Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Modernizing Monoliths: Transitioning Legacy Financial Systems to MicroServices

Pradeep Rao Vennamaneni
Senior Data Engineer - Lead, Citi Bank, USA
Email: praovennamaneni@gmail.com

ARTICLE INFO

Received: 10 Mar 2025

Revised: 26 Apr 2025

Accepted: 07 May 2025

ABSTRACT

Financial institutions, being the relevant banking industry players, must respond to the development in the banking sector toward an increasingly data-driven and fast-paced environment with the transition from legacy monolithic systems to the microservices model. In today's financial environment, where times are real and where data is processed without delay and in a cost-effective fashion, legacy systems are not flexible enough and monolith-driven; they are inadequate. One solution it provides is to divide a complex system into a series of independent services that can be scaled and updated without affecting the other parts of the system, thus providing flexibility and efficiency. This study analyses financial institutions' difficulties with upgrading their legacy systems using technologies such as Apache Kafka and Apache Spark and where to put the data in real time. It also involves integrating AI-driven microservices, notably generative AI, to automate customer service, improve risk management, and improve predictive analytics. Microservices are compared to the inefficiencies of a monolithic system, and the benefits of the microservices are discussed, namely, modularity, scalability, fault tolerance, and security. The other aspect of the paper also explores the best practices for implementing microservices, like choosing the right technologies, robust security measures, and proper management of the resources while migrating to them. This study aims to put forward a roadmap for the potential modernization of systems in financial institutions to stay competitive and react to the market dynamic through detailed analysis of microservices architecture and case studies.

Keywords: Microservices Architecture, Real-Time Data Processing, AI and Machine Learning, Kafka and Spark, Cloud-Native Infrastructure.

1. INTRODUCTION

The legacy systems of the modern financial world are predominantly based on the traditional monolithic system architecture and have become inadequate in the modern financial landscape. The solutions these systems once represented as a means of managing the complex needs of financial institutions are now perceived as impediments to progress. Since systems fail to keep pace, legacy systems cannot satisfy the rapidly growing needs of today's real operating financial markets with high throughput, speed, security, and flexibility. With growing expectations of financial institutions to advance and transform to compete in an ever more demanding environment of agility, data, and customer-centric decisions, modernization of the whole process has never been more urgent. In legacy financial systems, the architectures are monolithic, with large unified systems containing all their components as a single large application deployed monolithically. While they carried the day in earlier eras, these systems today are a problem in a technological and business context. Scaling legacy systems to cope with the growing volumes of data and transactions in current financial markets is an exceedingly hard challenge. It is a challenging task, as financial services systems are demanding systems that can handle variable demand without degrading performance.

Monolithic systems are quite inelastic. Updating or adding new features requires a new system; each update takes much longer and is very costly. It inhibits financial institutions from adjusting quickly to market developments, regulatory changes and customer demands. Moreover, security concerns have come into play as they have not disregarded legacy systems. The new, more complex cyber threats cannot be provided with the security necessary to

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

protect sensitive financial data under an old legacy system of systems protection. There are other concerns, such as increased maintenance and support costs. Several organizations struggle to stave off the growing cost of working with legacy systems as older technologies in these legacy systems are challenging to support. In some instances, institutions hire experts in legacy systems, and thus, a lack of resources is there when they hire; this only complicates the maintenance of the system. The disadvantages of monoliths could be overcome by using the microservices architecture. This system divides the complex system over its several independent services that work over their processes. This provides more freedom and flexibility and can be scaled up or down independently and independently per service. This change is applied to control the increasing complexity of current commercial banks in modern financial operations.

Microservices make it possible to achieve real-time data processing requirements for any financial service. In such a speedily changing market condition, financial institutions must be able to process transactions and analyze data in real-time to make the right decision. Microservices help scale efficiently with increased workloads. Microservices also allow for modularity, in which new features and services can be seamlessly doubled without disrupting the complete machine and introducing novelties. Each service has more separate security than a monolithic service, greatly increasing the chances of removing the surface of attack for any possible breach. In financial systems, customer data protection is most important to the organization, and this isolation is particularly needed because customers' data is so sensitive. Isolation and blocking off any security risks of any specific server makes it easier to secure the overall system by adopting microservices.

Time is of the essence in the modern financial system, especially in the era of real-time information-based decision-making, and real-time financial data processing is mandatory to an extent. Financial institutions rely on real-time data for tasks such as trading and investment analysis, fraud detection, and customer relationship management. Technologies such as Apache Kafka and Apache Spark enable real-time data processing in financial systems. Financial institutions relying on their Kafka and distributed streaming platforms can process an incredibly large amount of data in real time, meaning the necessary data needs to be delivered to all involved services quickly and efficiently. On the other hand, Spark provides powerful big data processing functionalities, making it a useful tool for processing and deriving insights from complex financial data in real-time. AI and large language models (LLMs) integration has been carried over to these microservices to enhance the functionality of the financial systems. Generative AI can automate customer service, risk assessment, portfolio management, and useful insights to give you predictive analytics. Implementing AI in financial institutions' microservices architecture can help make the financial systems smarter and deal with the changes in the market dynamically by taking real-time data as a basis and learning from it. Real-time processing and AI-driven microservices allow financial organizations to maintain competitiveness in an ever-data-driven environment.

This study seeks to delve into transforming traditional legacy financial systems to microservices-based architectures and the benefits, challenges, and skills that led to such a change. In the paper, the consequences of archiving and the limitations of legacy systems will be discussed, as well as the tendency to overcome these deficiencies through microservices architecture and the integration of Kafka and Spark for real-time data processing. Furthermore, AI-driven microservices, particularly generative AI models, will be explored in relation to their role in extending financial systems' capabilities. In this study, researchers will start by talking in depth about the challenges of legacy systems in the financial sector and proceed to the move to the microservices and the benefits therein. An analysis will be made on integrating real-time data processing with AI, followed by case studies and best practices for implementing the same. The goal is to give financial institutions a systematic plan to defeat modernizing their systems and compete in an ever-changing environment.

