White Paper

AEROSPIKE

Modernizing legacy mainframe applications with Aerospike®

Contents

Introduction and highlights

In this age of rapid technology transformation, new applications generate unprecedented data volumes and put pressure on legacy enterprise data infrastructures, especially mainframe systems. Once the preferred platform for business processes, mainframes struggle to meet the demands of contemporary, internet-scale applications such as customer-facing web applications, fraud detection systems, and real-time reporting. Mainframes have limited flexibility, high operational overhead and costs, are difficult to maintain, and require long lead times for developing and deploying new applications. These are among the reasons many firms try to offload work from their mainframes.

Aerospike is a distributed, multi-model database that helps firms power the most demanding real-time, internet-scale applications. The key pillars that have drawn organizations to offload legacy workloads and modernize with Aerospike include:

Risk-averse modernization - Aerospike's phased approach does not try to rewrite legacy code, thus minimizing the risk of modernization, while offering tangible benefits each step along the way.

Reduced complexity - Aerospike's scalable real-time data platform enables firms to use the same data platform across these mainframe modernization phases, eliminating extra re-platforming, and thus simplifying and speeding the process.

Lowest TCO - Fueled by its patented Hybrid Memory Architecture, Aerospike provides unmatched performance at a total cost of ownership (TCO) that's typically 80% lower than that of other modern databases. The cost savings are even more profound relative to mainframe platform costs.

Required database platform capabilities

These are just some of the requirements for a modern database to offload mainframe-based data infrastructure:

- Deliver fast, predictable data access to real-time applications.
- Scale easily and efficiently without high cost or operational overhead.
- Deliver high availability with zero downtime.
- Provide effective security measures.
- Integrate data from various sources and support popular streaming pipelines.
- Drastically reduce total cost of ownership (TCO) when compared with mainframe costs.

The mainframe modernization process

Replacing a fully functioning mainframe system is tough: such systems are inherently complex, support critical business processes and compliance initiatives, and often encapsulate industry-specific intellectual property developed over decades. For example, a brokerage firm might have tax base calculations built into its mainframe application, or a bank could have credit card processing functions built into its application. A "rip and replace" approach to mainframe modernization is seldom practical. That's why Aerospike advocates gradual augmentation of such systems using a phased approach. As Gartner Group notes,

> ***"Rip and replace is in many cases too costly, risky and time-consuming, and has a high impact on the business. We advise organizations to instead take an iterative approach: continuous application modernization."***

Gartner Group, "Use Continuous Modernization to Build Digital Platforms From Legacy Applications"

A phased approach to modernization

To understand Aerospike's mainframe modernization process, let's first review a common mainframe implementation. As shown in Figure 1, IBM DB2 serves as the system of record for legacy applications, perhaps with a file system, a distributed database, and a data warehouse also feeding data to applications. In this view, the legacy infrastructure is also the back end for modern applications.

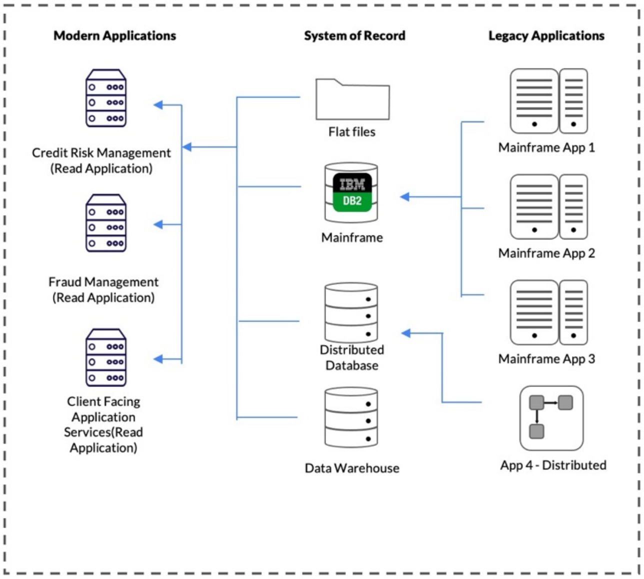

Fig. 1: Typical mainframe implementation

Figure 2 summarizes Aerospike's phased approach to modernization – an approach that's been proven at many large enterprises. We'll look at a few in the second part of this paper.

