A New Era in Core Banking Data

How Temenos and MongoDB Transform the Core Banking Landscape

May 2023

MongoDB.

Content

Management Summary

Core banking products, such as payments, checking, and custody accounts, are the backbone of our market economy. Consequently, these processes were amongst the first automated. The recent trends towards electronic and instant payments exposed the limitations of these systems as they were not built in the era of web technologies and hyperscalers. What should a modern architecture for such a system look like? And how can this migration work?

Temenos, one of the world's primary core banking software providers, embraces a cloud-first, microservices-based infrastructure that can be built with MongoDB, giving customers flexibility, while also delivering significant performance improvements. Financial institutions can embed Temenos components, like Pay-as-you-go, which delivers new functionality to their existing on-premises environments, on their own cloud deployments or through a full banking as a service experience with Temenos Transact powered by MongoDB on various cloud platforms. The results of their recent high-watermark benchmark using MongoDB are astounding.

We share with Tony Coleman, Temenos' CTO, the belief that choosing the architecture in combination with what he called "appropriate" technologies are what makes it possible to build scalable and flexible applications and application landscapes with lower cost per operation than ever before.

Based on Tony's comments we will explore architecture paradigms that support massively scalable, high-performance transactional applications such as core banking.

First, we must look at the internet giants and retailers. We get the impression that many of the technologies allowing the prior two to scale massively are not easily applicable to transactional workloads such as payments and account balances. Have you ever wondered why payment processes on most websites are much slower than selecting products and other functions? It's because, at the core, the internet giants rely on payments infrastructure built decades ago.

Next, we turn to the FinTechs and Neobanks that clearly are unencumbered by PL/1 and other mainframe technologies. These companies truly embrace technology as their competitive advantage yet only a few of them have spoken about their technology choices publicly. Even if they release some information about their infrastructure choices, their functional scope is usually much narrower than what a small regional bank would require, as they can leverage existing infrastructure and service providers.

Architecting for a better banking future

Gen Z, the mobile world, challenger banks, crypto, and DeFi in combination with regulatory changes are forcing financial institutions to change at an unprecedented rate. The design and creation of new products and services requires roll-outs measured in weeks and months rather than years; requiring an equally drastic technology development.

Banking solutions were first developed in the 1960s and usually have a life cycle of decades. Interestingly, it's not rare to see systems built in the 1980s followed by another wave of core banking systems built in the 1990s. Upgrades and changes almost exclusively focused on regulatory or technical upgrades, e.g. operating system versions, hardware upgrades, and new functionality were bolted on. These architectures were optimized under different constraints such as disk capacity and network capacity. Therefore, it is not surprising that they fail to support today's scale and pace of change.

Finance follows a path similar to that of the retail industry. Retail was built upon a static design approach with monolithic applications connected through ETL (Extract, Transform, and Load) and a robust "unloading of data". The fast move to omnichannel services required the movement from monolithic and tightly coupled architectures to component-based architectures. With this architecture, retailers can now innovate much faster and enhance their offerings quickly by deploying fit-for-purpose components into their stack. They have moved from application development to system composition. A great example is a company like Commercetools, which provides flexible and scalable components.

MACH (Microservices, API first, Cloud native, and Headless) codifies the design principles necessary for component-based architectures. Microservices are defined as small components which own functionality and data for a given business domain. They communicate through APIs that define the contracts between the different domain teams. Typical application domains consist of hundreds or even thousands of microservices. Operating such systems traditionally would not be feasible; therefore, they can only be built with cloud-native technologies. In order for Microservices to be able to be composed into applications, they have to expose all their capabilities and data through documented APIs. There is no need for user interfaces, which are captured by them being headless.

Temenos and MongoDB joined forces in 2019 to investigate the path toward componentized banking. Over the last few years, the teams have collaborated on a number of new and innovative services. The services are now deployed to multiple customers as part of the Temenos Infinity front-office offering. The multi-version tolerance and vertical scalability allow banks to deploy and upgrade components without interruption. Because these components are "in front" of the core banking platform Temenos Transact, banks can start using a componentization solution without disrupting their ability to serve existing customers.

Vertically scalable, ephemeral, and immutable services create critical requirements for the database layer. MongoDB provides a highly resilient, highly scalable data platform in a wide range of deployment scenarios — including fully on-premises, private cloud, public cloud, software-as-a-service, and even hybrid cloud. Tony Coleman, Temenos CTO, cited these capabilities as one of the key criteria for choosing MongoDB. In combination with geo-sharding, cluster-to-cluster replication, and sophisticated field-level encryption, MongoDB provides all the features for the breadth of requirements that Temenos customers have across more than 150 countries and their regulations.

