MEDICI™

The State of Data Platforms in Financial Services

Background

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.

There is 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.

Financial institutions are increasingly looking to move operational and delivery models from physical to digital platforms as more and more customers prefer these channels.

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.

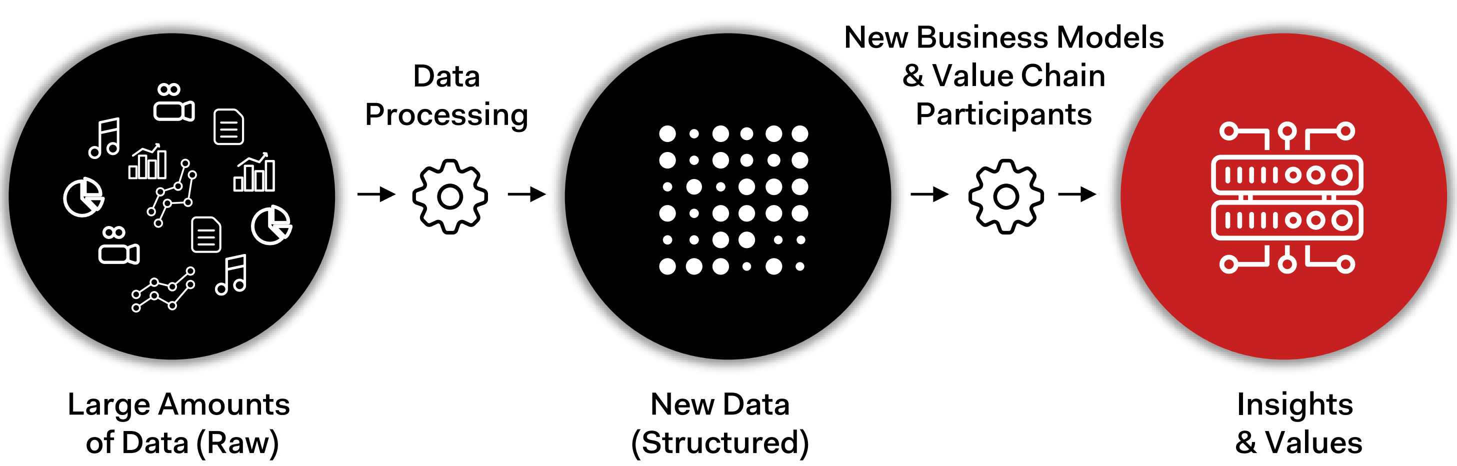
Along with better customer service, **real-time processing also enables for better risk management**. For regulatory purposes, financial institutions must measure and manage the riskiness of customer portfolios, requiring a move to real-time transaction processing for a large part of what a bank does.

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

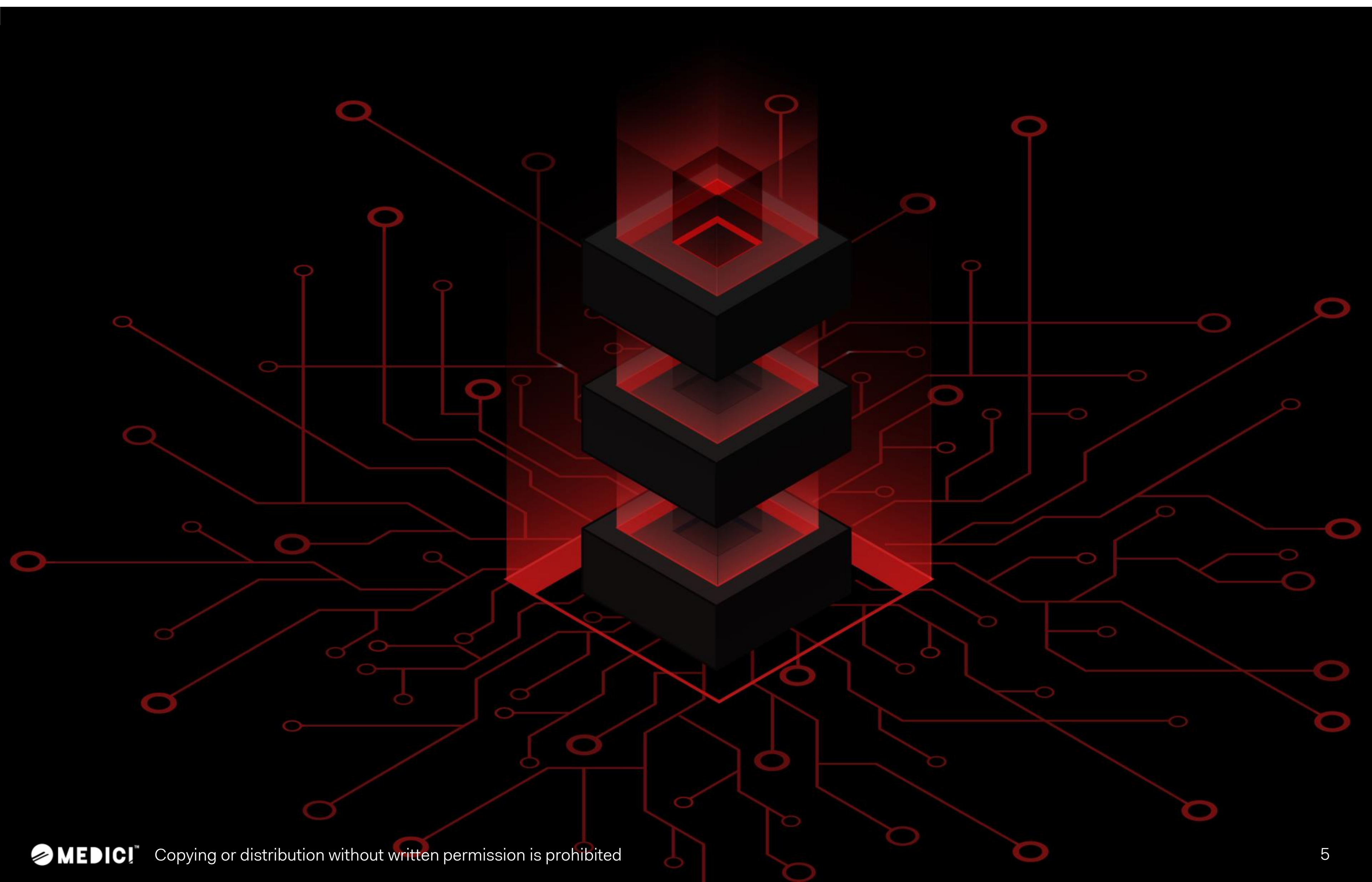
Financial services firms are **moving from batch processing to near real-time or real-time processing**. The faster settlement of trades at online traders has been a result of this shift. Customers attuned to the real-time nature of social media and other web-based services now expect the same in banking 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.