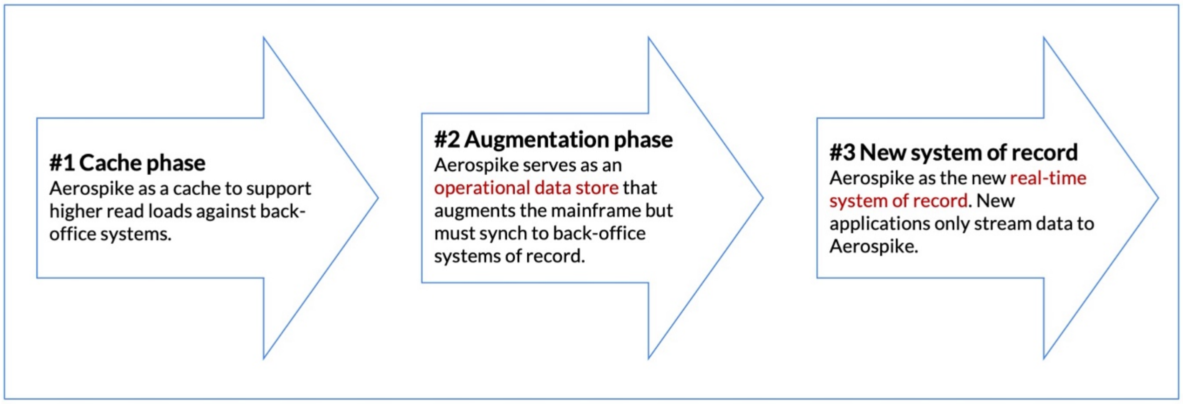
#1 Cache phase
Aerospike as a cache to support higher read loads against back-office systems.
#2 Augmentation phase
Aerospike serves as an operational data store that augments the mainframe but must synch to back-office systems of record.
#3 New system of record
Aerospike as the new real-time system of record. New applications only stream data to Aerospike.

Figure 2 - Aerospike's mainframe modernization process

The implementation of the modernization process can vary from company to company, but the methodology is the same. Let's walk through the three phases that comprise the typical pattern.

Cache Phase - Aerospike enhances reads

Requirements/issues:

Back-end systems are often pushed beyond their limits trying to meet the demands for access and visibility to corporate data. A legacy of dozens of systems and applications struggles in the face of millions of end customers. How can firms modernize their infrastructures to handle high volumes of reads from customer-facing and other applications and match or improve the customer experience? For many use cases, the volume of database reads far exceeds the write loads, and so that is a first place to focus. Reducing mainframe read loads can also meaningfully lower a company's MIPS (Millions of Instructions per Second) and their large associated cost.

Implementation:

Each time an application accesses the database, the data is pulled across a complex set of integrations across those dozens of systems that comprise the legacy infrastructure. As demands grow, so too does latency, i.e. delay for the application. If just one of the systems is unavailable or slowed by other requests, the application performance suffers.

To address this, organizations can pull all that data together in a database that serves as a real-time cache. The dozens of source systems are ingested and joined in a recurring batch process, and then changes to the source data are updated in the cache via a messaging queue.

Figure 3 illustrates this. Aerospike sits between the legacy systems on the right and the modern, client-facing applications on the left. The mainframe pushes the data onto IBM MQ Series and

Aerospike Connect for JMS pulls that data stream and pushes it into Aerospike. Connect for JMS is able to transform the mainframe-structured data into the proper format for Aerospike and the applications.

Aerospike serves as a low-latency, high-scale cache here. All of the legacy applications now access the Aerospike database, which is optimized for speed and availability. Similarly, modern applications would read from Aerospike.

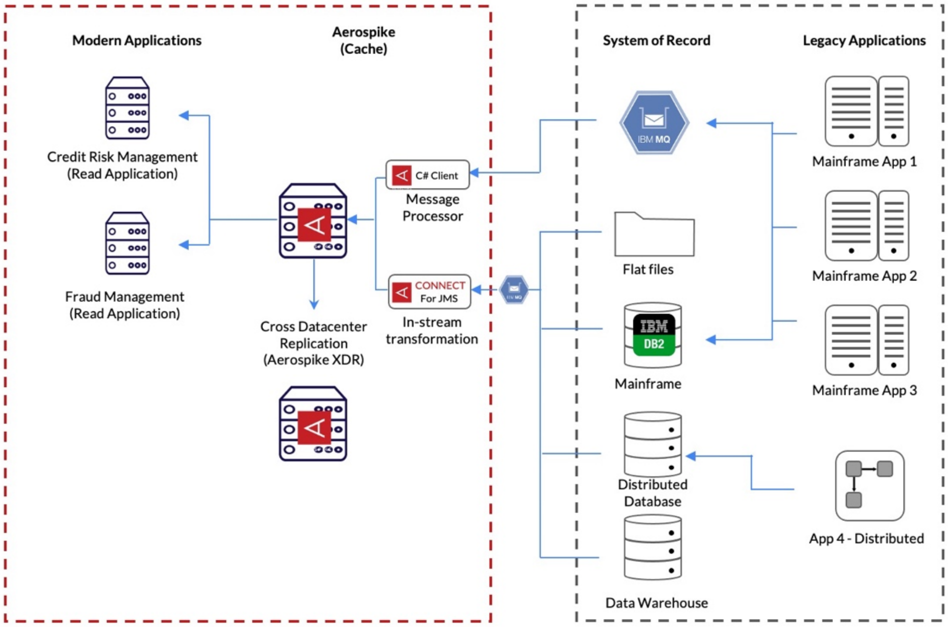

Figure 3 – Aerospike deployed as a cache

Adding Aerospike as a cache is a practical first phase to improve existing read workloads. Additionally, Aerospike provides a better path to scaling, enabling modern applications to deliver new capabilities that were previously thought impossible.

Benefits:

- **Improved data availability.** Aerospike holds data for reads, providing applications with fast access to data, even if source systems are unavailable.
- **Improved scalability.** The application layer can be scaled quickly and cost-efficiently by adding Aerospike nodes – an operation that requires no downtime and little operator involvement.
- **Improved application availability.** Modern applications are decoupled from the legacy system of record data, insulating them from legacy maintenance issues. Instead, applications run directly against Aerospike, a platform that provides 99.999% (five nines) uptime.
- **Improved data access speeds.** Aerospike delivers reads and other operations often in less than a millisecond.