Encouraged by their success with individual components, Temenos decided to adopt MongoDB to support their standardized high watermark benchmark in 2022 and again in 2023, announcing the astounding results at their annual Temenos Community Forum. To achieve these results, they used a production equivalent setup of Transact.

Selecting a database

Databases were historically a rather boring engine room discussion. While there was a myriad of discrepancies between Oracle, Microsoft, and IBM, the discussion about the right choice was often directed by platform plus personal preferences. Oracle on Solaris or AIX, DB2 for the mainframe, and Microsoft for Windows — the decision of which database to use fell along those lines. Vendors "optimized" their applications by slightly abstracting the SQL layer to a data layer or simply utilizing the smallest common denominator of functionality — to the chagrin of the database vendors who loved to push proprietary extensions.

The scalability and performance requirements in combination with the need for continuous upgrades have resulted in the requirement for the next generation of databases that are perfectly suited to exploit the ample resources and fast connectivity that are now ubiquitous. The simple decision path that software houses have been used to being replaced by a complex decision is littered with dead ends and inefficient routes.

Irrespective of personal preference, a number of criteria are fundamental to the correct path. If such criteria are not met, software vendors will have a bad case of locking to the wrong path with potentially dire consequences.

Freedom to run anywhere

When a software vendor chooses a database solution, several criteria have to be considered. Clearly, the benefits of the new technology should be significant. However, there also needs to be a way for existing customers to upgrade their software, even if they do not want to switch to the new technology. Financial service institutions have some of the most complex data secrecy and privacy rules to cope with. As a result of the complexity and diversity of rules and regulations, as well as their interpretation and risk tolerance, the preferred deployment options range from on-premises to software as a service with often specific disaster recovery and data location rules. On top of that, financial institutions have to cope with their existing infrastructure and past decisions and often are not able to move into the cloud immediately. Delivering capabilities that support always-on and integration into standard DevOps tooling becomes crucial for the vendor and the customers. Cloud-native does not imply cloud only.

Equally critical , banks have decided to work with a single preferred vendor or across multiple cloud providers. Any software vendor needs to be able to accommodate. Even for the purpose of SaaS solutions, some clients are very specific about their preferred cloud provider usage. More details can be found in our blog series. Beyond the capability to support many different deployment patterns, MongoDB allows runtime migration from one setup to another or with cluster-to-cluster replication, which can easily satisfy data locality requirements without sacrificing the scalability of cloud deployments.

MongoDB Atlas is provisioning the data service for Temenos across AWS, GCP, and Azure clouds. As the word “across” implies this includes the capability to operate one single instance across *MULTIPLE* clouds and being able to comply with regulators on one side – but more importantly with the requirement to be always on for clients. The multi-cloud capabilities including the needed security concepts, e.g. key management, are standard in MongoDB Atlas and live with thousands of customers worldwide.

When considering all these criteria, the decision process for the software vendor for the number of potential databases becomes… simpler. This is what Tony Coleman outlined at MongoDB World.

Transformation with a developer data platform

Data today no longer exists in the dark backend of solutions. They are recognized as an asset as well. The term data-as-a-product summarizes this change in attitude very succinctly. The acceleration of banking services and real-time payments are one of many forcing functions. Applications such as online fraud detection require a rethink of the management and access to data in a holistic way. Contextual data needs to be available for online processing but also be utilized to train and calibrate the models in large batch-like constructs. This creates the need for a holistic approach to everything data.

The rigid, normalized schema relational databases of any kind are not able to handle this diversity.

Over the last few years, JSON has emerged as the dominating standard for modern data. JSON has delivered on the promises of XML as being an extensible, easy-to-use and polyglot data format. Being easy to learn and work with is not its only strength. In fact, its support for changes in usage patterns of data models is equally important. Data is truly at the center of modern application development. User interface technologies have moved from statically linked observer patterns to the modern reactive style where user interaction is driven exclusively by data. The constraints imposed by disk and memory have long dropped aside in all but the most extreme domains. Schemas can be made backward compatible by including redundant data and or projection criteria on the query. These mechanisms have replaced the tight coupling and often complicated storage algorithms of the client/server architectures. Today, developer productivity has become the limiting factor that all organizations are suffering from. Industry research shows that financial service companies with their long tradition of building in-house solutions are suffering from ever-increasing app dev costs. JSON dominance as a data language is evidenced by the definition of JSON data models through standardization organizations such as SWIFT, ISO20022, and FIX.

> “We estimate that banks on average convert just five to ten cents of every dollar of tech spend into additional business value.” McKinsey

MongoDB is the leader in JSON-based data platforms. Not only does it deliver a high-performance, high-reliability, horizontally scalable document database, but it also increases developer productivity by providing a data platform that combines a broad

range of indexing options, advanced and clean management functions. The prior of which is supported by automated advisors, as well as a powerful aggregation framework that can easily be extended with UDFs and applies to adhoc queries as much as materialized views. Atlas expands the capabilities of the core banking platform even further by enabling full-text search capabilities, mobile connectivity with MongoDB Realm, and data retention and analytics with MongoDB Atlas Federation and MongoDB Atlas Data Lake. Additionally Atlas supersedes in availability and reliability any typical on-premises legacy systems through its sophisticated replication and backup strategies.