2. MICROSERVICES ARCHITECTURE: KEY COMPONENTS AND BENEFITS

2.1. Overview of Microservices Architecture

An architecture, also called a Microservice (MS), is a collection of loosely coupled, independently deployable services that can be scaled out or up as needed. Each service is tied to a specific business function or process and does not do that with any other services. This is the opposite of what is done with traditional monolithic architectures, where all the components are deeply intertwined into one code base. Microservices have each service running in its process

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

and communicating with other services using lightweight protocols such as HTTP, REST, or messaging systems (Salah et al., 2016). Microservices architecture makes it possible to decouple services so that teams can work independently on different parts of the application, and it helps improve efficiency and remove several dependencies. This allows us to focus on each service on one specific functionality, which is great as it helps update, maintain, and scale, particularly within complex areas like financial systems.

For example, in the case of the financial application, a first microservice can be responsible for the authentication of the user, a second one for processing a transaction, and a third one for producing a report. Each microservice is yet divided here and will be scaled individually based on resource requirements. For migration from a monolithic system to a microservice system to be successful, it is important to get a clear idea of how each service should be understood in terms of its boundaries, as an incorrect understanding of the boundaries may create integration issues and inefficiencies (Chavan, 2022).

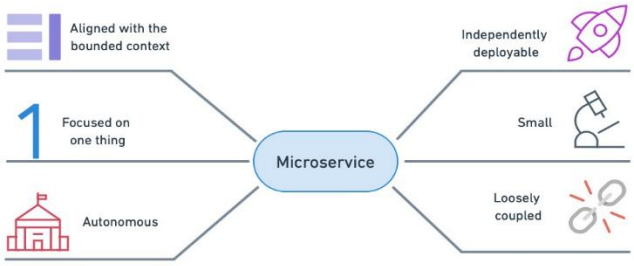

Figure 1: ***Microservice Architecture***

2.2. Benefits for Financial Institutions

In the context of processing real-time financial data and having real-time systems in place, microservices architecture has some significant advantages for financial institutions. Scalability is one of the advantages of microservices. In a usual monolithic architecture, researchers scale the entire app by replicating the entire system, which is expensive and inefficient. With microservices, microservices could be scaled based on the demand. For instance, while the load on the transaction processing service can accommodate additional quantities during peak times, other services, such as reporting, are unaffected. This scalability guarantees that financial institutions can handle and control transaction volumes that are subject to change well, which is why such performance is a must in the realm of the real-time financial environment (Allioui & Mourdi, 2023).

One other interesting benefit of fault tolerance. By definition, microservices are more resilient since the failure of one service does not necessarily crash the entire system. If a single service is missing, other services will not experience downtime. The financial industry depends on compliance, uptime and system availability to meet customer trust and regulation requirements. The isolation of failures and operation control without affecting any other is made possible using microservices, which provide better reliability for financial applications. Another distinction of microservices architecture is that updating and maintenance are easier. Suppose a component is updated in the monolithic system. In that case, researchers have to rebuild the whole application and redeploy it, which causes downtime, and a system that was not working before can now take a long time to start working. Some microservices allow developers to update the system without affecting the whole system. This allows development cycles to be accelerated and testing and debugging to be simplified.

Another major advantage of cost efficiency is that microservices help financial institutions utilize their resources according to best practices. They shift from spending resources by running a monolithic application to dynamically allocating the resources for the needed services. This approach reduces operational costs, especially in a cloud environment where metered resources can scale up or down as necessary. The significant factor driving the modernization of financial systems is real-time processing. Microservices allow real-time data to pass through services via an event-driven architecture. It becomes essential when downstream applications, such as high-frequency trading systems or fraud detection systems, need immediate processing of the incoming data to make timely decisions. Processing data in real time offers a competitive advantage in dynamic markets (Raju, 2017).

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

2.3. Microservices vs. Monolithic Architecture

Microservices and monolithic architecture should be compared with several important differences regarding the benefits and challenges of each approach. All the application components are tightly integrated into a single unit. When the system needs scaling or updates, the entire application must be worked with. The architecture of the microservices is also more flexible and modular, where the services can be developed, deployed, and scaled separately. This modularity makes it easier for financial institutions to respond to changes in market demands. Microservices are more agile. They give you more speed as you can work on different services inside the same team without waiting for another team's tasks to be done. That is especially true in the financial world, where the implementation speed will greatly contribute to competitiveness. Microservices also make it easier and quicker to test and deploy, as if the change were to just one service and the rest of the system is untouchable.

In a monolithic architecture, performance can be problematic when the application is large. However, downtime can be significant because one failure in one component can lead to a total failure of the entire system. On the other hand, microservices prevent performance degradation through isolation of failures, and division of work in parallel allows for faster development of components, hence higher overall system performance. Though it may have a higher initial development cost because of the need for specialized tools and infrastructure, the long-term cost saving makes sense. This allows financial institutions to scale individual services at demand and infrastructure cost as they only pay for the resources they use.

The security provided by microservices is much more granular. Institutions can secure each service independently, giving institutions the ability to impose tailor-made security measures tailored to the sensitivity of the data handled by each service. It is much more challenging to do this in monolithic systems where a security breach in one component could disclose the entire application to the risk (Yarygina & Bagge, 2018). Switching to a microservices-driven model involves a lot of effort and planning. With the advantages of scalability, fault tolerance, and agility, microservices have earned themselves rightly as a compelling choice for finance institutions today.

