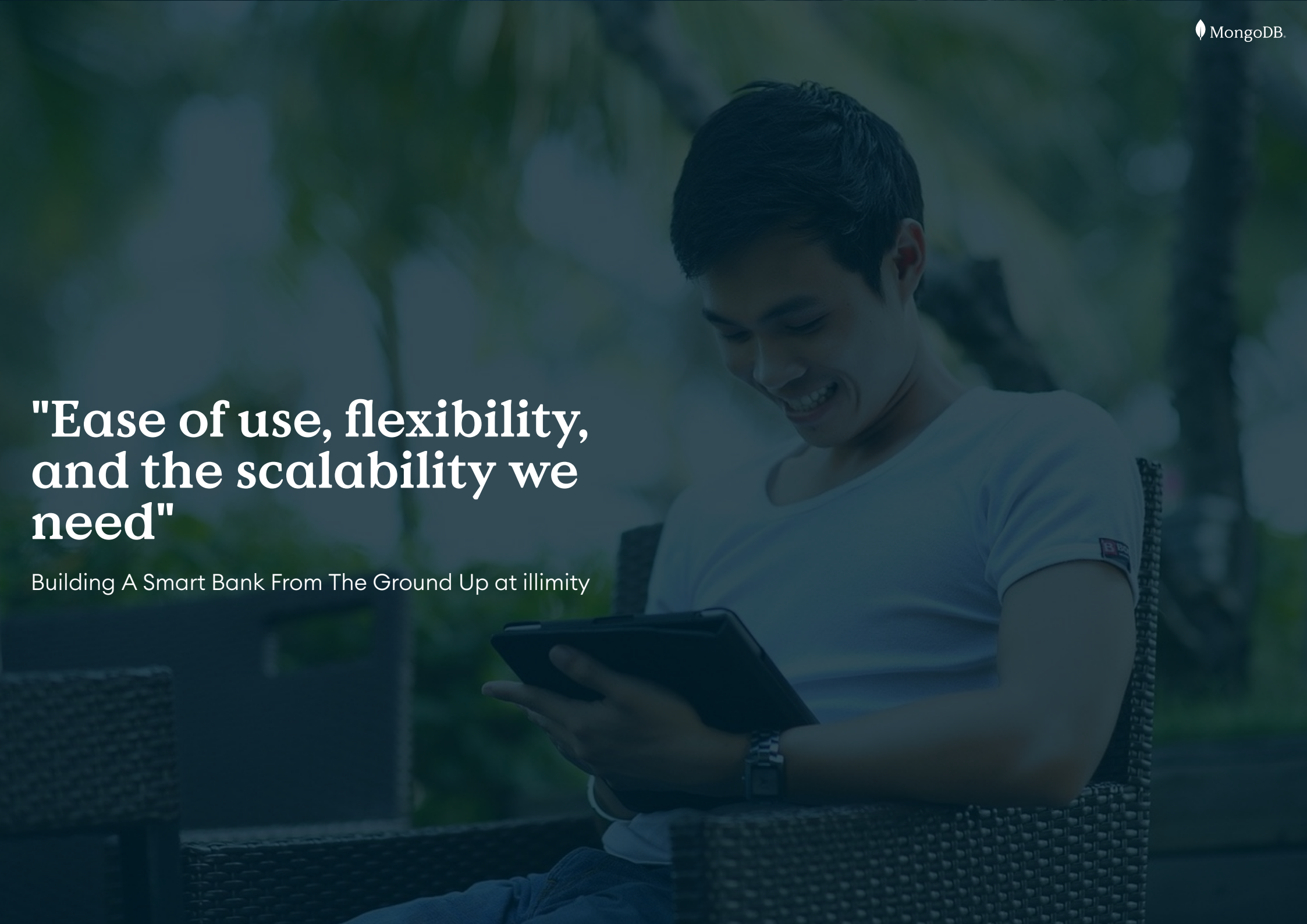
administer recurring payments such as loans, and set proactive alerts to keep control of their expenses.

**Digital piggy bank:** The digital piggy bank allows customers to define savings goals, automate savings rules, and track progress toward their goals.



# "Ease of use, flexibility, and the scalability we need"

Building A Smart Bank From The Ground Up at illimity





## Challenger bank illimity was founded by banking experts committed to overcoming the challenges of legacy banking.

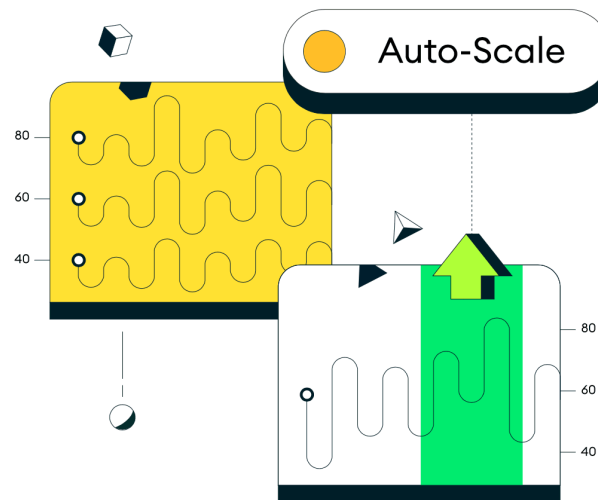
Determined to design a superior experience from the inside out, the illimity team focused on users, including internal users, such as anti-money laundering specialists and compliance professionals.