Bringing all these capabilities into one platform reduces the cognitive load on the development team as they are freed up to focus on driving value from data and functionality. Teams will not have to learn different interface paradigms and query languages – these consistency guarantees are only part of the effort saved. Companies always want to use the best technology for each job but are hamstrung by the operational efforts and the complexity from the different paradigms.

Even the most sophisticated data solutions struggle to support transactional workloads and more analytical loads. MongoDB brings all these capabilities to the developer's fingertips under a single technology framework. Working through a single consistent interface, multiple different types of workloads can be run using workload isolation if necessary. Temenos chose to isolate these different workloads through the CQRS pattern.

The amount of operational effort to run diverse ecosystems is often not taken into account in technology decisions. Each platform will have different monitoring, triggers, runtime characteristics, and integrations into the respective clients monitoring solution. The operational problem can be overcome with significant resources. Harder to overcome are the security risks that arise from the complexity of running on multiple technologies even if just as a result of incomplete understanding of the technology in use.

Beyond the basics

Besides these basic functions of traditional database systems, a true developer data platform needs to provision services for the data life cycle and more advanced functions to avoid redundant development. A great example is the implementation of a mobile payment function in a mobile phone app. Nothing prevents a developer from developing every single component of the data replication process themselves, but a good developer data platform provisions these mobile applications for the developer.

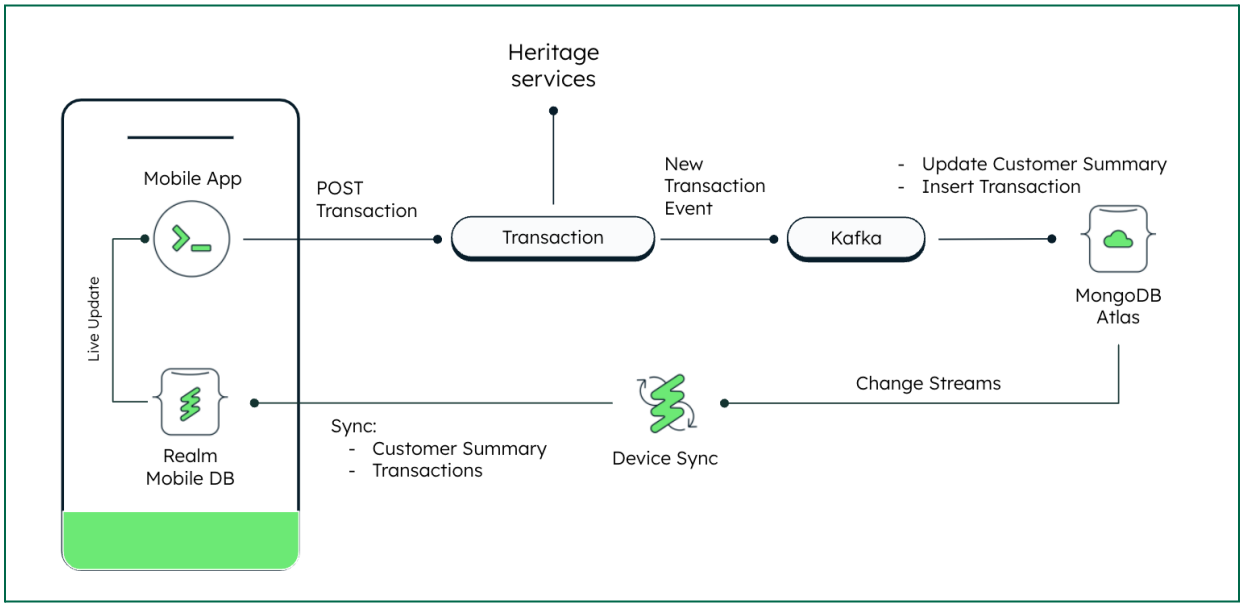

Figure 1: Payment transaction initiated on the mobile device

In the case above, a new payment transaction is initiated on the mobile device. The mobile device creates a document that reflects the "Saga pattern" and forwards the payment command to the legacy backend. Once the back-end service has enacted the payment command, the resulting changes to entities in the database are transmitted back to the data platform supporting the front-end application, which can update the status on the saga and inform the customer of the successful action. This pattern allows customers to initiate payments even if the back end is offline or even without connectivity to the back end at all. What used to be very complicated networking issues for making synchronous calls into the back-end are taken over by the device synchronization mechanisms of MongoDB Atlas.