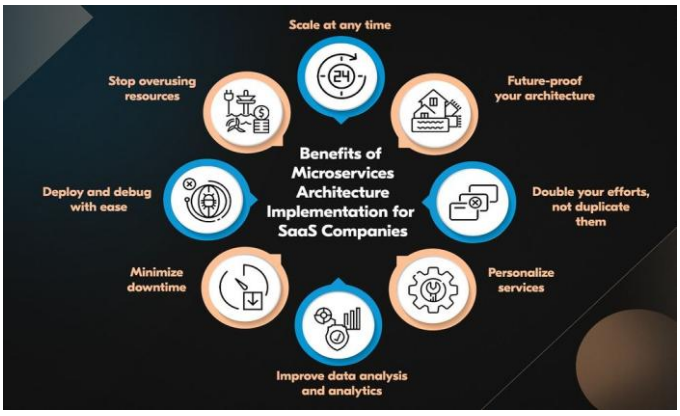

Figure 2: microservices-architecture-implementation-benefits-practices

3. REAL-TIME FINANCIAL AI ARCHITECTURES

3.1. Importance of Real-Time Financial Data Processing

In the modern-day financial sector, a critical approach to gaining speed to insight is the ability to process and analyze data in a real-time environment. Financial data processing in real-time is the continual and actual analysis of financial transactions immediately. With this capability, institutions can respond promptly to any movement in the market and invest more and lose less. One such case where real-time data is important is in areas such as stock trading, portfolio management, and financial analytics, among others, as every second counts.

The price of stock can also fluctuate rapidly in stock trading. Real-time data allows institutions to quickly change their strategies, buy or sell stocks, and optimize their portfolios. Algorithms trading also needs real-time analytics since decisions are made from live data feeds. Moreover, portfolio management requires real-time data because it remains the key for firms to manage the optimal mix of assets under market fluctuations (Owoade et al., 2024). Real-

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

time processing is also a boon to risk management because logical decisions can be taken in a real-time manner to mitigate risk to the minimum possible extent without reacting to a risk situation. Financial organizations can use real-time data to ensure they make informed, accurate, and timely decisions that are in sync with current market forces. As a result, firms can achieve outcomes with such consequences as increased returns on investments and customer satisfaction by responding to changing needs or issues that occur.

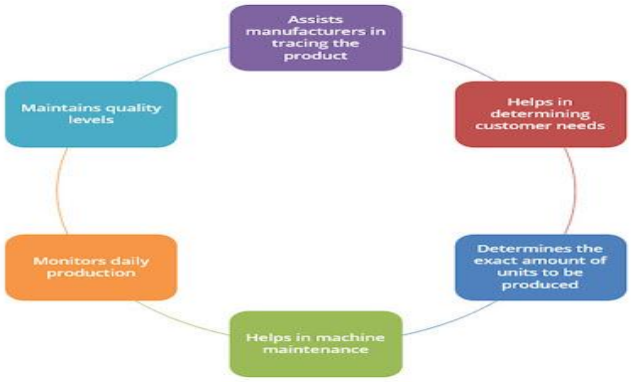

Figure 3: big-data-use-cases-in-the-manufacturing-industry

3.2. Kafka and Spark in Financial Systems

Kafka and Spark are two technologies actively used in the financial industry for real-time data processing. Korchitect is both a distributed and a real-time event streaming platform. With this tool, financial institutions can process millions of transactions and events per second and play an important role in real-time data piping. Kafka makes data highly reliable and quick to transmit, ensuring every financial transaction will be recorded and ready for immediate processing. For instance, Kafka is essential for streamlining the real-time price updates and transaction data to the traders who make real-time decisions based on the most current data in high-frequency trading. For example, Kafka's ability to scale horizontally allows for large spikes in the data volume without sacrificing performance to keep an overall good level of availability and low latency in financial systems.

Apache Spark is an open-source unified analytics engine capable of processing large amounts of real-time data. Its high-performance batch and stream processing makes it suited for financial analytics. Complex analytics on transactional data can be performed on, for example, real-time transactional data streamed from financial institutions into Spark. Provides powerful machine learning, data science, and statistics tools for predictive analytics and decision-making in the financial market (Boppiniti, 2019). Spark and Kafka have become a good pair for real-time financial transaction management. In another pipeline, Kafka streams the events, and Spark has to perform all the analytics and computations needed to discover insight from the data contained in the events. Such integration allows financial firms to process huge amounts of data very fast to provide analytical insights and make critical decisions.

Table 1: ***Key Technologies for Real-Time Financial Data Processing***

Technology	**Role in Financial Systems**	**Example Use Case**
Apache Kafka	Stream processing platform for real-time data	High-frequency trading and live transaction updates
Apache Spark	Real-time analytics for large datasets	Portfolio management and risk analytics
AI/ML Integration	Predictive analytics and decision automation	Automated trading, risk prediction, fraud detection

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

3.3. AI in Financial Microservices

Today, financial institutions use AI integrated into financial microservices. Predictive analytics, with the help of machine learning models in particular, can help more accurately forecasting and how to make decisions. For instance, this machine learning can be used to predict the stock market market, credit risk estimation and much more, to a certain extent, even financial advising. Since historical data are inputted into AI models and then 'find' patterns that humans may not see, financial institutions gain an advantage over their competitors during risk management and returns optimization.

Large language Models (LLMs) can give rise to systems capable of generating real-time data-driven recommendations on the fly and automating some of the decision-making processes involved in financial systems. Microservices based on AI may process large amounts of unstructured data, such as financial reports and news articles, which can lead to actionable insights.

Specifically, this ability is highly useful in real-time settings where decisions must be made quickly based on the most recent information. For instance, an AI-based financial system might look at the current market situations in real-time and change the investments automatically in a smart way of risk and return, considering the most recent data. On the same lines, LLMs are being used for fraud detection as they can quickly spot the anomalies and patterns that indicate the fraudulent action so that appropriate measures can be taken immediately (Ali, 2024). These chatbots and virtual assistants are often AI-powered to deliver personalized advice and assistance for a better customer experience.