- **Support for new read-centric applications.** Aerospike powers real-time applications once considered impractical or too costly to implement, such as sophisticated credit risk monitoring systems.
- **Increased maintainability.** Aerospike is easy to maintain, in contrast to the notoriously difficult-to-maintain mainframe that has typically grown more and more complex.
- **Reduced costs.** High MIPS are often a leading motivator to reduce mainframe workloads.

Augmentation Phase - Aerospike serves as an operational data store

Requirements/issues:

Phase 1 - the caching phase - prioritizes the availability and speed of reads for modern applications (e.g. an end customer checking their balance). In that phase, writes are made to the legacy infrastructure which serves as a system of record. They only stream into Aerospike after this step. This means modern applications would experience some delay in seeing updated data in Phase 1, the caching phase. Consider a brokerage where an end customer initiates a stock sale. Their account balance view may not immediately incorporate the recent sale.

This often leads organizations to a second modernization phase where the modern database handles both reads and writes. While many organizations see modern NoSQL databases as obvious solutions for cache use cases, they may be cautious about changing writes to the legacy system of record. Countless hard-coded legacy applications have been built to read from these systems. Therefore, this second phase must add writes to the modern database, while still providing asynchronous updates to the legacy system of record.

Implementation:

In the prior diagram, all the arrows were pointed from right to left, as the priority was preparing and serving data for modern applications to read it in real time. In this second phase, and in Figure 4, in the lower left, we add a modern application that also generates writes. Its writes follow two paths in the diagram.

The first is via IBM MQ and updates the legacy system of record. This conceptually existed in the prior phase and graphic, except we weren't looking at applications that generate writes. As in the cache phase, Aerospike Connect for JMS can transform this data, this time into the proper mainframe format.

The second path is directly to the Aerospike database. This is a real-time update, meaning it eliminates the delay that came from waiting for messages from the legacy infrastructure. The modern applications now have real-time data. For the retail brokerage example, the customer's portfolio now immediately reflects their latest trades.

Focusing on the left half of the diagram in the red box, it's clear that the modern database (Aerospike) now serves as an operational data store, serving reads and incorporating writes in real time. The legacy infrastructure and system of record still receive updates both from the modern application as well as legacy applications.

The other change in this second phase is it can further cut legacy/mainframe MIPS and the associated costs. The write workloads tend to be "spiky," often concentrated at particular times of the day. Organizations typically provision their mainframes to meet these peak loads. In the Phase 2 architecture, they can throttle the message queue to spread those writes over time. The modern, real-time applications aren't delayed by this queueing because the data is available in Aerospike. While the legacy system of record remains, this is a huge step in mainframe offload from a workload and financial perspective.

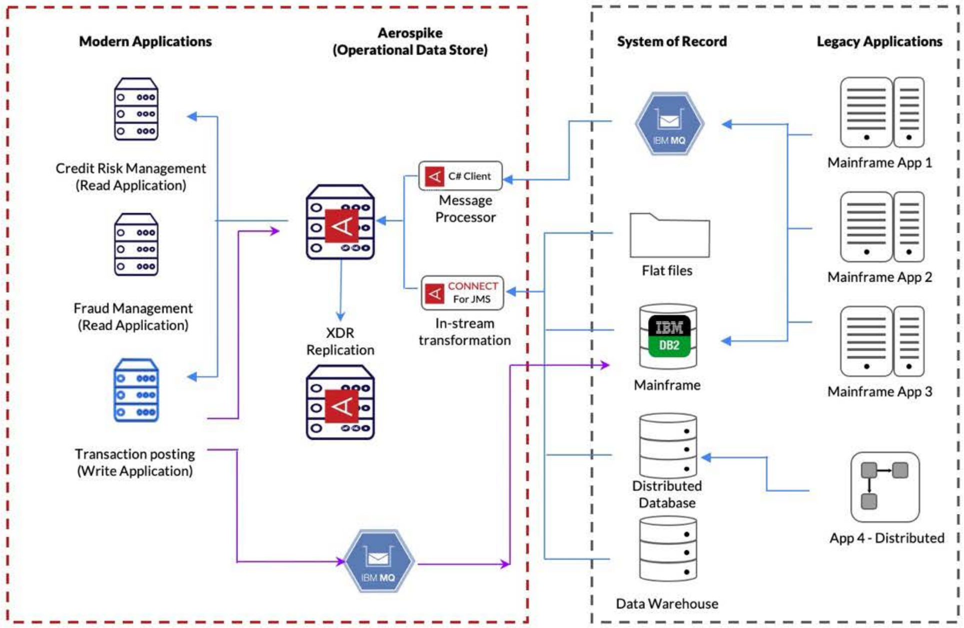

Figure 4 – Aerospike as an operational data store

Benefits:

- **Optimal support for new and legacy transactional applications.** The Aerospike operational data store is updated first, servicing the real-time needs of new transactional applications in a cost-effective, highly performant manner. The back-office system of record is updated asynchronously to support existing applications with minimal impact.
- **Improved availability.** When back-office systems are unavailable, online business requiring read/write data access is not affected.
- **Improved data scalability.** Aerospike's distributed nature and horizontal scalability allow the overall infrastructure and database to grow seamlessly.
- **Application flexibility and scale**. A modern microservices architecture and tools like Kubernetes make scaling the application simpler and easier to maintain.