Performance at scale

Increasing developer performance sounds great; however, ultimately the real-world performance and scalability of the platform is crucial. Things were easy when customers had to wait for their monthly statement or walk into a branch to get an update on their account balances. In our always-on, low-latency world, the strain on the data infrastructure is much more significant. In combination with millions of transactions, customer accounts and the must-have transaction reliability that modern banking, and

payment applications require, building an effective data infrastructure sounds like a daunting task.

While document databases are truly cloud-native, they often were thought as insufficient to deliver strong consistency guarantees that are required for the seamless working of banking applications. Interestingly, most of the strong consistency requirements that have been the staple of relational databases are getting eliminated by the adoption of a document model. The atomic update of a single document can replace the coordinated insert and update operations required to keep the various relations in sync.

Benchmarking is always subjective, and many benchmarks are performed in artificial clean room environments. While clean environments help with diagnosing issues and performance-tuning a software, they often fall short of being good proxies for real-world performance. Temenos decided to run a much less clean and thereby likely more realistic benchmark on their platform in frequent intervals.

Tony Coleman, Temenos CTO, gives us a bit of insight into how they achieved this level of performance.

In contrast to the retail-centric benchmark last year, the approach this time was to test broader functionality and include more diverse business areas – all while increasing the transaction volume by 50%.

The benchmark scenario simulated a client with 50 million retail customers, 100 million accounts and a Banking-as-a-Service (BaaS) offering for 10 brands and 50 million embedded finance customers on a single cloud instance.

In the test, Temenos Banking Cloud processed 200 million embedded finance loans and 100 million retail accounts at a record-breaking 150,000 transactions per second. In doing so, Temenos proved its robust and scalable platform can support banks' business models for growth through BaaS or distributing their products themselves. The benchmark included not just core transaction processing, but a composed solution combining payments, financial crime mitigation (FCM), a data hub, and digital channels.

"No other banking technology vendor comes close to the performance and scalability of Temenos Banking Cloud. We consistently invest more in cloud technologies and have more banks live with core banking in the cloud than any of our peers. With global non-cash transaction volumes skyrocketing in response to fast-emerging trends like BaaS, banks need a platform that allows them to elastically scale based on business demand, provide composable capabilities on-demand at a low cost, while reducing their environmental impact. This benchmark with Microsoft and MongoDB proves the capability of Temenos'

platform to power the world's biggest banks and their BaaS offerings with hundreds of millions of customers, efficiently and sustainably in the cloud."

Tony Coleman, Chief Technology Officer, Temenos

This solution landscape reflects an environment where everyone on the planet runs two banking transactions a day on a single bank. This throughput should cater to any Tier 1 banking deployment, in size and performance, and cover any future growth plans that they have.

Below are the transaction details that comprise the actual benchmark mix. As mentioned above it is a broad mix of different functionalities behaving like a retail bank and a fintech institute, which provides multiple product brands, e.g. cards for different retails.

Queries	Target	Mix %	# transactions	% transactions	Response Time
Get Loan Details	25650	17.10%	25860	17.23%	18ms
Get Balance	49500	33%	50975	33.90%	7ms
Get Loan Transactions	29450	19.63%	29630	19.74%	11ms
Get Transactions LIst	15000	10.00%	11750	7.83%	7ms
Get Repayment Schedule	11400	7.60%	15490	10.32%	11ms
Login	1500	1.00%	830	0.55%	
Booking	10640	7.09%	6900	4.60%	146ms
Reservation	6125	4.08%	8200	5.46%	37ms
Loan Creation	90	0.06%	105	0.07%	875ms
Repayments	90	0.06%	275	0.18%	200ms
Payments	550	0.37%	140	0.09%	25ms
Customer Create	5	0.005%	25	0.17%	50ms
TOTAL	**150,000**	**100%**	**150,060**	**100%**	

Besides the sheer performance of the benchmark, the ESG footprint of the overall landscape shrunk again versus last year's configuration as the MongoDB Atlas environment was the sole database and no secondary systems were required.