Aside from routing processes more efficiently, AI's use in financial microservices also helps enhance the rate and accuracy of decision-making. Nevertheless, one must ensure that the above-said AI models are up and running in an updated and trained manner to keep their relevance and efficiency. It necessitates an investment in AI technologies that continue and a solid place where models of AI can be plugged into the financial system. Technologies such as Kafka, Spark, AI, and machine learning are changing the financial industry by supporting real-time financial data processing (Blessing et al., 2024). Finance institutions that adopt these advances should be able to make faster and more precise decisions, develop effective performance, and increase customer satisfaction. The latest was the introduction of the new generation of Generation AI and real-time data analytics, which are new innovation-driving forces helping to stay competitive and agile in an increasingly rapid universe.

4. SECURE, CLOUD-NATIVE DATA: ENABLING SECURE MICROSERVICES

For organizations that are transitioning from legacy systems to microservices, using cloud-native infrastructure is necessary to succeed in such a transformation. The financial data from the native cloud environment can be supported by a platform that provides scalability, flexibility and affordability within a secure environment.

Figure 4: cloud-native-architecture-for-digital-platform-development

4.1. The Role of Cloud-Native Infrastructure

Financial systems must be introduced to the cloud-native infrastructure to scale, be flexible, and be expensive. Microservices designed for hoarding functions in a cloud-native environment are then compared to the traditional monolithic systems, which can be easily scaled horizontally depending on the demand. It is super critical in the

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

financial sector, as huge volumes are being transacted, and they can go up and down depending on the dynamics of the market, new rules introduced or user behaviour.

The biggest strength of cloud-native infrastructure is what it allows you to implement; what researchers are discussing about here is elastic scaling, where resources like computing power, storage, and networking are allocated dynamically. It allows the financial institution to handle large and variable amounts of data by keeping resource provision parallel to the data and lowering costs. Cloud-native platforms also allow the deployment of services in multiple regions to enhance resilience and reduce latency for end-user clients. Cloud-native architectures offer cost-effectiveness because they are implemented using the pay-as-you-go models (Chatterjee, 2023). With financial institutions no longer required to spend money on expensive, on-premises hardware, which can quickly be outdated, logging into a remote computer becomes as easy as it used to be. Instead, they may deploy cloud services provided by cloud service providers such as AWS, Google Cloud, or Azure, which must have high availability and robust security features. The move to the cloud reduces the infrastructure overhead. It facilitates the better and more efficient allocation of financial resources, with concomitant cost and operational savings in the longer term.

4.2. Security in Financial Data Processing

Handling data security in cloud-native financial services cannot be overstated. Generally, the General Data Protection Regulation (GDPR) and Payment Card Industry Data Security Standard (PCI DSS) are strict rules on financial institutions. For cyber-attacks, sensitive customer data that include personally identifiable information (PII) and financial transactions are the prime targets. Therefore, it necessitates protecting the above-mentioned data. The other benefits include encryption, identification and access management (IAM), security, and advanced monitoring. For example, end-to-end encryption is a supported cloud service that encrypts data at rest and in transit to prevent unauthorized access or tampering (Nabeel, 2017). In addition, financial institutions can increase the security of data by using dedicated hardware security modules (HSMs) to manage keys and store those keys in secure hardware rather than in software.

Another security feature it provides is the ability to build over robust IAM protocols. This allows organizations to create fine-grained policies for data that has to be accessed or critical operations that only authorized personnel can access. It lessens the likelihood of insider threats and increases security positions in the organization. Continuous monitoring and automated alerts are also important for protecting financial data. Cloud-native platforms can monitor real-time traffic, user activity, and system behavior. The system can trigger automatic responses by locking down the affected services or alerting the security teams if any anomaly or unauthorized access attempts are detected. Advanced security features have become necessary, ensuring compliance with regulations and financial data security against changing threats (Nyati, 2018).

Figure 5: cyber-shockwave-exposing-the-major-finance-industry-breaches

4.3. Cloud-Native Data Management

When implementing cloud-native microservices for financial systems, data management is very effective. Financial institutions handle vast volumes of structured and unstructured data, and wrapping the whole process around security, availability, and consistency is difficult. Cloud-native platforms provide several tools and techniques for managing financial data securely and efficiently. Including distributed databases and data lakes as key factors in

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

cloud-native data management is crucial. The main benefit of these systems is that a large volume of data can be stored in different nodes by financial institutions so that they can be accessible and resilient (Dupont, 2019). For a real-time solution, these databases are made for cloud-native environments to handle high transaction loads and achieve low-latency access. Data lakes also allow storing raw and processed data, running advanced analytics, and supporting insights from advanced analytics and AI.

Among other things, cloud-native data management requires control management. The tool's strength is the support of cloud-native environments to implement stiff access controls like role-based access controls (RBAC) and attribute-based access controls (ABAC). Such control also secures the financial strength of the ecosystem through available financial data and only allows access to already active users and systems. Another help cloud-native systems provide is the automation of data backup and restoration if disaster strikes. This allows financial institutions to ensure that data are regularly backed up and stored in far-ranging locations. This will reduce the possibility of program loss in system failures, natural disasters, or cataclysmic events because it reduces the chance of data loss and then business continuity.

This is a vital capability for combining secure data services with a real-time financial system. These systems facilitate those microservices to easily communicate with one another, and the data is declared secure and efficient. One example of such parallel processes is financial transactions, customer data, and market feeds that several microservices can process so that financial institutions can quickly process changes and requests from the market to customers, thus increasing data security and reliability. To enable microservices in current financial systems, secure and cloud-native data management is crucial (Raj et al., 2022). With the cloud synthetic in place, financial institutions can do a good job of scaling out their systems, securing their data, and making it much easier to manage their data. It is predicted that the more the financial sector adapts and keeps up with all the changes that are bound to take place over time, the more cloud-native technologies will continue to be key drivers of innovation, compliance, and security within the financial services space.

