MEDICI™

# DATA PLATFORMS

## IN FINANCIAL SERVICES

## THE NOSQL EDGE

MEDICI™

# WHITE PAPER

## Data Platforms in Financial Services: The NoSQL Edge

# Contents


NODE 06
NODE 05
NODE 02
NODE 04
BLOCK 01
BLOCK 01
NODE 01

# Executive Summary

The financial services industry is a crucial cog in the functioning of an economy. The industry acts as a force multiplier of economic activity from storing to creating wealth and providing access to credit. It has always strived to evolve with the changing needs of the consumer.

The digitization of financial services has been an ongoing process for the past few decades. The transition has not always been smooth and easy from legacy mainframe systems of the 60s and 70s to today's cloud-based financial services. Customers are increasingly adopting digital channels to access banking services and other offerings. **Neobanks have launched that are completely digital, without having any branch banking.** Financial services can no longer offer generic solutions that target a wide range of consumers. Be it an individual working in a multinational corporation or a student, a neighborhood business, or a large-scale enterprise catering to a global populace, the needs of each client are unique. To address this diversity, institutions are consuming data from multiple sources inside and outside the enterprise. These include structured data from sources like enterprise systems, market systems, and government databases, as well as unstructured data from social networks and media.

This white paper explores the criticality of real-time data processing in financial institutions and the common challenges they face. The growth in online transactions and web applications is driving a focus on fraud prevention, customer experience, **and a new digital imperative that requires new thinking on how data is stored, managed, and processed**. However, the challenges posed by legacy infrastructure, such as data inconsistency, difficulty in transitioning from mainframes to distributed workloads, and the absence of real-time data processing, have an impact on performance and intended business outcomes.

The emphatic acceptance of modern platforms and **technologies such as NoSQL databases (which are powering tech giants like Facebook, Amazon, and Google) has provided the opportunity to tackle these challenges effectively.** This report explores the various facets of NoSQL databases and how their growth has played a crucial role in transforming several financial services business areas. FinTech companies such as neobanks, online trading platforms, and payment platforms are already incorporating NoSQL databases in their architectures and are reaping the benefits of being agile and resilient.

This white paper also investigates the several advantages that **NoSQL offers over traditional databases for a financial institution**. The ability to handle structured and unstructured data formats, ease of connecting legacy systems with new, speedier front-end systems, horizontal scaling, and faster implementation are some of the points that are explored in detail. It also analyzes how each of these advantages translates to better performance in the various functions of a financial institution ranging from fraud prevention, better customer service and customer 360, targeted marketing, data analytics, and real-time digital identity verification.

Financial institutions are increasingly looking to move operational and delivery models from physical to digital platforms as more and more customers prefer these channels. **NoSQL will play a critical role in enabling this shift and in ensuring the high quality of services** that their new, digital clientele expects from them.

# 01. Application and Criticality of Data Processing at Scale in Financial Services

# Fraud prevention, personalized customer experience, and risk management

Financial services firms have multiple requirements that their data processing architectures must meet. **They process large amounts of data ranging from customer records, transactional data, and records** of multiple interactions with third-party providers such as regulators, vendors, and clients. The real-time processing and analysis of this data enables faster decision-making and provides a rewarding experience to the consumer.

**Fraud prevention is an important aspect** both in terms of financial and reputational damage. Since there is a transfer of wealth involved in these transactions, it is very crucial for these organizations to ensure that there are no attacks by unscrupulous entities.

The collection **of large sets of data on consumer spend and behavior enables firms to offer personalized services** catering to their requirements. Financial firms can cross-market various products and services based on the projected needs of the consumer and their spending capabilities.

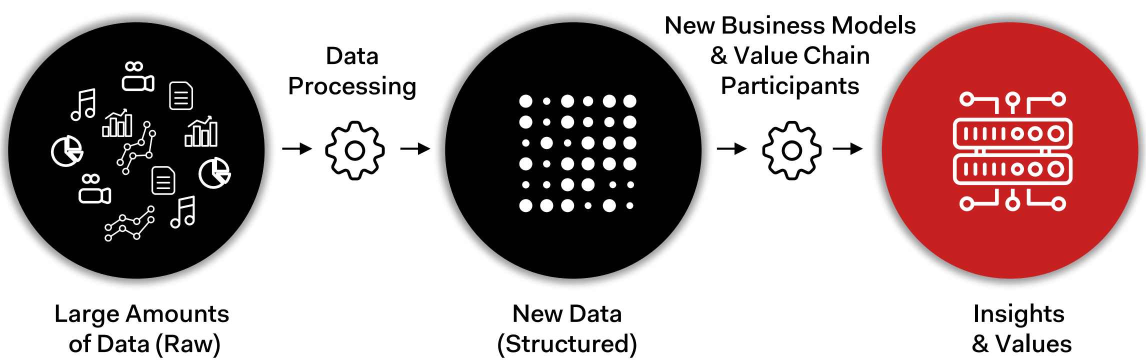
Along with better customer service, **real-time processing also enables for better risk management**. For regulatory purposes, financial institutions must measure and manage the riskiness of customer portfolios, requiring a move to real-time transaction processing for a large part of what a bank does.

# Explosion of data sources, data sets, and formats

For financial services enterprises, data resides in multiple siloed environments such as data warehouses, core applications, finance and risk solutions, analytics solutions, mobile applications, and social media interactions, to name a few. Since this **data is not meaningfully linked across these silos, it has become less accessible**, thereby compromising useful insights into customers, products, partners, sales channels, and financial performance.

The rise of FinTech firms has also put pressure on traditional players in the financial services industry to transform themselves digitally. **Digital-only banks, online trading platforms, and blockchain-based payment systems are some of the new business models** adopted by FinTechs in financial services. These developments have now led to the traditional players converting themselves into data-driven enterprises that can compete with the more agile, web-based FinTech startups.

## Figure. 1 Data from Multiple Sources Leading to Better Insights




To do this, they have to process large amounts of data arriving from various sources such as images, audio, and video generated through mobile applications and other devices. This new data, combined with newer business models and participants in the value chain that increase the digitization of financial services, provides enterprises with opportunities to gain additional insight and value.

## Shift from batch to near real-time to instantaneous

Financial services firms are **moving from batch processing to near real-time or real-time processing**. The faster settlement of trades at online traders has been a result of this shift. Customers attuned to the real-time nature of social media and other web-based services now expect the same in banking and other financial services. When they use an ATM or a credit card or apply for a personal loan, they want the systems they are using to return the right answers immediately.