### Illimity is built on the modern MACH principles of:

- **Microservices** for communication, including open banking as the interface for all services.
- **API-driven interfaces** between different business functions.
- **Cloud-native technologies**, which were used from the start, as illimity had no legacy systems holding them back.
- **Headless architectures**, allowing Illimity to share its API-driven services with additional client banks.

MongoDB naturally fit illimity's requirements and was part of the endeavor from the beginning.

The continuous development of new products shows that MongoDB gives Illimity a faster time to market compared to products built on relational databases. illimity calls it "revolutionary but reliable."



"Today, MongoDB Atlas is offering illimity ease of use, flexibility, and the scalability we need,"

Renzo Rognoni, Team Leader Data & Analytics at illimity.



"Our staff devote significantly less time to database management. Access to information is easy and immediate," said Renzo Rognoni, team leader data & Analytics at illimity.

"The cloud hosting on Microsoft Azure allows us to look at growth without worry. All of this in compliance with the strictest data security and privacy standards," he added.

[Read our 5 phase guide to mainframe modernization](#)

# "The primary driver was development velocity"

Personalized banking from scratch at Current

**Current, a challenger bank started in 2015, began with a simple premise: It could better help its customers achieve financial success if it approached them as people rather than as account numbers.**

To bring that vision to life, Current built its own ledgering system called Current Core. Current chose MongoDB as the database best able to support Current Core, thanks to its:

- **Strongly consistent data model**
- **Enterprise security**, including client-side field-level encryption, to address security, audit, and compliance requirements
- **Multidocument distributed transactions** with ACID guarantees

Current Core features event-driven architecture that holds every transaction event in a MongoDB collection. These events construct debits and credits on the customer's ledger.



"MongoDB gave us the flexibility to be agile with our data design and iterate quickly. The primary driver was the development velocity."

Current CTO, Trevor Marshall



The platform translates all transactions, such as direct withdrawals, mobile check deposits, cash deposits, peer-to-peer payments, ATM transactions, and point-of-sale card purchases into events, then stores them in a ledger that lives in MongoDB.

"By collapsing a collection of events, you can derive the current state of the user," says Current CTO, Trevor Marshall. "This has enabled Current to usher in an era of truly

customer-focused, not legacy bank account-focused, financial services."

Search is provided by MongoDB Atlas Search, which sits right on top of the data stored in Current Core. That eliminates the overhead, such as a layer of synchronization, that would be required to integrate a separate search engine.

"Our company mission of creating better financial outcomes for people is reflected all the way down to the way data is stored in MongoDB," says Marshall. "It is uncommon, especially in financial services, for the data model to support the business so directly."

That's good for customers, but also for Current itself: It has doubled its member base in less than six months and is now approaching 3 million, increasing revenues by more than 500% in the past year.

**Watch our talk on Data Mesh for Financial Services**



# Escaping the Monolithic Mainframe

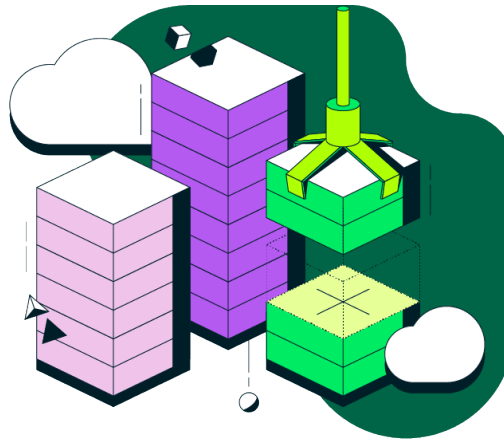
A Top 5 global bank moves to MongoDB

**In 2017, one of the five largest banks in the United States selected four databases -- MongoDB, Oracle, SQLServer, and MySQL — as enterprise-wide standards.**

Just four years later, the bank is using MongoDB for more than 125 projects across every line of business, in the front, middle, and back offices. This steady growth in the adoption of MongoDB at the expense of other databases was driven by MongoDB's abilities to solve for:

- Data residing in many source systems and the need to deliver it to multiple consuming systems
- Growth in the need for real-time data delivery
- Easier access to data for developers
- Changing requirements, creating the need for a flexible database
- High availability, resiliency, and data durability

Teams across the bank frequently use MongoDB to bring data closer to the edge via operational data stores to address needs not met by systems of record, other operational data stores, or data lakes.



Examples include:

**Credit card operational data store:** Built on MongoDB, the data store reduces dependency on a third-party mainframe, handling more than 95% of read operations.

**Card and retail integration service:** To modernize its legacy SQL Server monoliths,

the bank is developing more than 80 microservices across its card business, supported by MongoDB.

**Risk and finance data service:** An operational data store built on MongoDB distributes data from a data lake to dozens of consuming systems within the risk and finance group.