5. TRANSITIONING FROM LEGACY SYSTEMS: KEY CONSIDERATIONS AND STRATEGIES

5.1. Planning the Transition

The transition must be phased to avoid the disruptions caused by migrating legacy financial systems into microservices. Organizations must delve into the core objectives and divide migration into doable stages, with every stage well planned and executed. A systematic approach provides the advantage of a more organized and less erratic transition. Determining which systems should be migrated first is one of the most important items to consider when planning the migration. Generally, legacy financial systems are large and intricate, so one cannot migrate it all at once. As a result, which systems can be modernized can have a huge impact on the success of the overall migration. One common practice is, to begin with unimportant systems that will not cause serious damage to regular money-related exercises if they are held off for brief timeframes or experience downtime. It reduces the risk of operations interruptions during a nascent phase.

Dependencies necessary between systems need to be considered. Legacy components usually have direct connections, and some depend on one another to exchange or process data. As organizations transition to microservices, interdependencies of legacy systems must be assessed, which requires an organization to develop an integrated plan for managing these relationships (Wolfart et al., 2021). Organizations will be able to address dependencies earlier on, lessen their chances of integration problems, and ensure that the microservices ecosystem is robust and works effectively. As an illustration, known project goals and well-defined phases increase the likelihood of successful transitions, especially in sectors with high financial consequences of system downtime, such as finance (Karwa, 2024). A clear roadmap, backed up by experienced project management, is very important to guide the process.

Table 2: ***Challenges and Solutions in Transitioning from Legacy Systems to Microservices***

Challenge	Description	Solution
Integration with Legacy Data	Legacy systems often have tightly coupled data structures, making migration challenging.	Dual-sourcing strategy with event-driven architecture

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Challenge	Description	Solution
Resistance to Change	Employees may resist transitioning to new technologies and architectures.	Communication, training, and fostering a collaborative culture
Cost of Migration	The initial investment in infrastructure, resources, and training is high.	Phased migration with proper resource allocation and cloud-based solutions

5.2. Dealing with Resistance to Change

The migration from legacy systems to microservices is no exception, and organizational resistance to change is a common challenge during any transition. When existing, established technology is more familiar than new technology, employees and leadership may view changes in the technology as a means of disrupting business normalcy, thus being hesitant to embrace new technologies. This resistance has to be overcome to promote the smooth adoption of microservices. Effective communication is one of the main ways of managing resistance. The benefits of the transition should be articulated by leaders (such as system scalability, decreased costs, and faster innovation). Presenting microservices as a long-term investment in the company's success reduces fears and shows how the change will benefit the company.

Another aspect to be strengthened in order to overcome resistance is training and upskilling teams. In this case, microservices are quite involved (the technology stack they use, Kafka, Spark, and cloud-native tools) and require specialized knowledge and skill sets that all your team members need. Continuous learning opportunities provided to employees will keep them updated on the work with fresh architectures and technologies (Engelmann & Schwabe, 2018). Ensuring that technical teams know the right tools to assist in working microservices, such as MongoDB, for real-time data is also critical. In addition, organizations should appreciate creating a culture of collaboration and innovation, which is perceived as the best defense against change resistance. By soliciting their feedback and addressing issues, organizations can bring their employees along for the ride, as it were, from the get-go. It increases employees' investment in the transition and encourages them to embrace the changes.

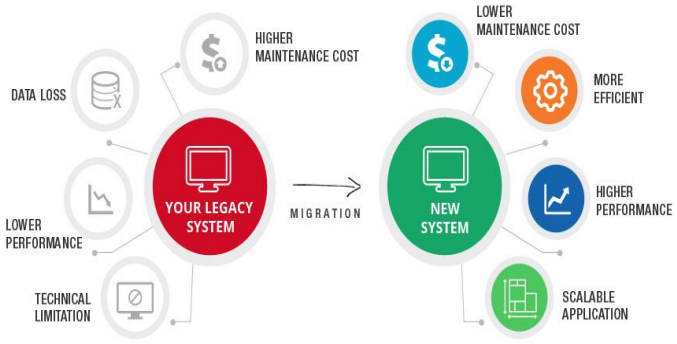

Figure 6: ***Common application migration strategies***

5.3. Cost and Resource Allocation

Transitioning from legacy systems to microservices entails a considerable investment in funds and, particularly, time. The amount of money the financial institutions will need to allocate for infrastructure, software training, and human resources to implement the microservices architecture successfully is substantial. Migration to microservices includes the initial capital costs and additional ongoing maintenance and optimization of the new systems, which will concentrate some of the overall costs.

Infrastructure is needed to support the new microservices ecosystem, which is one of the primary costs. Usually, cloud platforms opt for hosting microservices because they are scalable, flexible, and cost-efficient. Nevertheless, the planning should be done carefully to allocate the needed resources for processing power, storage, and network capacity in the case of cloud solutions, too. By adopting practical solutions like MongoDB to allow them to manage

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

their big data in real time, the server infrastructure will adapt according to its scale without burning the pockets (Dhanagari, 2024).

Another critical aspect of the migration process was human resource allocation. The technical side of the migration has to be conducted, and the microservices integration smoothly should be done based on the technical aspects of the organization. Some of these resources are cloud architects, DevOps engineers, and data scientists with prior experience working with Kafka and Spark technologies, which are fundamental to enabling the real-time processing of financial data. Long-term resource commitments will increase due to the aspects indicated earlier, such as the need for continuous training. In addition, it is also necessary to consider the costs of managing hybrid infrastructure in the transition period. Microservices are being deployed, and treating legacy systems to continue to run in parallel will result in a hybrid complex architecture. Management of this is careful and involved to minimize redundancy and to ensure that the systems talk to one another without much overhead.