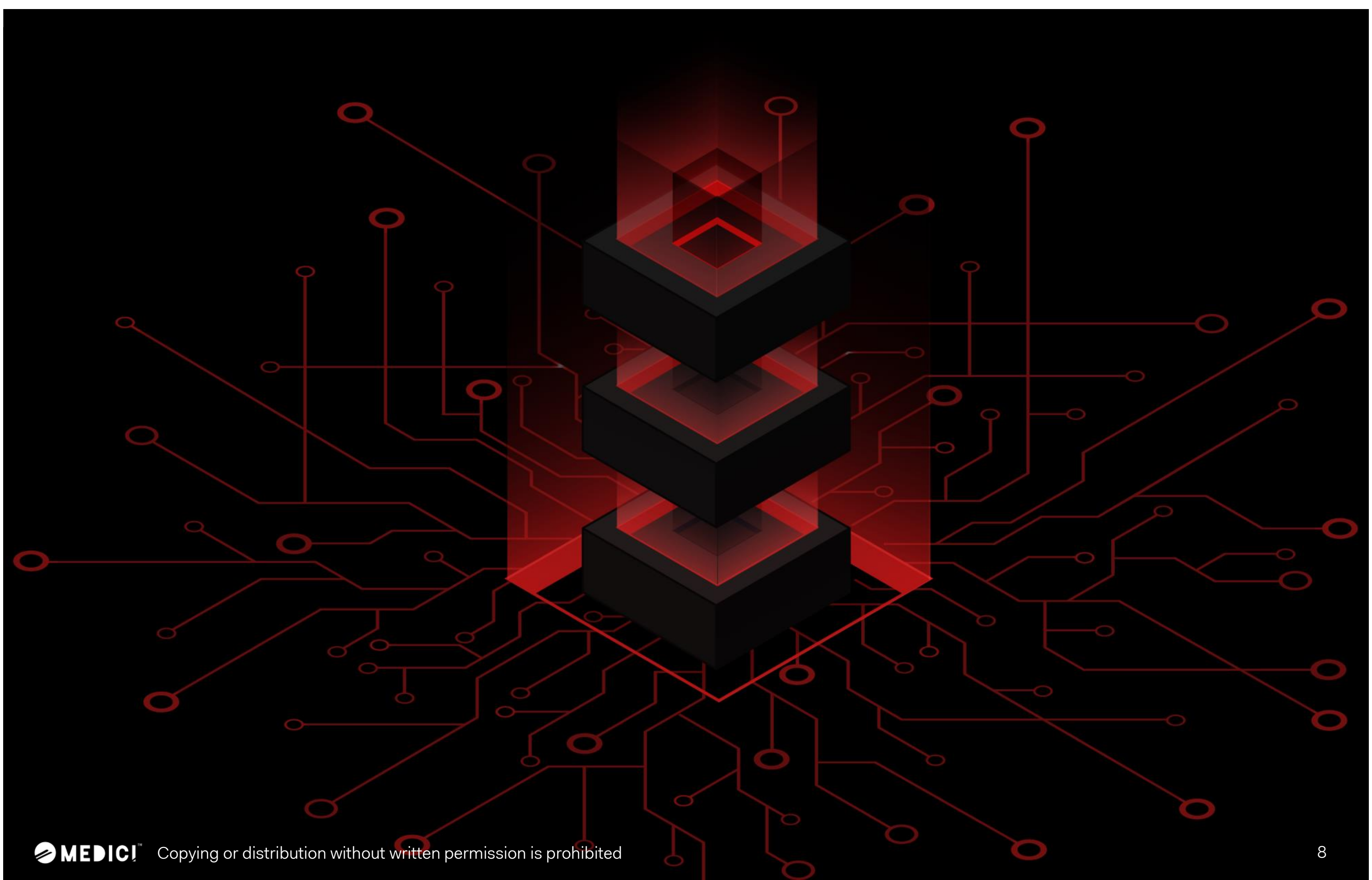
The need for real time transaction feedback and closure has been felt most in the payments industry. Worldwide, central banks are running change programs to make payments and settlements **fully real time, often driven by tight millisecond SLAs**.

## Growth of web-scale applications

The web-scale model **offers multiple advantages, including resiliency, flexibility, and on-demand infrastructure**, which are the requirements of most financial services organizations. The open environment of web-scale makes it easy to standardize protocols and create a unified stack that communicates efficiently between the hardware and software.

Traditional banks are increasing the use of web-scale applications, especially in internet and mobile banking, which enables customers to access financial services through their devices.

These changes in consumer behavior and the competitive landscape have pushed incumbent financial services players to adapt to the open environment of a web-scale infrastructure. **Banks are trying to find the balance between running traditional infrastructure and moving certain areas of their operations and services to web-scale applications** to ensure that the consumer's experience is not impaired. A web-scale architecture offers systems capable of delivering the resiliency, scale, and performance that institutions need to keep customers satisfied.



# 02. Common Challenges in Data Processing in Financial Services

## Data silos caused by organizational silos

Data silos pose a significant roadblock in making business decisions. These are caused by departmental silos within organizations such as marketing, operations, finance, and risk that tend to be confined locally. The big picture gets lost, hampering the full view of a client as well as collaboration across the organization. Agile, iterative product and process design and redesign depend heavily on data from these departments. **In the absence of a clear framework to share and access data across the organization, interpretations are inconsistent, and opportunity loss is high**. For example, a product suggestion from the customer service department based on customer feedback cannot be efficiently executed if there is insufficient data from other departments who are also impacted.

**For a long time, banks have been held guilty of not personalizing and contextualizing to the customer despite having troves of data about them.** Compared to providing solutions, selling products has been a result of data fragmentation and the lack of a unified data strategy in banks.

## The increasing need for real-time processing

In traditional architectures, processing data is largely batch-based. With increased digitization over the years, the focus has shifted to bettering customer experience. In financial services, **better customer experience manifests through real time discovery of products and solutions, personalization, pricing, and adaptation of processes**. Faster transaction response and SLAs are also key expectations. This has necessitated moving data and analytics closer to online processing systems. However, traditional platforms are not designed for such highly scalable operations.

Real-time availability of data also influences risk assessment across various operational domains of a financial institution. Examples are transaction risk in payments, credit risk in treasury operations, and market risk for traders. Overcoming increasing localization of data in distributed systems is a significant challenge.

## Always-on availability and performance

Mission-critical applications, especially consumer-facing apps, demand always-on availability. As technology has improved, an increasing number of processes and interactions have moved to digital channels. **An increase in volumes coupled with the transition to distributed architectures has put the spotlight on performance and availability**. Although clustering and load balancing have existed for a long time, there is a need to have modern ways of managing them. Automated and algorithmically managed load balancing and fault tolerance is the need of the hour.

## Consistency of data

As modern architectures increasingly rely on NoSQL databases that co-exist with relational databases, data integrity becomes a critical aspect. In the process of **storing, accessing, and processing data, unforeseen changes, including deletion, conflicting writes, or stale reads, could happen to data, making it inconsistent across systems**. Real-time systems from legacy NoSQL providers have not historically been able to be strongly consistent.

At the same time, consistency needs to be balanced with performance and availability.

## Transition from mainframes to distributed workloads

The bulk of back-office processing, compliance, and settlement operations in financial services are managed by mainframe systems due to their indisputable stability. However, when the requirement is to manage very sophisticated, frequent, fast-paced, and complex calculations such as risk and online analytics, **there is a need to move these workloads to distributed real-time databases to serve those operational and transactional applications**—linearly scaling mainframe systems to handle workloads that they are not designed for adds prohibitively to the overall total cost of ownership (TCO).

Additionally, an increasing number of datasets do not work well with relational systems. E.g., social media content. The capability to handle non-tabular data is one of the key reasons for the success of NoSQL databases in recent times.

## Processing data for AI/ML in real-time

For machine learning models to work in real time, continuous data streaming is essential. **Historical data from multiple sources across the organization need to be combined and be subjected to machine learning (ML) algorithms on a real-time basis**. For example, in fraud analysis, machine learning helps to detect and recognize millions of patterns on a user's purchasing history instead of the few captured by creating rules. Fraud can be predicted in a large volume of transactions by applying machine learning algorithms to raw data. Machine learning calls for highly scalable data stores and processing that can accommodate both structured and unstructured data. In most cases, distributed systems are required necessitating the combination of base systems of record with surrounding applications.

The ETL layer that typically handles this multi-source processing must support extreme throughput for write operations and must be able to accommodate a high number of requests, such as balance inquiries (in the case of ATM transactions). There is a need for platforms supporting Big Data and machine learning frameworks to work with advanced libraries and languages.