- **Mainframe workload reduction.** Customer-facing transactions (writes) are handled by new applications deployed on Aerospike, reducing stress on the mainframe.

New SoR Phase – Aerospike serves as the system of record

Requirements/issues:

As organizations successfully navigate these first two phases, a greater share of their applications are modern applications. That is, they depend less on the legacy infrastructure, data, and system of record. The old infrastructure may live on for years to support legacy applications. It may be too costly or risky to replicate or replace applications that are written in COBOL or otherwise hard-coded and cryptic.

However, for modern applications, the modern database (Aerospike) can serve as a new system of record.

Phase 2 showed that applications have faster access to current data when the data doesn't need to go through the legacy infrastructure. Phase 2, though, had a refresh step where writes from modern applications eventually were made to the legacy system of record.

In the third phase of modernization, organizations look to further offload legacy/mainframe workloads and move to a modern system of record. In order to be a system of record, the new database needs the enterprise features that organizations expect from their legacy systems. It requires real-time backup and recovery, high availability, strong consistency, enterprise security, and more.

Implementation:

In Figure 5, we introduce an additional application that reads and writes only to and from the Aerospike database. In this example, Performance Reporting is the sample modern application, marked in green.

Unlike the flow from the phase 2 example, in purple, it does not stream updates into the legacy infrastructure. This continues to give the performance advantages of the prior phases and unifies the cache, operational data store, and system of record in a single platform. This further offloads the mainframe, significantly cutting MIPS and costs to the organization.

A modern, distributed database like Aerospike is built to scale, and phase 3 fully takes advantage of it. This lays the groundwork for new applications to be added rapidly and provide real-time performance at scale.

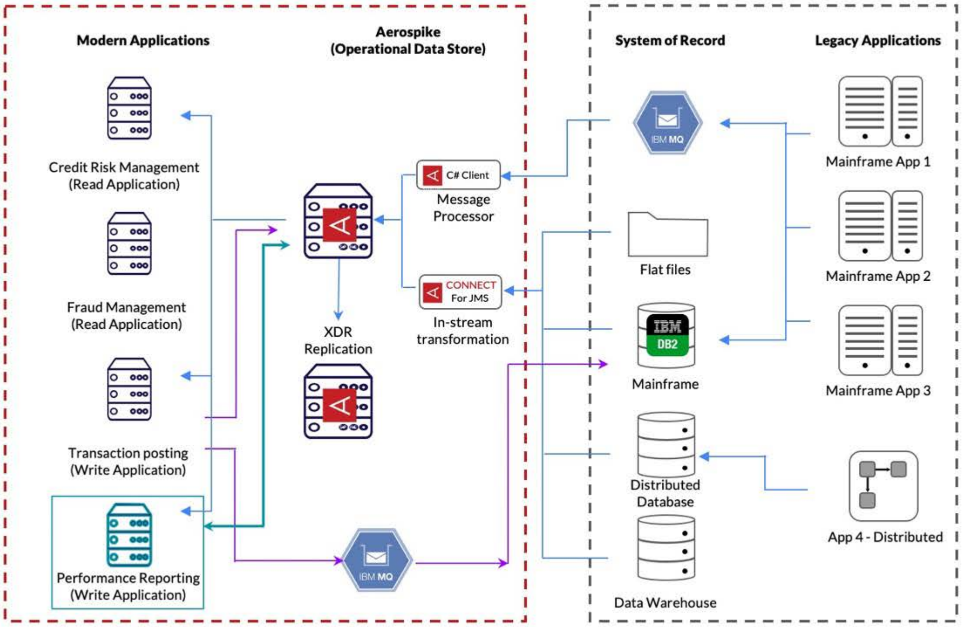

Figure 5 – Aerospike as a new System of Record

Benefits:

- **Exceptional scalability.** This configuration can support data sets ranging from gigabytes to petabytes.
- **Exceptional performance at scale.** The new system of record provides real-time data access with high availability and the potential to scale over time.
- **New applications.** New real-time system of record applications can be developed to support the non-linearity of the web.

Customer case studies

Aerospike has observed this three-phase pattern to be an effective way for organizations to modernize their applications and offload some of their legacy costs and infrastructure. That said, the cobbled-together nature of legacy infrastructure is why each organization has variations in its journey.

Aerospike is uniquely able to handle all three of these phases without any replatforming. Some databases will thrive as a cache. Others will work well at 1-2 terabytes but break down at scale.

We have customers who are simultaneously running all of these types of workloads - cache, operational data store, and system of record. Some may adopt them all at once or in a different sequence. In fact, we've seen customers deploy Aerospike for write workloads before using them for read workloads. It all depends on their specific requirements. In the next section, we'll look at three specific customer examples.

DBS Bank

DBS Bank is the largest bank in Singapore, with around 30 million customers across the Asia Pacific region. Its roots go back to the 1960s, and it has a long history of running its applications on mainframes. As it modernizes, the bank strives to improve the capabilities of its applications while reducing costs.