Planning the transition from legacy systems to microservices in financial institutions is complex. Coordinating the resistance requires effort, and the process requires much resource investment. While the upfront needs and the associated organization challenges may seem intimidating, the fact that paying for these in advance must be done with money tied up means organizations do not make these investments without reason. The need to migrate to microservices IS a strategic decision for any financial institution that wishes to remain competitive in the modern digital age (Mazzara et al., 2018). The first is to tackle the problem using a phased approach, in which organizations first identify and gain some agreements on common priorities, empower employees and invest in their learning and development, and properly allocate resources.

6. INTEGRATION CHALLENGES AND SOLUTIONS

In particular, legacy financial systems are integrated into a microservices architecture but with great complexity. This arises from the differences in data structures, system dependencies, and performance requirements.

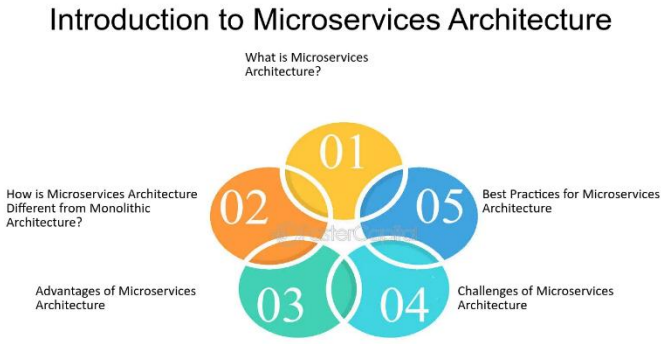

Figure 7: ***Implementing Microservices Architecture to Enhance Legacy Systems***

6.1. Data Integration in Microservices Architecture

Integration of legacy data sources is one of the most prominent challenges when moving to a microservices architecture. Generally, legacy systems base their databases on monolithic databases that are very coupled to the application logic. On the other hand, with a microphone that uses decentralized databases, each service handles its data. The implication is that legacy data integration with microservices should be done with due care and be free of any damage to data integrity.

The first data integration approach uses dual-sourcing strategies to enable legacy systems to coexist with new microservices. This strategy merges microservices with legacy databases by letting the two systems operate together. The persistence systems also remain active until the full transition, and these services can consume and manipulate data using microservices as new services are enacted. The advantage of this approach is that it minimizes the disruption to business operations while phasing the migration (Goel & Bhramhabhatt, 2024).

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The other critical aspect is maintaining data consistency throughout the old and new systems. Since the microservices are normally meant to be run in a distributed manner, this makes it difficult to keep the different services and legacy databases consistent. An approach is to design an event-driven architecture where, based on events, they ensure that the microservices and legacy system stay in synchronization. For instance, the events can notify other services or databases to keep other systems' data consistent and up to date. There can also be latency issues when integrating a legacy system into a real-time processing system. The main advantage of Microservices is that data retrieval and processing are usually fast, but this is not always the case with other legacy systems that were not built to be fast (Wang et al., 2021). For this reason, companies use asynchronous processing and caching mechanisms to ensure that real-time data processing retains efficiency when working with slower legacy systems.

6.2. Cross-System Communication

The integration process is integral to ensuring the flow of communication between microservices and legacy systems. API management plays an important role in stitching services together in a microservice pattern because services within a microservice pattern often rely on API management. Simultaneously, legacy systems may not be able to run according to modern API standards or have basic changes to make public APIs. To support this, the API gateways of the organizations need to be strong so that they can be managed with microservices and traditional systems. Through these gateways, routing, and error resilience in the case of error are all handled, keeping the line of communication open between systems and secure.

Service orchestration is needed to coordinate the operations when cooperating with multiple services. Orchestration is necessary to gain control over interactions between services and to ensure that workflows are executed as required, much less restrictive than legacy systems without modularity. Campaigns to process data in and out of microservices and legacy systems are orchestrated by tools such as Apache Camel or Kubernetes (Koya, 2024). These tools ensure that cross-system workflows are effective, that services can communicate with each other, and that data can 'flow' across systems. Another strategy to promote communication through events is to adopt an event-driven architecture across systems. Event communication is asynchronous. Hence, it increases responsiveness and optimizes the data and system processes. Events are emitted by Legacy systems when microservices update, capture and process data. This approach separates systems to make the microservices and legacy systems more flexible and efficient in communication.

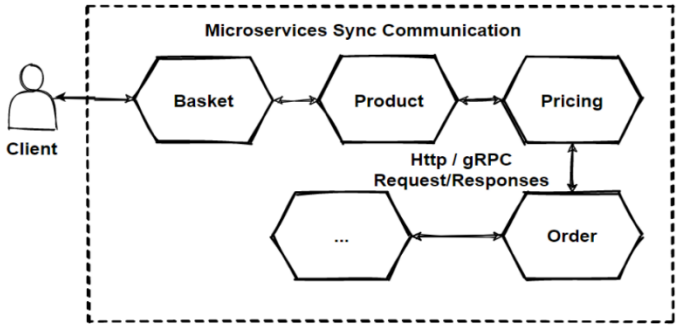

Figure 8: ***Microservices Communications***

6.3. Ensuring Performance at Scale

In addition to their operational costs, the performance of financial systems managing the several million transaction volumes in real-time at scale needs to be achieved. Another strategy to promote communication through events is to adopt an event-driven architecture across systems. Event communication is asynchronous. Hence, it increases responsiveness and optimizes the data and system processes. Events are emitted by Legacy systems when microservices update, capture and process data. This approach separates systems to make the microservices and legacy systems more flexible and efficient in communication. Performance optimizations are one of the largest examples of load balancing. In microservices, it is necessary to ensure that traffic is routed to multiple service instances so that no traffic can get blocked at a single service. Dynamic load balance techniques can balance the number of requests to all services in the pool, such as the round robin, and least connections (Kaur et al., 2015).