# 03. Data Handling and Processing in Traditional Architectures

## Not designed for extreme real-time workloads

With traditional relational databases, we have to choose which workload to optimize. They can either handle operational workloads that cover day-to-day business transactions or analytical workloads intended for business intelligence and analysis. **We cannot have both as relational databases are unable to handle the mix of these two workloads together**. These databases were designed to be specialized at the cost of flexibility; hence they are not known to handle these multiple workloads efficiently.

## Vertical scaling

Relational databases offer vertical scaling as they are designed to run on a single server to maintain the integrity of the table mappings and bypass the problems of distributed computing. With this architecture, if a system needs to scale, customers must buy bigger, complex, and more expensive proprietary hardware with more processing power, memory, and storage. **Upgrading is also a challenge, as the organization has to go through a lengthy acquisition process**, and then take the system offline to make the change. All this while the number of users is increasing, causing greater strain and heightened risk on the resources.

## Concentrates on data integrity over performance

Data can undergo multiple operations in support of decision-making, such as capture, storage, retrieval, update, and transfer. **Data integrity verifies that data has not been altered in transit from creation to consumption**. It can be considered as a measure of the validity and fidelity of a data object. Relational databases concentrate more on data integrity at the cost of performance primarily because:

- Relational databases can become complicated as the amount of data grows, and the relationship between various sets of data becomes complicated.
- Relational database systems can be complex as they lead to siloed databases where the information cannot be shared from one system to another.

## Built for persistence and batch processing

In storing data in a computer system, persistence means that the data survives after the process with which it was created has ended. For a data store to be considered persistent, it must write to non-volatile storage. Traditional RDBMS store persistent data in the form of tables and records and cannot store objects and their relationships. The objects have necessary features like inheritance and persistence, which do not translate well into tables and records.

**In batch processing, a group of transactions is collected, entered, and processed over a period of time.** Batch processing is asynchronous to one's applications and hence harder to integrate and manage. A typical RDBMS or relational database management system is efficient in batch processing as it involves structured data, which makes it ideal for real-time online transaction processing.

## Not in-memory

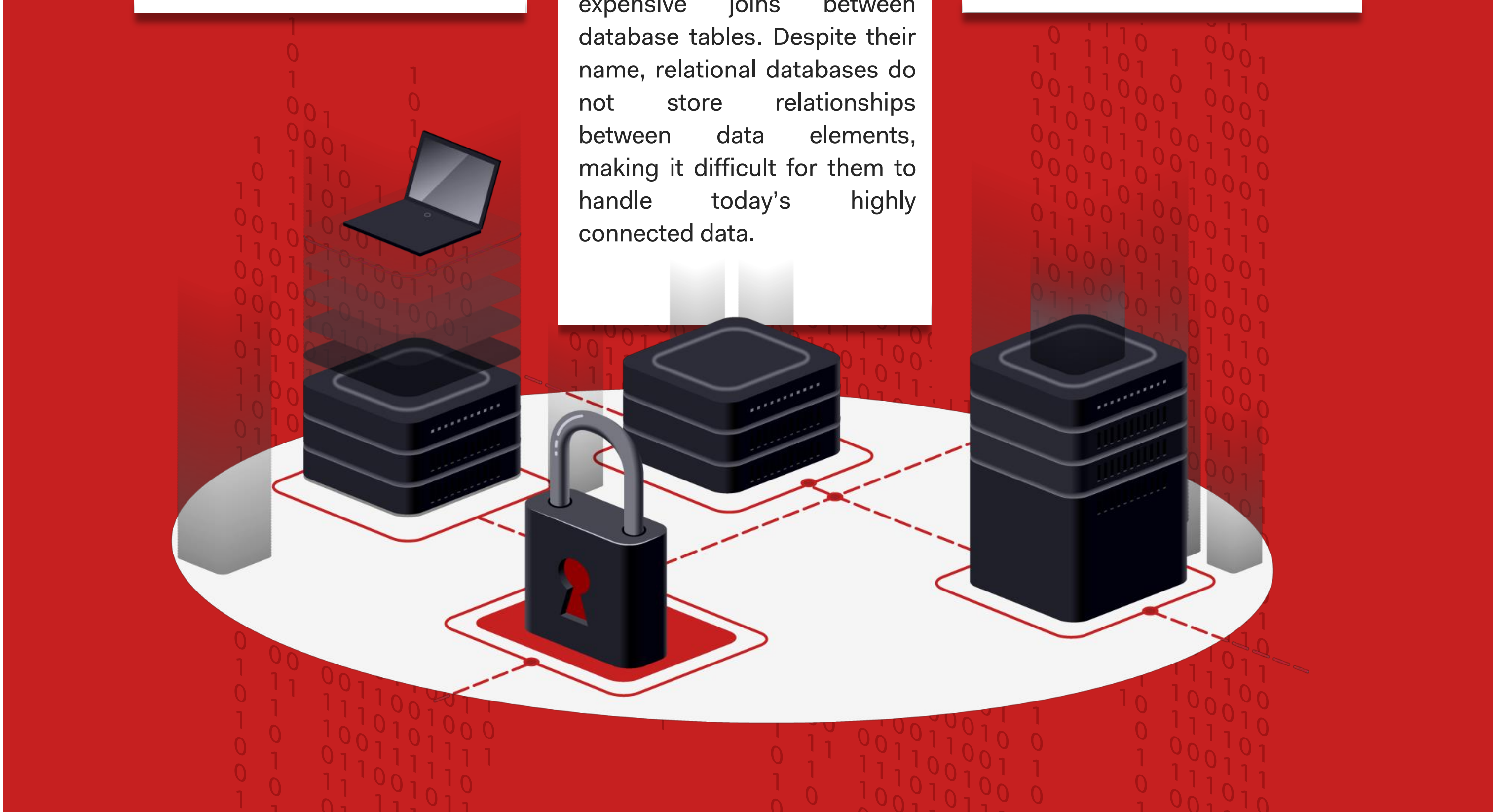
An RDBMS with table space stores files on SSDs. The RDBMS will still cache or buffer that data as it is a built-in function of the DBMS, and it was relevant when all data persistence was accomplished using disk storage. **Most SQL-based DBMSs are not designed to run in memory, and hence, this puts an additional effort to minimize disk IO and paging**. A DBMS works very hard to keep the relevant data in memory and the cache; hence the IO is slow. This is because database data is significantly larger than the main memory, and it is volatile. RDBMS ensures work with write-ahead logging to a non-volatile store and other techniques to ensure data is never corrupted, even in case of an unexpected shutdown.

## Slower response when implemented for real-time feedback

Many SQL queries still are not fast enough to support a particular application's needs. **They struggle with the rapid growth is not only in the velocity and volume of data but also in its variety, complexity**, and interconnectedness, i.e., the data relationships present in a dataset. Relational databases were designed for tabular data, with a fixed schema and consistent structure. They work best for problems that are well defined at the outset. However, attempting to answer questions about data relationships with RDBMS involves numerous and expensive joins between database tables. Despite their name, relational databases do not store relationships between data elements, making it difficult for them to handle today's highly connected data.