DBS has deployed and expanded use cases with Aerospike, and it has followed many elements of the three-phased approach. One of its big priorities has been to reduce overall mainframe MIPS (Millions of Instructions per Second) and the associated expenses.

Phase 1 - Reads

With this in mind, DBS Bank deployed Aerospike with an initial focus on reads. "The more read loads you can take out with a scalable database like Aerospike, the more you will save MIPS. It's always more reads than writes." See DBS Fireside Chat: Modernising finserv applications for real-time decisioning to hear firsthand.

Phase 2 - Reads and writes with mainframe sync

As Aerospike successfully handled initial use cases, the bank deployed Aerospike more widely. Some use cases are not transactional, separate from the mainframe workloads. For these workloads that don't require strong consistency, they use the Aerospike Availability and Partition tolerant (AP) mode. This phase was akin to the operational data store paradigm. Modern applications would read and write to Aerospike and the legacy mainframe endured, getting updates from Aerospike. Aerospike Connectors including JMS deliver real-time updates between the Aerospike database and the DBS existing legacy system.

Phase 3 - Transactions

DBS continues to expand its Aerospike usage. As described above, the bank prioritizes availability (AP mode) for certain workloads in the operational data store. Now, though, they use the Strong Consistency (SC) mode for many workloads and use Aerospike as a system of record. "Places where we need the primary source of truth to be Aerospike cannot be availability-based. We need it to be consistency-based. We get both of these in one database only." As they've moved through the phases towards using Aerospike as a system of record, they've benefited from Aerospike's enterprise features and reliability. Aerospike's shared-nothing architecture has helped them simplify database management and scale.

As DBS continues to supplement and offload legacy infrastructure and applications, it has embraced an architecture of microservices and real-time streaming. The three-phase pattern is all about balancing the need to onboard modern applications while maintaining the legacy ones.

As shown in Figures 6 and 7, DBS has built a global data model with a sophisticated ingestion engine that ties together its streaming and batch workloads. Aerospike's connectors to Apache Kafka and Apache Spark have made it a seamless part of this evolution. The JMS connector also facilitates co-existence with the legacy mainframe.

See [DBS Bank: Building a Real-time Ingestion Platform using Kafka and Aerospike]() to learn more.