**Teller view:** A resilient caching layer ensures the durability of in-branch teller transactions in the event of a system failure.

**Enterprise payments data hub:** MongoDB serves as the operational data store for commercial payments data. This directory maps relationships between payees and payers and uses machine learning to apply confidence scores to those relationships.

**Enterprise payments execution:** Commercial payments are now executed on MongoDB after being migrated from a legacy third-party system.

**Cyber intelligence platform:** An operational data fabric enables multiple cybersecurity services, including the bank's application data explorer and its network data explorer, which protect the bank at its perimeter and also from internal threats.

Notably, the bank decided to use MongoDB to build an [ODL](#) in the aftermath of a well-publicized data center outage. During that outage, the bank's cache went down, causing middleware applications to turn directly to the mainframe, which couldn't handle the volume.

De-risk your move away from the mainframe with an ODL in [The Five Phases of Banking Modernization](#)

The bank decided to move its operational functions from the mainframe to an operational data layer based on MongoDB.

The [ODL](#) now sits between the systems of record and customers, shielding those customers from the fallout of any future data center outage. The data layer serves more than 30 million digital retail customers of the bank.

As the ODL slowly becomes a real-time data store, it will eliminate the need for various teams to build duplicate datasets, simplify the tech stack, and enable teams to streamline their application development.

For some use cases, the ODL is already becoming a true system of record. Digital apps can use APIs to access the ODL, rather than building their own ODLs containing the data they need from the system of record.

Based on the success of these and other initiatives, the bank is relying on MongoDB to support a number of new functions currently in development:

**Data-driven decision automation:** Using microservices, the bank is in the process of

automating its home loan origination process. Dozens of business projects are scheduled to be similarly automated through the end of 2022, using more than 120 microservices.

**Digital payments microservices:** Execution of retail payments such as BillPay, Me 2 Me, Zelle, and Wire will move from a monolith to a microservices-based architecture.

**ISO 20022 message gateway:** The bank will be able to send and receive payments messages externally in the updated ISO 20022 format.

**Real estate valuation service:** The bank is moving to a single home-grown valuation service, replacing a legacy system that uses three third-party and four internal applications.

**Capital markets ODS:** An operational data store will support those capital markets operational use cases not already met by existing data hubs.





**"We've demonstrated to  
the finance industry a  
better way to use data."**

Trade Ledger Disrupt Lending

## In the past, lending has been difficult and expensive.

According to Trade Ledger, 200 million companies globally have struggled to get the credit they needed, when they needed it.

That's why Matthias Born and Martin McCann got their heads together over a beer in a Sydney pub to tackle banking's most challenging technology puzzle – how to streamline lending and credit decisioning in a highly regulated, risk averse and competitive industry.

They created Trade Ledger as a lending Software as a Service (SaaS) tool, embedded within customers' online credit application process, that transforms their ability to lend to any growth business in need of capital.

By rethinking lenders' approach to managing data, the company has signed global and regional banks and alternative lenders, and

grown exponentially, tripling in size over the past year. It now employs over 120 people in Australia and the UK.



The past few years have been transformational. We've demonstrated to the finance industry that there is a better way to use data. MongoDB provided us with a great foundation for our data strategy. We've been helping financiers to lend more, at lower risk, faster and for greater profit.

Matt Born, CTO and co-founder of Trade Ledger."



Trade Ledger unlocks capital by gathering and analyzing data to facilitate credit decisions for banks and non-bank lenders.

According to Matt: "Lending is about having quality data sources to make informed decisions regarding the provision of lending facilities to businesses, and to enable lenders

to enhance their credit risk profiles on an ongoing basis as they lend."

To build a better way, Trade Ledger had to overcome some significant technology hurdles. The flexibility of MongoDB's document model allows Matt's team to bring data from multiple sources with different schemas into a single data source without writing long lines of code or breaking the database.

Matt also wanted this flexibility within a managed service which could take the operational and management burden off his small team.

MongoDB Atlas's flexibility, adaptability, and speed to market is helping Trade Ledger deliver incredible improvements – customers are reporting a 50 per cent increase in conversions with a 50-70 per cent reduction in costs.

# Further Reading

MongoDB for Financial Services





# About MongoDB

MongoDB's multi-cloud developer data platform (Atlas) is emerging as the standard for financial services firms who want to free their data, empower developers, and embrace disruption.

Atlas, our developer data platform powers transactional processing, relevance-based search, mobile applications, real-time analytics, and more through an elegant and integrated data architecture. MongoDB also provides the first-class developer experience of a flexible document data model and unified query interface, while meeting the most demanding requirements for resilience, scale, and data privacy. To learn more about MongoDB, visit [MongoDB.com's financial services page](#).

Below, we have included some additional resources:

- [\[Webinar\]](#) The future of payments: next generation payment architectures and data monetisation
- [\[Panel discussion\]](#) Modernizing the mortgage process with data
- [\[Webinar\]](#) Staying one step ahead of payments fraud with Featurespace and MongoDB
- [\[White paper\]](#) The 5 Phases of Banking Modernization
- [\[Ebook\]](#) How Financial Services Eliminate Cloud Concentration Risk