## Caching layers built on top of operational layers

Caching layers built on top of operational layers automatically caches frequently accessed data from the origin database. In most cases, the underlying database will utilize the cache to serve the response to the inbound database request, given the data is resident in the cache. This greatly increases the database's performance by lowering the request latency and reducing system and memory utilization on the database engine. **A crucial characteristic of an integrated cache is that the data cached is consistent with the data stored on disk by the database engine.**



# 04. NoSQL Databases in the Changing World of Financial Services

# Digital transformation in financial services and the changes imposed by it

Financial services institutions, especially banks, have long been struggling with legacy systems. When implemented in the '60s and '70s, they ensured that these financial institutions were ahead of the curve in terms of computerization; however, now they have become massive IT burdens that are dragging these behemoths into the ground. **One of the major issues is the reduced pool of IT staff who are well-versed in these legacy systems**.

## Issues faced in changing these legacy systems:

**COST**

Most of the time, firms have to go for a complete overhaul, and that involves massive costs in terms of both hardware and software procurement.

**CHANGE MANAGEMENT**

Most of the top management in financial services firms are still resistant to change. They fear short-term problems over long-term benefits and are unwilling to take the call to bring about this change.

**BREAK-IN OPERATIONS**

Every second is crucial for the financial services industry. The industry is apprehensive that any wholesale change to the systems could result in a long-term break-in operation and potential loss of business.

**SKILLED MANPOWER**

There is a reduced pool of IT staff who are well-versed in these legacy systems. Their absence makes it difficult to engage in IT transformations effectively.

### Other Issues with Banking Legacy Systems

- Big banking systems implemented in the '70s were written primarily in COBOL, which was introduced in the '60s.
- In the '70s and '80s, banks were the leaders of tech innovation and saw the birth of products such as ATMs, BACS, and international card payments.
- With the arrival of the internet, customers wanted access to financial transactions from the comfort of their homes, and banks responded by investing in client-side software.
- Gradually, they worked out that client-side software was not the right solution compared to the standards established for internet traffic and figured out that a website worked way better for a fraction of the cost.
- The Big Data Revolution that has taken hold in recent years is now causing financial institutions around the world to enhance their capacity to access and mine data from all sorts of sources.

## The emphasis of digital transformation in financial services during the last 20 years has focused on the following points:

1. The emergence of digital delivery of financial services products and services
2. Increased automation of the financial services enterprise (internal digitalization focused on process automation for efficiency and cost optimization)
3. Increased reliance on data for analytics-based business decision-making
4. The emergence of cloud —both platforms and technologies—brought in the need to change legacy architectures to take advantage of the agility and lower cost of cloud computing.

Many banks are still struggling with shifting from the legacy systems and are severely behind the curve. Neobanks are hot on the heels of traditional banks, and with more digital avenues available, customers will be frequenting physical branches even less. This is the time for banks to go all out in technology adoption and ensure that they usher in a new era in innovation and customer satisfaction.

## What is a NoSQL database?

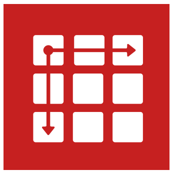
One of the changes seen in IT systems of financial institutions is the adoption of NoSQL databases. The next few sections will examine what a NoSQL database is and how its adoption can help in the digital transformation of the financial services industry.

NoSQL is usually referred to as not only SQL, non-SQL, or non-relational databases. NoSQL databases such as key-value pairs are built for prime throughput versus relational databases with relative dependence. Using loose dependencies and quick indexes **NoSQL databases are perfect for streaming analytics and IoT applications** because data can quickly be stored and referenced from multiple, disparate data sources.

NoSQL databases represent an approach to data management and database design that's useful for very large sets of distributed data. They encompass a good range of technologies and architectures, seeking to resolve the scalability and Big Data performance issues that relational databases were not designed to deal with. **NoSQL is particularly useful when an enterprise has to access and analyze massive amounts of unstructured data or data stored remotely on multiple virtual servers within the cloud**.

# Types of NoSQL databases

## Columnar Database



In column-oriented NoSQL databases, data is stored in cells grouped in columns of data instead of rows of data. Columns are grouped into column families that contain a virtually unlimited number of columns that can be created at runtime or while defining the schema. In these databases, read and write functions are done using columns rather than rows. The benefits of storing data in columns over relational DBMS are fast search, access, and data aggregation.

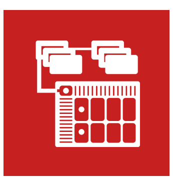
Column databases are primarily used in systems that maintain counters, content management systems, blogging platforms, services that have expiring usage and systems that require heavy write requests.

## Graph-based NoSQL Database

Graph databases are essentially created upon the Entity-Attribute-Value model. Entities are also known as nodes, which have properties. It is a very agile way to describe how data relates to other data. Nodes store data about each entity in the database. Relationships describe a relationship between nodes, and the property is the node on the opposite end of the relationship.

Graph-based NoSQL databases are well-suited for graph-based search, network and IT operations, and social networks.

## Key-value store NoSQL Database



Key-value stores are the simplest NoSQL data stores to use from the perspective of APIs. The client can either get the value for the key, assign a value for a key, or delete a key from the data store. The value is a hold that the data store just stores without caring or knowing what is inside. It is the responsibility of the application to understand what was stored. Since key-value stores always use primary-key access, they generally have better performance and can be easily scaled.

However, new-age Big Data applications have sparse datasets, and traditional databases cannot handle them as effectively. In the case of facilitating efficient selection of a subset of the dataset based on business constraints, **hybrid NoSQL databases use secondary indexes by providing alternative access paths to the base records**. Alternate to a scan, these special index structures can help identify the records that qualify, and then only those records are retrieved from the base table. Secondary indices are defined over one or more attributes and are often constructed over non-primary-key attributes.

Key-value databases are primarily used for storing user session data, maintaining schema-less user profiles, and user preferences. A lot of data generated in financial institutions is time-series data as the industry is highly regulated, and adequate audit trails are required to be maintained. NoSQL databases are used to handle such data that has a timestamp and includes a time order.

## Document Store NoSQL Database

Document store NoSQL databases are similar to key-value databases in that there is a key, and there is a value. Data is stored as a value, and its associated key is the unique identifier for that value. The difference here is that, in a document database, the value contains structured or semi-structured data. This structured/semi-structured value is referred to as a document and can be in XML, JSON, or BSON format.

Document store databases are well-suited for e-commerce platforms, analytics platforms, content management systems, blogging platforms, and more.

## Growth of NoSQL databases

**NoSQL technology was originally created and used by Internet leaders such as Facebook, Google, Amazon**, and more, which required database management systems that could read and write data anywhere in the world while scaling and delivering performance across large data sets and billions of users.

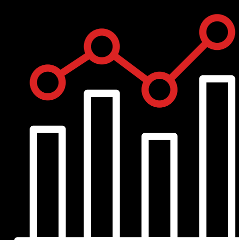
However, a large number of institutions worldwide are now adopting NoSQL databases to handle bigger datasets more efficiently and to ensure analytics and real-time decision-making using the same. **Industries such as media, financial services, e-commerce, research, and analytics are increasingly moving towards NoSQL databases** to handle their data management because of its major advantages, such as lower cost, distributed computing, and high scalability.

### Some of the major NoSQL database providers include:

| | | | | |
|---|---|---|---|---|
| AEROSPIKE | amazon web services | DATASTAX | Microsoft | Couchbase |
| Google | MarkLogic | mongoDB | neo technology | Objectivity |

As per a report from Allied Market Research,

- The global NoSQL market is expected to reach **$22.08 billion by 2026.**
- The industry is expected to grow at a **CAGR of 31.4%** (2019-26) [1].



As more and more people start transacting on the internet, the volume of data generated increases manifold, and companies are looking for data management solutions that can handle them better.