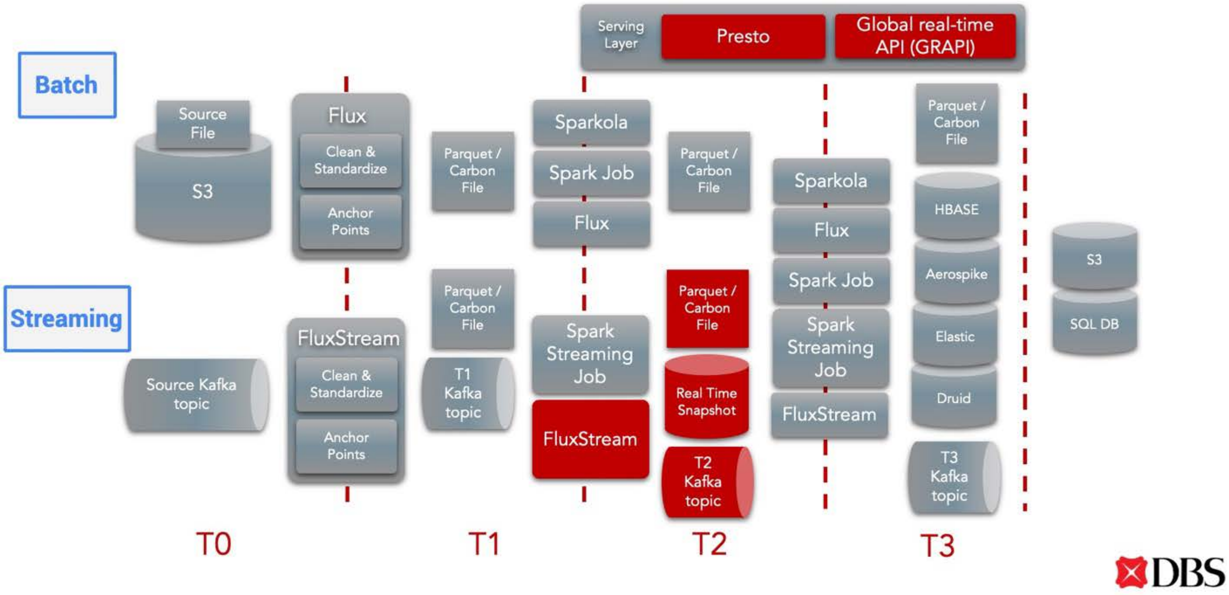

Figure 6 - DBS Bank's modern architecture for batch and streaming

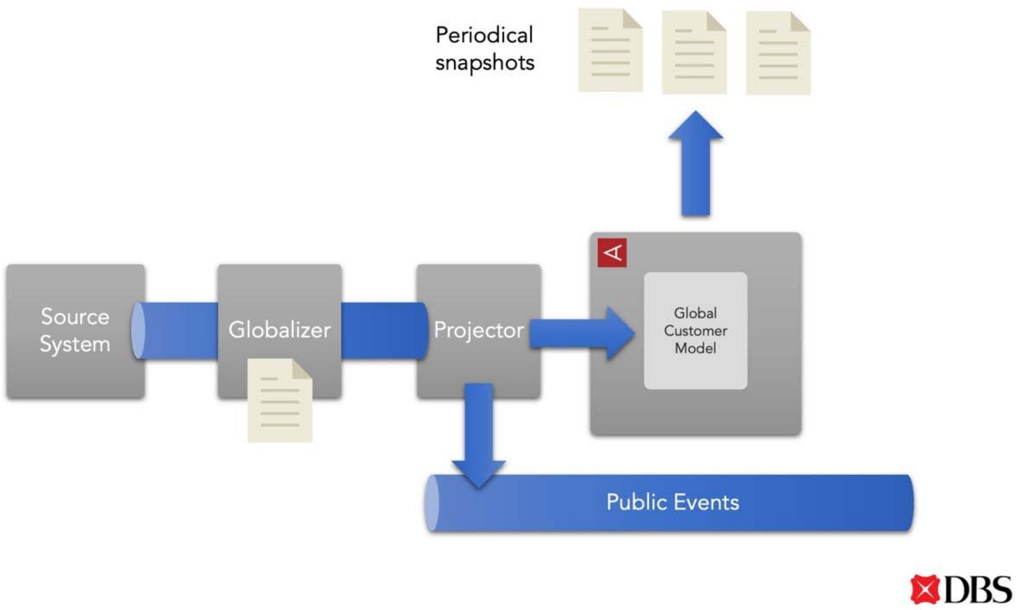

Figure 7 - DBS Bank pairing real-time data in Aerospike with periodic snapshots to files

Requirements/technology challenges:

- Replace the slower batch-oriented platform with a real-time streaming platform.
- Support data ingestion from various sources, including a mainframe.
- Deliver low latency and high scalability.

Benefits gained from using Aerospike:

- Single data platform that supports low latency and ease of scale.
- Data from various sources streamed into Aerospike in real time.
- Mainframe data is streamed via JMS to Kafka into Aerospike.
- Lowest TCO, while exceeding competitors' capabilities.

Global Top 3 Brokerage

This customer is one of the world's largest brokerage firms and has added use cases with Aerospike over several years. With a strong emphasis on customer service, it's essential to provide real-time capabilities to its millions of retail and business customers, from fraud detection to portfolio access, to risk management. These capabilities historically depended on a mainframe and other legacy applications.

Customer requests are highly concentrated in the first hour of each day, and the brokerage had to set its mainframe capacity to meet this peak demand. This was very expensive, and they sought a way to augment the mainframe.

Phase 1 - Cache

To address this, the brokerage firm initially pulled all of that data together in a database that serves as a real-time cache. The dozens of source systems are ingested and joined in a recurring batch process, and then changes to the source data are updated in the cache via a messaging queue.

Figure 8 illustrates this. Aerospike sits between the legacy (including mainframe) systems on the right and the modern, client-facing applications on the left. In the brokerage's deployment, a periodic set of ETL jobs populates Aerospike via IBM MQ, along with a message processor, and Aerospike Connect for JMS.

Aerospike serves as a low-latency, high-scale cache here. All of the customer account views now access the Aerospike database which is optimized for speed and availability. Similarly, modern applications around credit risk and fraud now read from Aerospike.

Historically, when the brokerage firm served data to a customer-facing application, it had to retrieve data from over 20 legacy systems and applications. This was costly and suffered from inconsistent performance as any delay in one system would slow customer interaction. The brokerage's data showed that 80% of its workloads were for reads, so using Aerospike to cache these reads was a clear priority for phase one.