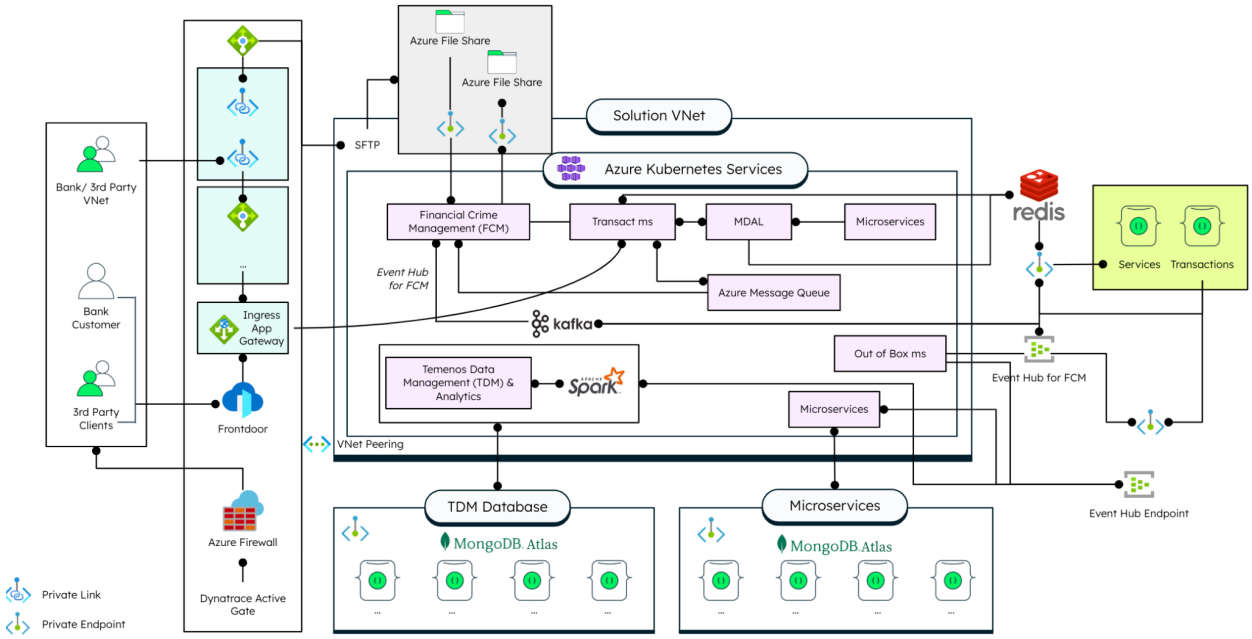

Figure 2: Temenos Transact optimized with MongoDB

The JSON advantage

Temenos made significant engineering efforts to decapsulate the data layer, which was previously stored as PIC, and make JSON formatted data available to their user community. MongoDB was designed from its inception to be a database focused on delivering a great development experience. JSON's ubiquity made it the obvious choice for representing data structures in MongoDB's document data model. Below in figure 3, you can see how Temenos Transact stores data vs Oracle or MSSQL vs MongoDB. Temenos and MongoDB have an aligned data store – Temenos Transact application code operates on documents (JSON) and MongoDB stores documents in JSON in one place, making it the perfect partnership.

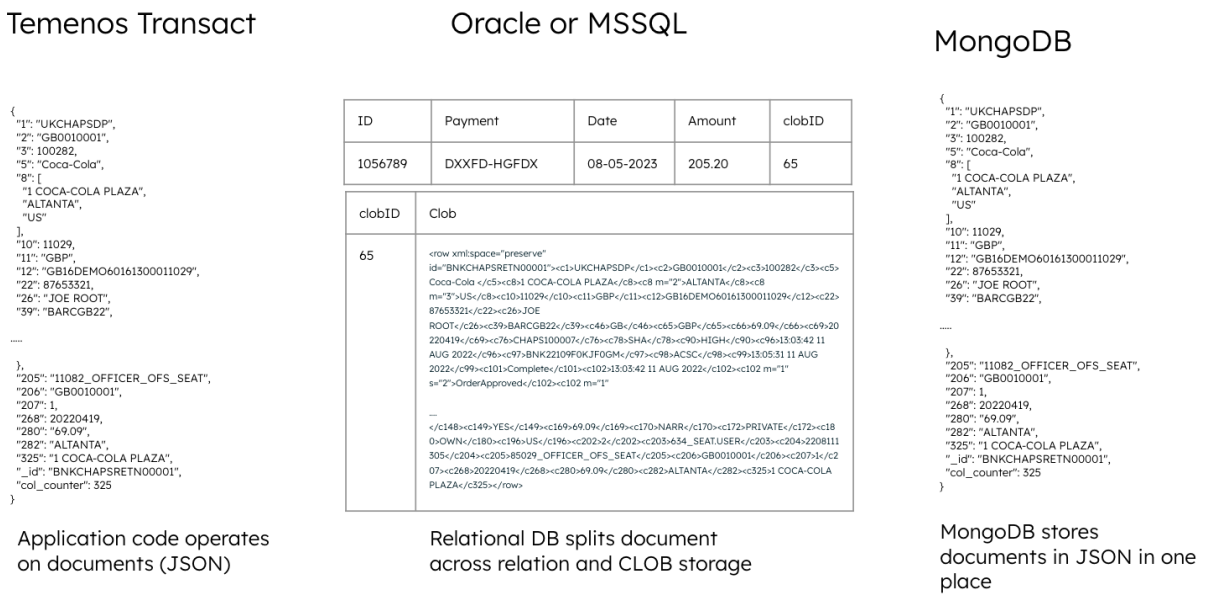

Figure 3: Temenos Transact data store vs Oracle or MSSQL vs MongoDB

MongoDB enables the user community through its concept of additional nodes in the replica set to align further secondary applications integrated into the same database without interrupting and disturbing the transactional workload of Temenos Transact. The regular occurring challenge with legacy relational database management systems (RDBMS) where secondary applications suddenly have unexpected consequences to the primary application is a problem of the past with MongoDB.