Along with efficiency, NoSQL databases also offer cost-effective modes to handle data as compared to traditional databases. **A Forrester report had concluded that NoSQL products saved companies more than 50% of the cost compared to conventional DBMS** [2]. NoSQL database implementation is relatively easier and typically uses less expensive servers to manage rising data and transactions. In contrast, RDBMS databases are costlier and use larger servers and storage systems. So the storing and processing data cost per gigabyte in the case of NoSQL is several times less than the cost of RDBMS. These advantages are causing enterprises to switch over from relational databases. According to Gartner, by 2022, 50% of existing proprietary Relational Database Management Systems (RDBMS) instances will have been converted or be in the process of conversion [3].

Organizations like Aerospike are at the forefront of the growth by offering real-time NoSQL data platforms to multiple leaders in industries such as e-commerce, advertising, online gaming, telecommunications, and particularly financial services. **The Aerospike solution enables greater efficiency in areas such as customer experience, payment processing, fraud prevention, pre-trade risk reduction, and customer 360, among others.**

# 05. Where NoSQL Databases Fit in Financial Services

## NoSQL databases are better equipped to handle larger data sets

Some applications need simple object storage, whereas many others require highly complex and interrelated structure storage. NoSQL databases provide the required support for a range of data structures ranging from simple binary values to complex hierarchical structures.

**NoSQL databases can handle smaller binary values, lists, maps, and strings at high speed in key-value stores**. Slightly more complex, related information values can also be handled by key-value systems with complex data types or be grouped in column families within big table clones. A web of interrelated information can also be managed flexibly and related in triple and graph stores in a NoSQL database.

**NoSQL databases are increasingly being used for Big Data analysis. This is because NoSQL databases typically follow the BASE approach** instead of the ACID approach of relational databases (although this is now beginning to change). By being Basically Available (BA), it guarantees the availability of the data at all times. With Soft state (S), the state of the system could change over time, and Eventual (E) consistency ensures that the system will eventually become consistent once it stops receiving input. Legacy NoSQL databases, for the most part, give up the A, C, and D requirements, and in return, they improve scalability with Aerospike having notable capabilities. These include single-record ACID transactions and strong consistency at still great speed and scale – including across data centers.

## Ease of handling both structured and unstructured data formats

RDBMS cannot be considered the best solution for all situations as they cannot meet the increasing growth of unstructured data. Traditional databases cannot process unstructured and unpredictable information. However, modern applications demand an occurrence-oriented database. These have to be highly flexible and operate on a schema-less data model.

If a financial services institution only plans to pull data similar to an accounting spreadsheet, i.e., the basic tabular structured data, then the relational model of the database would be enough to fulfill the business requirements. However, current trends demand just the opposite: storing and processing unstructured and unpredictable information. This information comes from multiple sources. With the increasing size of the database or increasing number of users, RDBMS using SQL suffer from major performance and scalability issues, thereby making real-time unstructured data processing more difficult.

On the other hand, **NoSQL is a flexible and cloud-friendly approach to process unstructured data with ease dynamically**. With the availability of several mobile and web applications, it is very common for financial institutions to have millions of users who will generate a lot of unstructured data.

**NoSQL databases simplify the representation of multi-level hierarchies and nesting using the JSON (JavaScript Object Notation) format.** In the dynamic schema universe where changes happen every hour, it is impossible to stick to the 'get it right first' model, which was a success with the outdated static schema.

Web-based enterprises such as Amazon, Facebook, and even modern-day neobanks, require a database with NoSQL capabilities that can match up with changing data models, thereby offering them greater flexibility in operations while maintaining resiliency and uptime. They have the added advantage of not having to deal with legacy systems, unlike traditional banks. This enables them to work in real time and offer a unique customer experience not seen before in the financial world. From instant account opening to the quick transfer of money and personalized services, neobanks have caught the imagination of customers. A lot of this is enabled through real-time processing of data and business analytics. NoSQL databases can help traditional financial services organizations and neobanks alike to enhance their offerings further and innovate on products and services using real-time data analytics.

## High-performance handling designed for real-time feedback at web-scale

The main purpose of keeping a database in-memory is performance. Reading and writing data that is in-memory is typically faster than data stored on a flash drive or a disk. As a result, the user does not have to wait for disk I/O in order to update or query data with two notable exceptions. The first is the **Aerospike data platform has both a Hybrid Memory Architecture™ and an All-Flash storage engine, each of which can access from disk virtually as fast as in-memory**. The second instance is systems leveraging Intel Optane® DC Persistent Memory, which "raises all boats," that is, makes in-memory systems and the two aforementioned Aerospike approaches faster as well.

For in-memory systems, data is added to the main memory with the help of specialized indexing data structures. This data is always available in-memory but is also persisted to disk with logs and database snapshots. Lastly, the ability to read and write data so quickly and efficiently in an in-memory database enables mixed transactions and read/write workloads (although write workloads can be much slower and tend to bog down legacy in-memory NoSQL systems).

Many real-time applications in the modern world need the performance of NoSQL database structures. This helps ensure that decision-makers in the financial services industry have the most relevant and timely information when interacting with a customer, vendor, or the management in their own company. With this information in hand, they can prepare and present an accurate picture to the concerned parties rather than having to wait for information to be returned or basing their decision on outdated information. Having this data on hand so quickly provides a sure-shot competitive advantage for enterprises using tools such as customer relationship management solutions.

## Built to connect legacy systems with newer and faster front-end systems

Many traditional financial institutions still use legacy systems for various functions in their organizations. Some of these systems have large amounts of data, and the process to convert them involves humongous effort and high costs. As a result, many of these organizations continue to retain these systems. However, they work with modern databases in areas such as front-end operations that have a lot of web-based interactions. **NoSQL offers a powerful way to monitor key performance indicators in real-time, which helps with decision-making, SLA adherence, and forecasting**.

Under a legacy approach to business intelligence, data from such online transactions are likely to languish in data warehouses. However, with a NoSQL database in place, that same information could instead be accessed for providing real-time insights into consumer behavior. For example, in a digital front-end for a traditional bank, key-value stores (for storing user sessions), graph stores (for tracking customer spend), document stores (for payment transactions), and column families (large scale web analytics and support for reports) can all be utilized through NoSQL.

Firms have to keep the following points in mind while linking legacy systems with NoSQL databases:

1. Identify integration points in the legacy stack, as integrations can happen at the Object Relational Mapping or database layer without any additional overhead
2. Ensure there is a strong architecture in place to enable the seamless integration of NoSQL and in-memory computation systems

## Built to drive two-paced development of modern architectures

The massive growth in mobile, Big Data, and cloud technologies has greatly changed market dynamics in every industry, including the financial services industry, driving the convergence of the digital and physical worlds, and changing customer behavior. A new approach is required to achieve such a scenario, **one that allows web-scale innovation in order that technology teams can meet changing business requirements** while enabling existing systems to continue running reliably, securely, and efficiently.

## Can scale better horizontally as the data grows

NoSQL databases manage the sharding of a database across several servers. Hence, if data storage requirements increase, one can continue to add inexpensive servers and connect them to the database cluster, making them work as a single data service. This is known as horizontal scaling, and since it follows the denormalization concept where it can store duplicates, there is no need for a single point of failure in a NoSQL database. **Providing durability and high availability of a NoSQL database by using less expensive hardware and storage is one of NoSQL's major advantages.**

In a relational database, we would need to purchase more powerful and thus increasingly more expensive hardware to scale up vertically. If we were to increase the amount of data storage, we would have to increase the cost of the hardware needed as well substantially.