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Autoscaling is a good choice for companies with precise performance needs and who need to change the number of service instances based on traffic demand.

An important factor for further performance optimization scaling. Services are designed to scale independently; thus, a service can be scaled individually based on its needs. An example would be a transaction processing service requiring more power during peak hours while the reporting service scales only for some hours. Horizontal scaling spins up new instances of services as the load increases without changing their performance, thus enabling financial systems to achieve high transaction throughput.

For such sensitive financial systems as those that process large amounts of transactions, errors must be prevented so they can be operational. The service should be in a state where one failure does not require the whole system to crash. Patching these types of system failures can be addressed using techniques such as circuit breakers, retry logic, and data replication, which will prevent services from failing. These strategies evolved and are retained because the financial industry is prone to system downtime due to exorbitant financial losses in finance. Data integration, communication, and performance issues arise when legacy systems are integrated into a microservices architecture (Ogunwole et al., 2023). This also enables organizations to shift to microservices with performance and reliability assurances necessary for operating systems with real-time financial requirements in a dual-sourcing strategy using event-driven architectures, scale, and fault tolerance.

7. THE ROLE OF GENERATIVE AI MICROSERVICES IN FINANCIAL SYSTEMS

7.1. Introduction to Generative AI in Finance

Large language models (LLMs) or generative AI are extracting disruptiveness across the financial services automation, decision making, and the most important part – customer service. These AI-created microservices mean complicated tasks can now be automated, and these services handle the large amount of data involved. Financial organizations can overcome their main challenges by adopting Generative AI, which includes real-time decision-making, risk management, and customer service.

It also has quite wide results on the applicability of generative AI to many aspects of financial operations. Customer service automation is one of the main reasons AI adoption is happening; virtual assistants resolve everything related to customer support and chatbots that run on LLMs for around 24 hours. These are systems that AI drives to help streamline support, bring down operational costs, and increase responsiveness. In fraud detection, AI is trained to identify normal transaction behavior, and its existence allows it to analyze the transaction data in real time and to give you suspicious activity flag alerts (Khurana, 2020). In addition, AI-powered predictive models have made risk assessment more accurate by simulating different financial scenarios, giving better insights into investment strategies, and reducing possible losses. Another must-have generative AI application for delivering personalized financial advice takes customer data to help them prepare personalized financial strategies or investment portfolios to satisfy the customer.

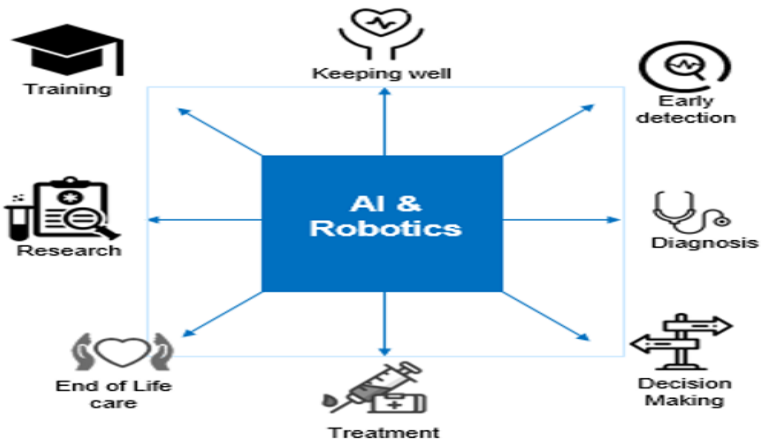

Figure 9: Decoding the genesis of Hyperautomation

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

7.2. How Generative AI Integrates with Microservices

The generative AI microservices are holistic and will integrate into the other microservices in the financial ecosystem to build a robust architecture. Experts can use APIs and event-driven architecture to instrumentalize the communication between the AI microservices and the other system components to flow rather smoothly. Generative AI is increasingly used in the real-time processing of financial systems, as generative AI can generate dynamic outputs that use up-to-date financial information.

Integrating machine learning pipelines further increases these micro services' adaptability and AI. By forming these pipelines, the AI models can learn continuously, which means that the learning models are doing their best to refine their predictions and decisions as they find out more data. To illustrate, suppose a customer wants financial advice; an AI model would process the data inputs, and a personalized recommendation would be generated. This recommendation can be passed on to portfolio management systems or even trigger a transaction to provide a smooth end-to-end process because this can integrate with other financial microservices (Cristofaro, 2023). The financial institution now has at its disposal a highly responsive, efficient infrastructure that is put together by combining real-time data, event-driven microservices, and machine learning models.

Table 3: ***Comparison of AI Models in Financial Applications***

AI Model	Application in Finance	Strengths	Limitations
Generative AI (LLMs)	Automated customer support, real-time decision-making	Can handle large amounts of unstructured data	High resource requirements, data biases
Machine Learning (ML)	Predictive analytics, risk management, fraud detection	Can make accurate predictions with historical data	Requires large datasets and continuous updates
Deep Learning	Algorithmic trading, fraud detection	Exceptional performance in pattern recognition	Computationally intensive, overfitting risk

7.3. Business Benefits and Challenges

There are several business advantages to integrating generative AI microservices into existing financial systems. One of the biggest advantages is increased automation, which makes processes less reliant on human intervention and sharper over-system efficiency. Using AI-driven capabilities to make decisions makes the predictive power more accurate and improves financial outcomes. Processes such as loan appraisal or fraud detection can be automated, and costs and errors can be drastically reduced among many financial institutions.

The other major benefit is improved customer experience. It allows institutions to personalize financial services by leveraging AI to give highly relevant advice or product recommendations because it positively impacts customer satisfaction and retention. At the same time, this implementation does have some challenges. One of the major concerns is the accuracy of the model. Data quality is important for financial institutions because AI models are only as good as the data they are trained on, and the models must be updated often. Prediction is costly, with severe consequences, such as an improper investment strategy or inaccurate risk assessment.