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

Data integrity for real-time data 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 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.

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.

Data Handling and Processing Limitations 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

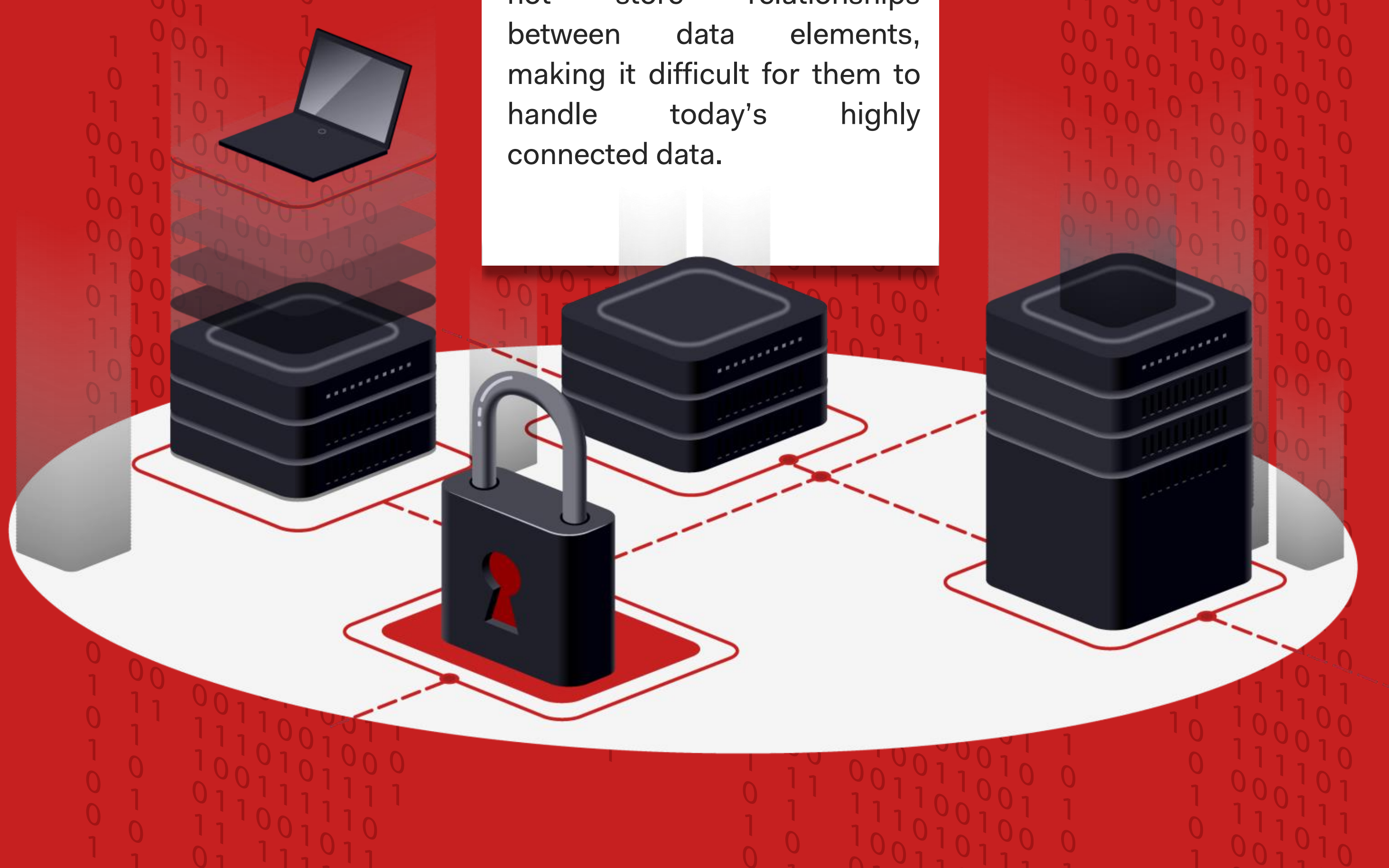
An RDBMS with table space stores files on SSDs. The RDBMS will still cache or buffer that data as it is a built-in function of the DBMS, and it was relevant when all data persistence was accomplished using disk storage. **Most SQL-based DBMSs are not designed to run in memory, and hence, this puts an 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**.



Conclusion

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. Institutions are looking 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. The industry is looking for robust data analytics for improvements in business performance and customer service.

Financial services have traditionally been innovators in technology. In recent times parts of 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.

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