## Easier and faster to implement compared to traditional databases

NoSQL databases have increased in popularity with the rise of Big Data-based applications. Unlike relational databases, **NoSQL databases are much cheaper to scale, capable of managing unstructured data, and better suited to the current agile development approaches**. The advantages of NoSQL technology are compelling, but replacing a legacy relational system can be daunting.

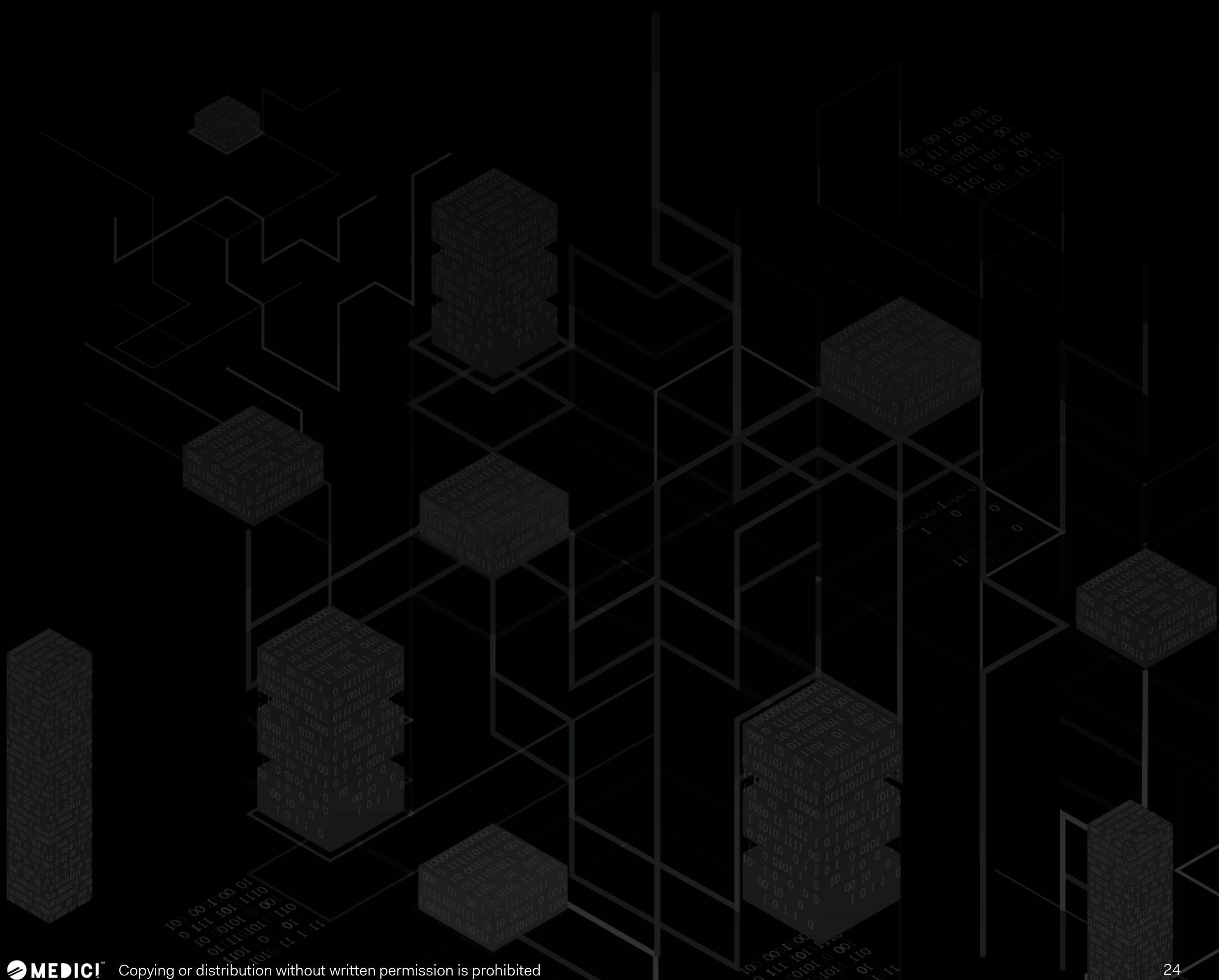
NoSQL databases are often open-source; users can download the software and build their applications on it for free (albeit with either scale or feature limitations and (or) lack of support). They are also inexpensive and easily available across multiple software vendors. **NoSQL database implementation is easy and typically uses cost-effective servers to manage exploding data and transactions. In contrast**, RDBMS databases are more expensive and use big servers and storage systems. Hence the storing and processing data cost per terabyte in the case of NoSQL can be many times less than the cost of an RDBMS.

## Support for better performance handling

The performance of a relational database model can be enhanced with more powerful hardware. However, as discussed earlier, this process is costly and time-consuming. The continuous upgrade process of hardware is the problem that NoSQL databases are trying to solve. Instead of having to purchase new servers with more memory, **NoSQL databases apply a scale-out model where we can easily add new servers to the cluster**. This enables the organization to grow without having to replace their existing hardware investment. This benefit alone makes it worth considering NoSQL for many modern use cases.

In the question of multi-row transactional support, it has been observed that many use cases today do not require updating multiple rows and tables in a transactionally guaranteed manner, i.e., as ACID transactions, which is why NoSQL works so well, in comparison. Also, since NoSQL databases use a denormalized model in which all information about an entity is presented in a single record, there is no requirement to update multiple rows.

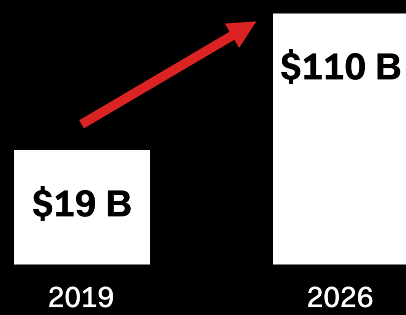
# 06. Opportunities and Business Benefits That Can Be Derived from NoSQL Databases



# Improved fraud prevention and financial crime monitoring

Fraud prevention in financial services is a great example of an application that benefits from real-time analytics. **It is a very crucial aspect for banks, trading institutions, and payment processors**. Failure in these areas results in losses: both financial and reputational, and both false positives and false negatives can have massive business implications.

The global fraud detection and prevention market size stood at **$19.90 billion in 2019 [4] and is projected to reach $110.04 billion by 2026 [5]**. This represents a recognition that fraud is becoming incredibly challenging and can only be addressed with cutting-edge technology.




RSA reports a **600% increase in mobile fraud over the last three years, till 2018 [6]**. As more transactions occur on mobile channels, the element of fraud occurring on the platform is only going to increase.

These days, fraud detection is handled by:

- Analytics
- The ability to apply data to detection through anomaly identification
- Predictive classification
- Clustering

Fraud detection works on various decisions based on knowing the customer, monitoring transactions, and identifying patterns. Many companies are now building transactional analytics systems for fraud to complement their existing architecture.

A NoSQL database offers real-time decision-making on large datasets. Latency plays a crucial role here, and the lower the latency, the more the tasks that can be performed to check the various parameters involved in fraud detection. **NoSQL databases enable real-time analytics on live transactions by removing the latency associated with moving data from operational databases to data warehouses for analytical processing**. Through Aerospike, financial institutions will be able to combine live transactional streaming data with large amounts of historical data for even greater fraud detection accuracy.

## Real-time digital ID validation

ID verification has gained increased significance in the global, digital world. As more and more of the global population connects to the web for their various requirements, it is imperative that firms identify genuine customers against anti-social elements.