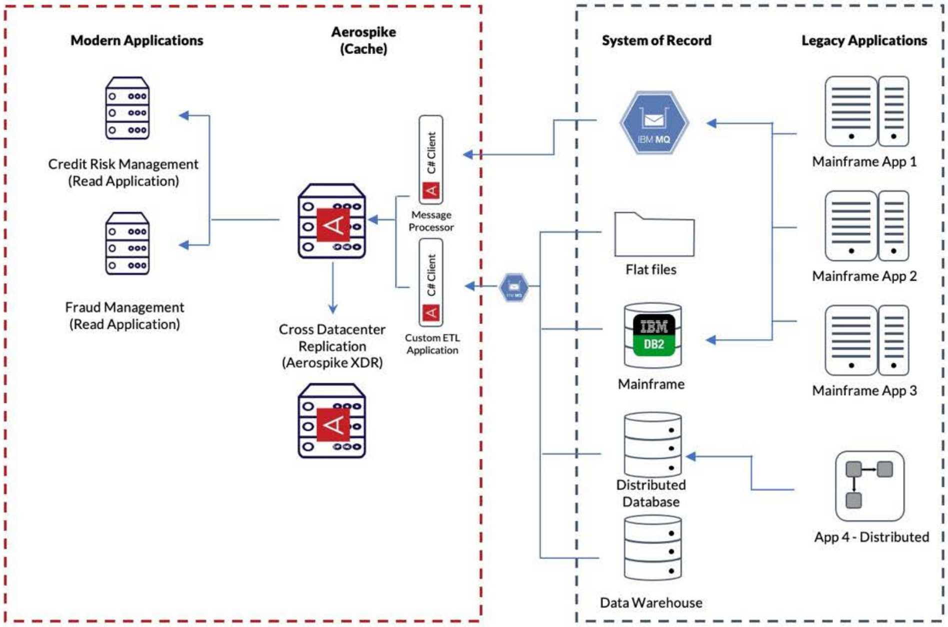

Figure 8 - Top 3 brokerage starts with Aerospike as a cache

Phase 2 - Intraday operational data store

In the first phase, any writes first went back to the legacy infrastructure before reaching Aerospike. Thus there would be a delay as applications accessed the Aerospike cache during the trading day. If an end customer initiated a stock sale, their account balance would not immediately reflect the update.

In this second phase, Aerospike handles both reads and writes, including transactions. Since so many legacy applications have been built to pull from legacy databases, this second phase had to add writes to the modern database while still providing asynchronous updates to the legacy system of record.

With Aerospike's ability to ingest huge volumes of data, in this phase it could handle the transaction feeds of all the asset classes into a single Aerospike cluster. This is highly available, experiencing zero downtime in several years of deployment.

Phase 2 further helped them cut their mainframe costs.

Phase 3 - System of record for modern applications

Today, Aerospike serves as their intraday system of record for all trading. All transactions and updates are captured in Aerospike along with current market data. Any time a customer views their portfolio online, it is pulled from a query off of Aerospike for that real-time visibility. It runs with strong consistency.

This has saved them millions of dollars in mainframe costs. It runs on an efficient 20-node Aerospike cluster. The mainframe still exists for internal finance and reconciliation. Aerospike provides the data for eventual consistency in the mainframe.

The new real-time architecture has also enabled use cases and revenue streams that would have never been possible before. One example is their margin lending business. Margin lending is highly regulated, to protect individual investors and minimize overall risk. For instance, investors can't purchase on margin with high-risk "penny stocks" as collateral. The brokerage has policies in place to enforce these requirements, including a snapshot of the investor's risk profile based on their portfolio. In the past, this risk profile was calculated once per day, and that was in place when the investor requested a margin loan. Now with the new real-time architecture on Aerospike, it calculates these risk profiles every three minutes throughout the day. Given the frequency of trades by margin traders, this intraday trading risk capability helps them stay compliant and support increased trading volume.

Requirements/technology challenges:

- Transaction volumes: 250M transactions and 2M price updates daily.
- Growth rate: Need to grow 4x or more.
- Data consistency: Data inconsistent from mainframe to cache.
- API support: APIs from browser-to-server.

Benefits gained from using Aerospike:

- Server footprint reduction: 150 servers reduced to 20.
- Cost savings: $10,000 saved per day with mainframe offloading.

For more information, see Top 3 Global Brokerage Firm grows margin lending.

Top 5 Asset Management and Brokerage Firm

A separate blue chip financial firm has also moved through a similar pattern with Aerospike. This 75+ year-old company is one of the world's largest financial services companies and currently manages assets totaling trillions of dollars.

Modernization and cloud migration

This financial firm has embarked on a modernization journey, moving nearly all its applications to a cloud-based infrastructure. As we've seen with other modernization efforts, they've specifically prioritized certain areas and capabilities. For one, they've chosen to move the data to the cloud first.

As the firm's Head of Data Architecture and Engineering says, "data is a critical asset, and moving it to the cloud first ensures any applications that are subsequently moved there will have access to the data they need." Aerospike has been a big part of this, particularly for the modern applications with real-time demands.

Modernizing its data warehousing approach has been a primary initiative for the firm's data architecture and analytics team. Prior to modernizing, data was spread across over 100 different data warehouses, data marts, and other data repositories. They've migrated much of that to Snowflake, a modern, cloud data warehouse. However, even this approach gave them challenges.

Phase 1 - Cache

Following the pattern we've described, the firm first utilized Aerospike as a cache. Data warehouse use cases continue to be essential for dashboards and other business intelligence (BI) capabilities, but they typically can't handle modern applications' real-time requirements.

The firm's VP, Data Analytics Architecture discussed this evolution: "As we need to scale to meet our current demand, we had to move to a more real-time, modern data platform, and that's where Aerospike comes in."