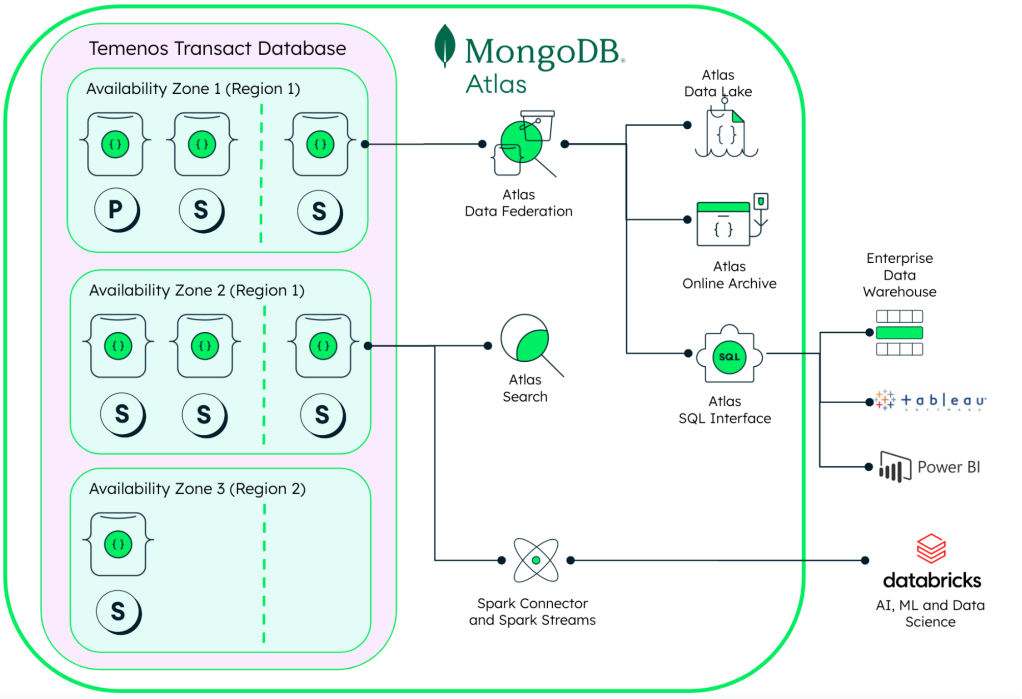

Figure 4: Workload Isolation with MongoDB

MongoDB Atlas will operate in most cases in three availability zones, where two zones are located in the same region for pure availability and a single node is located in a remote region for disaster recovery. This environment provides the often required RPO/RTO "0" while delivering unprecedented performance. Two nodes in each of the first availability zones provision the transactional replica set and ensure the consistency and operation of the Temenos Transact application. In each availability zone, a third isolated workload node is co-located with the same data set as the other two nodes but is excluded from the transactional processing. These isolated workload nodes provide capacity for additional functionalities. In the example above, one node provides access to the MongoDB Atlas Federation and a second node provides the interface for MongoDB Atlas Search. As the nodes store data in near real-time - replication is measured in sub milliseconds as they are in the same availability zone - this allows exciting new capabilities like real-time large language model (LLM), e.g. ChatGPT, or machine learning connecting to a Databricks lake house. The design is discussed in more detail in this article.

The below diagram shows a typical configuration for such a cluster setup in the European market for Microsoft Azure: one availability zone in Zurich, one availability zone in Geneva, and an additional node out of both in Ireland. Additionally, we configured isolated workloads in Zurich and Geneva. MongoDB Atlas allows the creation of such a cluster within seconds, configured to the specific requirements of the solution deployed.