In 2021, **cybercrime damages could reach $6 trillion** [7]. The Global Risks Perception Survey by the World Economic Forum identified cyber-related issues, such as cyberattacks and data fraud or theft, within the list of top 10 long-term risks globally.


ID verification firms across the world connect with multiple digital ID databases to ensure that clients are able to verify the identity of the people they interact with, instantly. Many of these clients belong to the financial services industry, who have to ensure that ID fraud does not happen at various levels of interactions with customers and clients. The ID verification firms have to instantly make API calls to these databases and display the results in almost real-time. NoSQL databases reduce read latency in such situations and ensure that these organizations are able to handle real-time customer trust decisions virtually, thereby eliminating false positives and massively enhancing fraud detection.

## Personalized offerings using improved real-time data analytics

Personalization is the buzzword in the financial services industry these days. **Customers do not want generic products and services that cater to a wider population**. Each individual's financial requirements are different, and they expect their bank or any other financial institution to offer them solutions matching their needs.

**Customers are now asking for flexible credit card interest rates that vary based on the repayment and spend capacity**. Insurance policies have to be tweaked based on localized risk factors, and investment professionals have to design bouquets based on individual risk appetite. Banks can use data from multiple sources to offer personalized loan products as well. For example, they can assess the risk of lending to an online vendor by analyzing their inflows from a payment gateway statements, reviews of their product on the online portal, and marketing reach based on the metrics of their social media handles.

For this, organizations must collect all possible information about their customers to prepare a comprehensive user profile of the individual. Those profiles will often have recent user behavior, segments loaded from an analytics system, partner cookies, social media engagements, online shopping preferences, and more. **Companies need robust data management tools such as NoSQL databases to store and access these multiple sources**. These will, in turn, enable the organization to make real-time analytics on user preferences and ensure that products and services are provided to cater to the specific needs of the customer.

# Targeted marketing

**Targeted marketing is closely tied to the personalization aspect mentioned above**. Publishers, advertising operations, and agencies use various platforms and tools for buying, selling, delivering, and targeting display advertisements. These video and mobile ads need to occur in real-time as the customer can arrive and leave a digital property in a fraction of second.

**Targeting marketing has become the most effective medium for brand engagement from social media sites**, displaying ads based on viewer preferences to recommendation engines on streaming platforms and e-commerce sites. This is only possible through large-scale data crunching offered by NoSQL databases using data sourced from user profiles, cookies, and social media engagements. Managing massive amounts of audience segmentation data exchanges behind this transaction between players such as buyers, sellers, and ad exchanges requires low latency and efficient data management on the part of databases.

# Pre-trade assistance

As stated, **NoSQL databases can be used to offer personalized services by studying customer behavior through data from multiple sources**. Another use case of the same is data analytics for trade decisions and risk assessment. Pre-trade assistance uses analytics to study information in real time from various sources, both highly structured market data, and unstructured news information and research. Some of these data sources are also geospatial, such as commodities data.

Instead of just displaying this data, **real-time analytics can be performed using NoSQL databases that provide a holistic view of understanding a particular instrument**. Traders can use this data to arrive at an objective analysis of the instrument's performance and make informed decisions accordingly. Traders can also be alerted about significant changes derived from this information, thereby giving them an edge over others, as they are relying on holistic analytics rather than unstructured data.

# 07. Conclusion

# Conclusion

Similar to how the dotcom boom of the late 90s and early 2000s heralded a change in retail through the arrival of Amazon, or social interaction through Facebook, the financial services industry is ripe for such a seminal change. **This change is already visible with the success of neobanks, such as Revolut in the UK and Nubank in Brazil**, that are targeting different ends of the consumer spectrum with their digital offerings. Through their commission-free trade offering, others, such as Robinhood, are changing the way traditional industries have functioned for decades.

Technology is going to play a massive part in this new environment. Customers who are accustomed to instant updates through social media will expect a similar service from their financial providers. This is only possible by removing constraints posed by traditional architectures that have existed over the years. Traditional databases are struggling to cope with the massive influx of data from multiple sources and multiple formats. On the other hand, **NoSQL databases bring immense value when an enterprise has to access and analyze massive amounts of unstructured data or data that is stored remotely on virtual servers within the cloud**.

NoSQL databases will be one of the pioneering technologies for financial institutions in the coming years, enabling them to succeed in the digital environment. As firms around the world look at cost-effective measures to run businesses, technology will be the main focus. With the growth in data, it will become uneconomical for firms to spend on scaling databases vertically by adding more compute and storage capacity. Instead, with **NoSQL, applications can store their data in purpose-built databases, which is especially true in the case of unstructured data**. This will also aid in robust data analytics for improvements in business performance and customer service.

Financial services have traditionally been innovators in technology. However, the baton has passed on to other areas in recent times, and the industry has struggled to cope with customer expectations. Innovations such as cloud and business analytics are yet to realize their full potential at scale in the industry. Modern, cutting-edge platforms such as **NoSQL databases will help accelerate innovation to achieve its full potential so that the industry meets its consumers' demands and expectations**.

# 08. The Aerospike Advantage

AEROSPIKE

## Aerospike is a global leader in next-generation, real-time NoSQL data solutions.

Traditional architectures call for a large DRAM-based cache in front of a persistent store for high throughput and extremely low latency over large volumes of data. By contrast, **Aerospike's Hybrid Memory Architecture™ (HMA) offers a fundamentally different approach**. By persisting data on fast SSD devices and leveraging primary key indexes (whether in DRAM, SSD, or Intel® Optane™ DC Persistent Memory), many advantages are realized without compromising performance while adding persistence, correctness, and security.

Aerospike empowers customers to **fight fraud instantly, increase shopping cart size, deploy global digital payment networks**, and deliver one-to-one personalization for millions of customers.

Aerospike's global line-up of **customers includes Barclays, a Top 3 Global Brokerage Firm, DBS Bank, European Central Bank, LexisNexis Risk Solutions, and a Top 3 Global Online Payments System**. The company is headquartered in Mountain View, Calif., with additional locations in London, Bengaluru, and Tel Aviv.

---

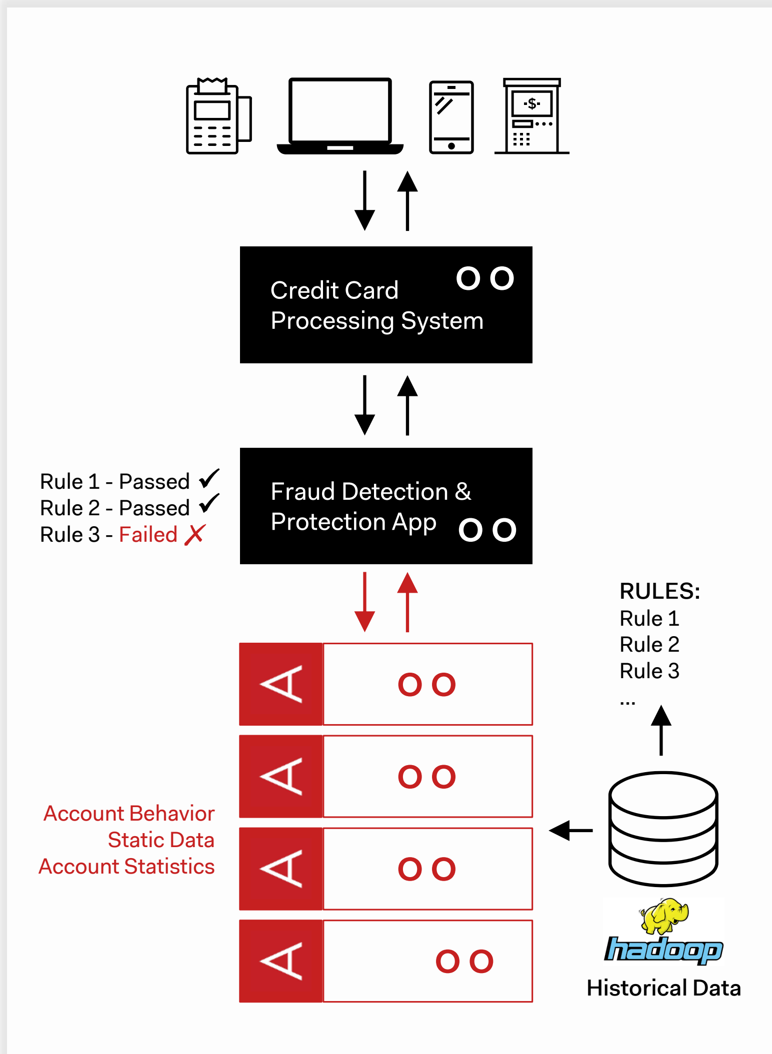
## Aerospike's Hybrid Memory Architecture has made a significant difference in financial services.