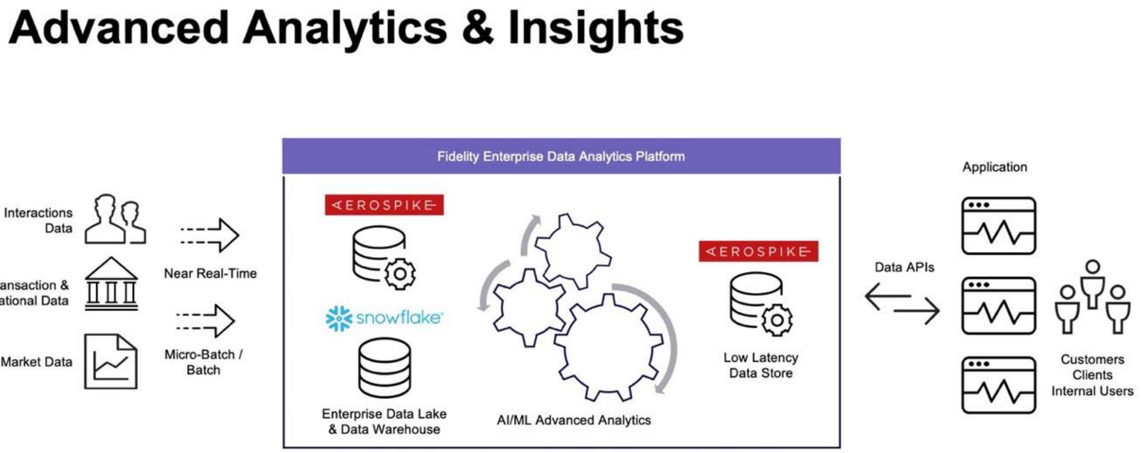

Figure 9 - Architecture with Aerospike as a Snowflake cache

As seen in the architecture diagram (Figure 9), the firm ingests a variety of data types into the Snowflake and Aerospike environment. In particular, they read real-time interaction data and load it into Aerospike. The less time-sensitive data is loaded into Snowflake in micro-batches. The downstream applications that were accessing legacy infrastructure and then Snowflake, now access the Snowflake/Aerospike Enterprise Data Analytics Platform.

Phase 2 - Expanded use cases

With time, the firm has expanded the use cases it deploys on Aerospike. For example, it runs a large AI/ML advanced analytics platform that uses the interaction, transactional, and market data. It then trains and validates models and ultimately calculates scores in real time. Once scored, the platform needs to provide that data in a very low-latency fashion. They accomplish this with Aerospike, which serves the APIs that external and internal applications use for low-latency data access.

While they approached Aerospike initially as the Snowflake cache, they've relied on the different storage engine capabilities to supplement the data environment. With use cases of 100s of gigabytes of data, they weren't able to easily load all that into a memory data store like Memcached or Redis. They found it incredibly expensive and impractical to store all that data in memory with those typical technologies.

Phase 3 - Shared service

The financial firm has continued to modernize its wider environment and expand how it uses Aerospike. They run Aerospike on AWS, but being multi-cloud was a requirement as they build for the future and flexibility.

Part of their modernization journey includes CI/CD and a requirement to "rehydrate" their instances every 60 days. The ability to swap out nodes in the Aerospike cluster without disrupting any read/write operations was a huge benefit in complying with this rehydration mandate. The uptime and reliability have given the firm confidence to broaden the adoption beyond its core analytics team.

> ***"If you look at Oracle or even Postgres or MySQL databases, it's hard to keep something always on, always running. And with Aerospike it was really, really easy. And in fact, our group actually built a managed service around Aerospike to offer this to all the other business units."***
>
> - *VP, Data Analytics Architecture*

Requirements/technology challenges:

- Incorporate warehouse/analytics into personalization, market risk analysis, and other operational processes at scale.
- Provide more accurate, near-real-time decision support with realistic analytics.
- Enable digital capabilities through highly scalable APIs across business units.

Benefits gained from using Aerospike:

- Modernized cloud technology stack with support across multiple cloud service providers.
- Superior performance for mixed workloads of reads and writes.
- Multiple storage engines, including memory, flash, device, and file.
- Easy to set up, manage, and comply with enterprise cloud and security requirements.

Summary

Mainframe modernization is critical for internet-scale business environments. Supporting such efforts requires a proven methodology and a data platform capable of meeting aggressive performance, scalability, availability, security, and operational demands at a manageable cost.

Aerospike's approach to modernization enables firms to accomplish their goals in phases, minimizing risk and delivering tangible business benefits at each step in the process. Firms can extend their mainframe system usage and improve their customers' experience while building a new, modern, scalable data infrastructure that supports them even at a petabyte scale. The flexibility, power, and robustness of Aerospike are proven by its ability to be used as a cache, an operational data store, and finally, as the new system of record. As data volumes and transactional workloads increase, Aerospike's cost advantages over a mainframe-only approach become more pronounced. Aerospike's proven mainframe migration path allows its customers to extend not only the life of their mainframes but also obtain a modern, cost-effective data architecture.

Additional resources

- Case Study - Global Top 3 Investment Bank and Brokerage
- Case Study - DBS Bank Case Study
- Video - Fidelity: Caching Snowflake data with Aerospike for advanced analytic
- Benchmark - Petabyte Scale Benchmark
- Solution Brief - Aerospike Multi-site Clustering
- White paper - Query real-time operational data at scale in Aerospike using SQL

About Aerospike

The Aerospike Real-time Data Platform enables organizations to act instantly across billions of transactions while reducing cloud infrastructure up to 80%. The Aerospike data platform powers real-time applications with predictable sub-millisecond performance up to petabyte scale with five-nines uptime with globally distributed, strongly consistent data. Applications built on the Aerospike Real-time Data Platform fight fraud, provide recommendations that dramatically increase shopping cart size, enable global digital payments, and deliver hyper-personalized user experiences to tens of millions of customers. Customers such as Airtel, Experian, Nielsen, PayPal, Snap and Yahoo rely on Aerospike as their data foundation for the future.

For more information, please visit https://www.aerospike.com.