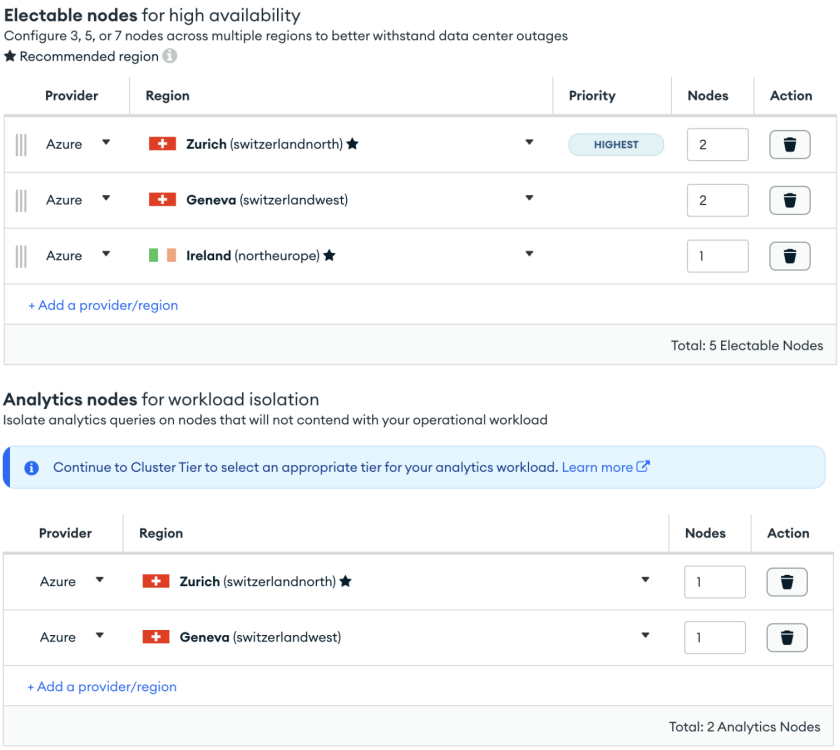

Figure 5: Typical configuration for a cluster setup for the European market for Microsoft Azure

Should the need arise, MongoDB can have up to 50 nodes in a single replica set so for each additional isolated workload, one or more nodes can be made available when and where needed. Even at locations beyond the initial three chosen! Below is a snapshot out of the integration of MongoDB Atlas and the Temenos runtime - seamless single screen to monitor the Temenos backend throughput.

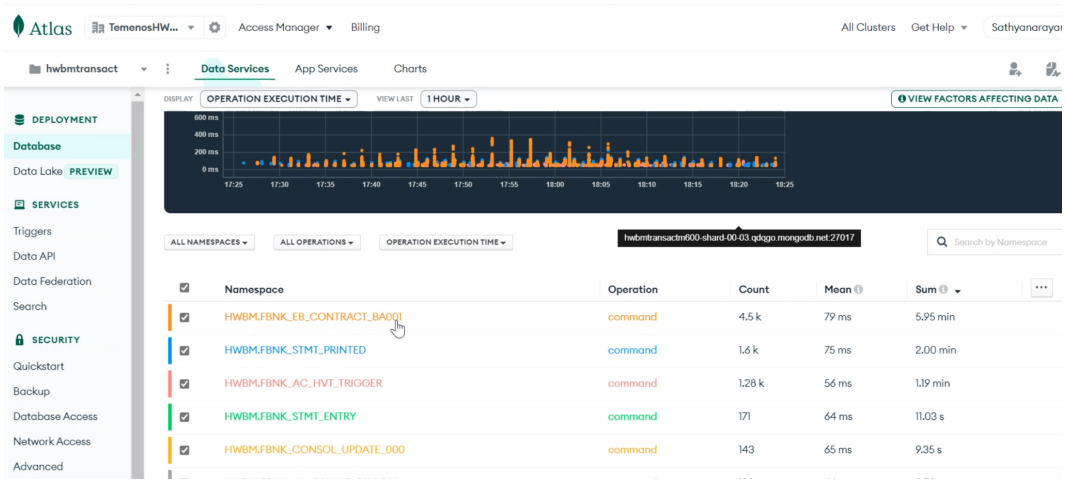

Figure 6: Integration of MongoDB Atlas and the Temenos runtime

For this benchmark the use of a MongoDB Atlas cluster M600 was utilized which was oversized based on the CPU utilization of 20-60% depending on the node type. Looking backward a smaller MongoDB Atlas M200 would have been easily sufficient. Nonetheless MongoDB Atlas delivered the needed database performance with one third of the resources of last year's result, but delivering 50% more throughput. Additionally MongoDB Atlas performed two times faster in throughput per transaction (measured in milliseconds).

Conclusion

Signed, sealed, and delivered. This benchmark should give clients peace of mind that MongoDB is more than fit for purpose to underpin core banking services just as Temenos have done it with Transact. While thousands of banks rely on MongoDB for many parts of their operations ranging from login management and online banking to risk management and treasury, Temenos' adoption of MongoDB is a milestone. It shows that there is significant value in moving from a legacy database technology to the innovative MongoDB developer data platform. This includes better performance, faster innovation, reduced technical debt along the way, and a cleaner architecture landscape for financial institutions, their software vendors, and service providers.