### Top 3 Global Online Payments System: Fraud detection for digital payments

One of the leading global payments providers, is minimizing its annual fraud losses by improving its **fraud detection algorithm SLAs by a factor of 30x with Aerospike**.

The firm moved from a 2-layer architecture consisting of Oracle RAC and 360 Terracotta servers to a 20-node Aerospike cluster. Now it runs fraud detection rules against 99.95% of its transactions within its target SLA of 750 milliseconds. The figure here illustrates the firm's fraud detection infrastructure with Aerospike.

For more information, click here.

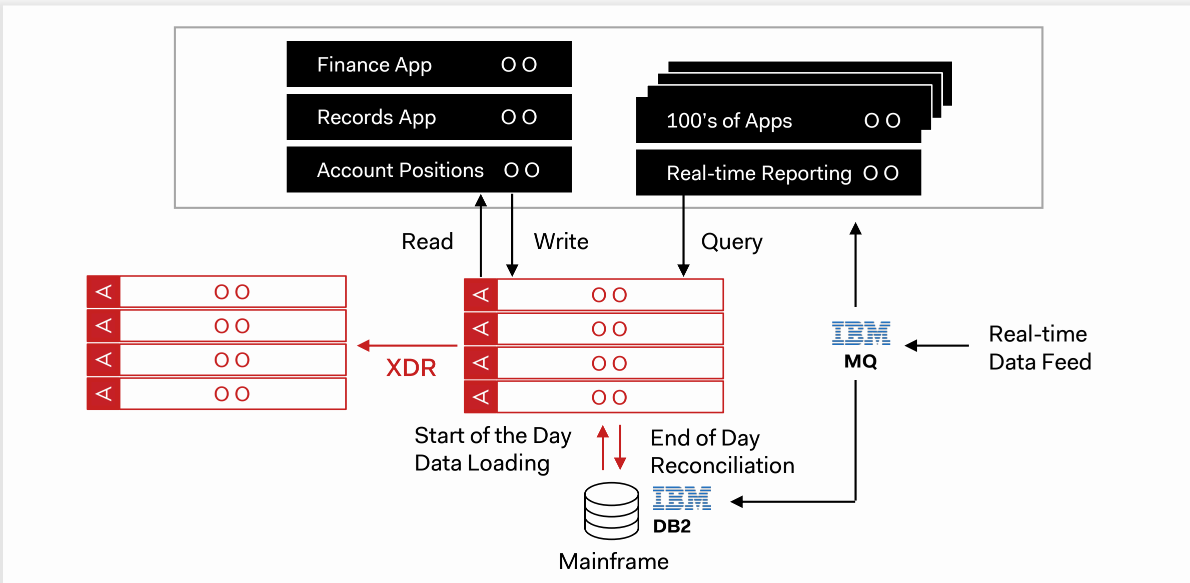


## Top 3 Global Brokerage Firm : Intra-day trading for financial services

One of the largest asset managers in the world, with over $3 trillion in client assets, uses Aerospike to offload work from its mainframe solution for financial trades.

Initially, the brokerage firm used a caching layer over its mainframe DBMS to improve response time. When the firm determined that the cache would need to grow from 150 to 1000 nodes to meet future business needs, it turned to Aerospike. Today, a 12-node Aerospike cluster serves as the record system for intra-day trades, replacing the cache and offloading some work previously done on the mainframe. The mainframe continues to serve existing applications, and the firm regularly transfers data between the two systems, so each can fulfill application-specific needs.

With its modernized infrastructure (shown below), the firm now enjoys a five-fold increase in processing speeds. Database access times dropped to sub-millisecond even though the database size increased from 4 to 14TB. Furthermore, **Aerospike enabled the firm to accomplish this with 90% fewer servers deployed, saving an estimated $10,000 per trading day**.




For more information, click here.

---

## LexisNexis Risk Solutions/ThreatMetrix: Tracking digital identity

ThreatMetrix, a global solution for digital identity intelligence and authentication, uses Aerospike to handle over 130 million transactions a day to manage real-time customer trust decisions.

The firm was processing more than 50 million transactions per day globally, focusing on false positive reduction and fraud prevention (payment) using a custom SQLite clustered solution from which they migrated to Cassandra. However, they were facing highly unpredictable SLAs—with 130bn+ records and growing, leading to significant latencies in cross data center synchronization and very high per API cost of operations.

After shifting from a 96-node Cassandra cluster to a **28-node Aerospike cluster, they were able to triple the performance leading to predictable SLAs and a two-third reduction in response time**. ThreatMetrix is projected to save $3.32 million in three years with the move to Aerospike.

For more information, click here.

# Bibliography


1. NoSQL Market Report, Allied Market Research
2. The Forrester Wave: Big Data NoSQL, Forrester
3. Magic Quadrant for Operational Database Management Systems, Gartner Research
4. Fraud Detection & Prevention Market Size, Share & Trends Analysis Report, Grand View Research
5. Fortune Business Insights Report
6. Quarterly Fraud Report, RSA
7. Cybersecurity Ventures Official Annual Cybercrime Report

# About MEDICI

Since 2013, MEDICI has been pioneering the definition and organization of the new global FinTech industry for the benefit of financial institutions, startups, and investors.

Over the years, it has been a story of numerous Firsts in FinTech!

We were the first independent source of data-driven research dedicated to covering FinTech innovation globally, published every day since August 25th, 2013, now offering an archive of 5000+ insights for the benefit of this new industry.

We built the industry's first content curation and ecosystem collaboration platform in 2015. Today, this proprietary technology supports the world's largest FinTech community of 200,000+ globally across 1000+ enterprises, 13,000+ startups across 65+ sub-segments covering every hub of innovation in every continent.

**Website:** www.goMEDICI.com | **Twitter:** **@gomedici**

# Thank You

**Aditya Khurjekar**
CEO & Founder
ak@goMEDICI.com
@khurjekar

**Amit Goel**
CSO & Founder
amit@goMEDICI.com
@amitTwitr

## Global Contacts

**Salil Ravindran**
Head, Digital Banking
salil@gomedici.com

**Giuseppe Marchese**
Head of Business Development, Europe
giuseppe@gomedici.com

**Subhagini Chaudhary**
Head of Sales, Southeast Asia
subhagini@gomedici.com