If you would like to learn more about how you can use MongoDB to move towards a composable system, architecting for real-time adaptability, scalability, and resilience, take a look at the below resources:

- Componentized core banking built upon MongoDB
- Tony Coleman, CTO at Temenos and Boris Bialek, Global Head, Industry Solutions at MongoDB discuss the partnership at MongoDB World 2022
- Remodel your core banking systems with MongoDB

About MongoDB

MongoDB empowers innovators to unleash the power of software and data. Whether deployed in the cloud or on-premises, organizations use MongoDB for trading platforms, global payment data stores, digital end-to-end loan origination and servicing solutions, general ledger system of record, regulatory risk, treasury and many other back-office processes. At the core of our developer data platform is the most advanced cloud database service on the market, MongoDB Atlas, which can run in any cloud, or even across multiple clouds to get the best from each provider with no lock-in.

To learn more about MongoDB, visit MongoDB.com

About the authors

Joerg Schmuecker, Director Financial Services Industry leads the global FSI practice at MongoDB. He led the benchmark efforts and the Temenos integration from the MongoDB side and is a veteran of the banking industry. Joerg joined MongoDB in 2022 after a 16 year career at Credit Suisse in multiple roles and locations and last as the head of Macro IT.

Boris Bialek, Managing Director, Industry Solutions leads the industry solution practices at MongoDB and focuses on modernization and true innovation of FSI solutions. His experiences range from core banking, payments and cards to trading and risk & treasury. He is an industry expert in data technologies and recognized speaker and author. Before joining MongoDB, he worked for a lifetime at FIS, IBM, Dell and Compaq Computers. He obtained an MS degree from the Karlsruhe Institute of Technology.

Resources

For more information, please visit mongodb.com or contact us at sales@mongodb.com.
Case Studies (mongodb.com/customers)
Presentations (mongodb.com/presentations)
Free Online Training (university.mongodb.com)
Webinars and Events (mongodb.com/events)
Documentation (docs.mongodb.com)
MongoDB Atlas database as a service for MongoDB (mongodb.com/cloud)
MongoDB Enterprise Download (mongodb.com/download)MongoDB Realm (mongodb.com/realm)

Legal Notice

This document includes certain "forward-looking statements" within the meaning of Section 27A of the Securities Act of 1933, as amended, or the Securities Act, and Section 21E of the Securities Exchange Act of 1934, as amended, including statements concerning our financial guidance for the first fiscal quarter and full year fiscal 2021; the anticipated impact of the coronavirus disease (COVID-19) outbreak on our future results of operations, our future growth and the potential of MongoDB Atlas; and our ability to transform the global database industry and to capitalize on our market opportunity. These forward-looking statements include, but are not limited to, plans, objectives, expectations and intentions and other statements contained in this press release that are not historical facts and statements identified by words such as "anticipate," "believe," "continue," "could," "estimate," "expect," "intend," "may," "plan," "project," "will," "would" or the negative or plural of these words or similar expressions or variations. These forward-looking statements reflect our current views about our plans, intentions, expectations, strategies and prospects, which are based on the information currently available to us and on assumptions we have made. Although we believe that our plans, intentions, expectations, strategies and prospects as reflected in or suggested by those forward-looking statements are reasonable, we can give no assurance that the plans, intentions, expectations or strategies will be attained or achieved. Furthermore, actual results may differ materially from those described in the forward-looking statements and are subject to a variety of assumptions, uncertainties, risks and factors that are beyond our control including, without limitation: our limited operating history; our history of losses; failure of our database platform to satisfy customer demands; the effects of increased competition; our investments in new products and our ability to introduce new features, services or enhancements; our ability to effectively expand our sales and marketing organization; our ability to continue to build and maintain credibility with the developer community; our ability to add new customers or increase sales to our existing customers; our ability to maintain, protect, enforce and enhance our intellectual property; the growth and expansion of the market for database products and our ability to penetrate that market; our ability to integrate acquired businesses and technologies successfully or achieve the expected benefits of such acquisitions; our ability to maintain the security of our software and adequately address privacy concerns; our ability to manage our growth effectively and successfully recruit and retain additional highly-qualified personnel; the price volatility of our common stock; the financial impacts of the coronavirus disease (COVID-19) outbreak on our customers, our potential customers, the global financial markets and our business and future results of operations; the impact that the precautions we have taken in our business relative to the coronavirus disease (COVID-19) outbreak may have on our business and those risks detailed from time-to-time under the caption "Risk Factors" and elsewhere in our Securities and Exchange Commission ("SEC") filings and reports, including our Quarterly Report on Form 10-Q filed on December 10, 2019, as well as future filings and reports by us. Except as required by law, we undertake no duty or obligation to update any forward-looking statements contained in this release as a result of new information, future events, changes in expectations or otherwise.