It is also important to manage the bias of the AI. Biases in historical data could unintentionally be passed on to AI systems, resulting in discrimination in areas such as credit scoring, approval of loans, and so on. To avoid such risks, financial institutions must ensure that their AI models are fair, transparent, and meet all regulatory standards (Lee, 2020). In their pursuit of regulatory compliance, financial institutions must ensure that their AI solutions comply with industry regulations, such as the General Data Protection Regulation (GDPR) or the Dodd-Frank Act (Sardana, 2022), if they do not wish to face legal repercussions.

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

7.4. Real-World Use Cases of Generative AI in Finance

Generative AI has been integrated into many financial institutions to improve their services. Not much data is used in traditional credit scoring, such as historical credit history. Generative AI, however, will let users put much more data to use, such as real-time transactions versus historical data. With it, financial institutions better determine creditworthiness even for an individual without a past credit history. A credit score that keeps changing is based on up-to-the-minute data, thus having a dynamic and more accurate picture of a person's financial health. As a result, generative AIs are employed in algorithmic trading to predict market trends, make trading more advantageous, and serve other similar purposes. In the second case, AI models are explored to allow one to trade based on historical data, market properties, and current conditions. Furthermore, the use of Generative AI was useful in Risk modelling. It provides a large range of money situations to enable the institutions to price risks and create counter-risk strategies (Xu, 2018). Built on this AI-infused insight, integrating AI-driven insights with legacy financial systems gives it an unprecedented ability to perform more dynamic and complex tasks than ever before.

Integrating generative microservices in financial systems essentially shares major impacts on them, changing traditional financial infrastructures into more intelligent, responsive, and automated ones. Financial institutions can leverage AI's capabilities of real-time data processing, automation, and personalization to improve services, efficiency, and customer service. These technologies are already being applied in fields such as medical technology and finance, and their impact on the financial sector is expected to grow as these technologies mature.

Table 4: ***Real-World Use Cases of Generative AI in Finance***

Use Case	Description	Benefits
Fraud Detection	Detecting fraudulent activities in real-time	Faster detection of suspicious activities
Predictive Analytics	Forecasting market trends and investment risks	Better decision-making and risk mitigation
Customer Service	Automating customer support with AI-driven chatbots	Improved customer experience and satisfaction

8. SUCCESSFUL CASE STUDY: LEGACY SYSTEM MODERNIZATION IN A GLOBAL BANK

8.1. Background of the Legacy System

One of the largest global banks had its back core banking systems supported on a monolithic legacy architecture. These systems, which were decades old, were deeply embedded and supported basic financial services such as transactions, loan processing, customer data management, and account management. That outdated infrastructure became too much for the bank in the long run. This monolithic structure proved slow to scale and did not offer sufficient support in processing times and volumes (Lorignon et al., 2020). With the HTTP of the financial institution expanding its operations to other countries and dealing with modern commerce, such as real-time data processing, availability, and secure transaction handling, there was no doubt that the legacy system would not be able to bear up with the banking sector.

Another thing with the legacy system that was one of the most pressing issues was that it lacked flexibility. The system was continually updated, making it cumbersome, and downtime was not uncommon, causing major banking services to cease while they were working on it. Further, the architectural monolith-ness also caused havoc if even the slightest change had to be performed, as it flowed throughout the entire system. Security concerns were also on the bank's side as they could not apply security measures to the evolving security threats in the digital financial ecosystem because the legacy system was not designed to handle such threats. Since these limitations were understood, the bank decided that modernization was required to improve efficiency, scalability, and security. The intention was to transition towards a more flexible, scalable, and secure microservices architecture, which would use real-time data processing and AI-powered services to enhance the system's performance.

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

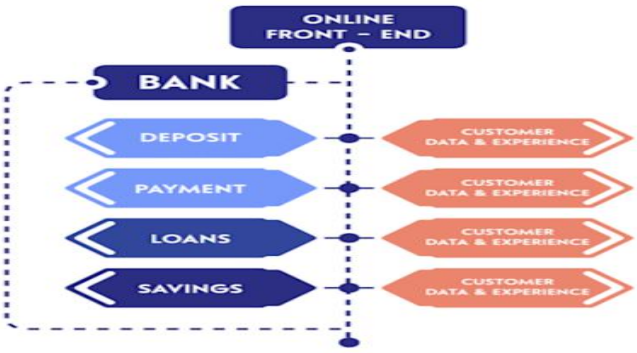

Figure 10: ***Core Banking Systems Primer***

8.2. Approach to Microservices Transition

The journey to a microservices architecture was not simply a technical affair but a well-thought-through company transition to a different architecture. The bank proceeded with a phased approach, starting with a review of its current infrastructure and key components that would benefit most from modernization. It was decided that the legacy system would be separated into a smaller independent microservice that could independently run and communicate over the API.

The bank used several modern tools and technologies to manage this transition. Stream processing was done using Apache Kafka, and the bank could efficiently handle various microservices' real-time data flows. Apache Spark was used to process real-time data analytics and integrate financial transactions, customer info, and services using AI-driven technology to deliver predictive analytics, fraud detection, and personalized customer experience. The system also used generative AI models to help streamline the work in customer service with automated responses and real-time advice, improving customer satisfaction nicely.

During the migration, security took a back seat for the bank. Security is added to the pipeline by putting security tools like Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Software Composition Analysis (SCA) tools within DevSecOps with the implementation of a process. By implementing this, any potential security vulnerabilities were uncovered, which is essential as early as possible and before going live. One of these was the critical proactive security strategy to ensure that the sensitive financial data was not compromised during the transition (Konneru, 2021). With microservices, the bank could independently decouple its systems to scale different services on demand. This flexibility enabled the bank to respond more quickly to operations during high-traffic periods, particularly during market openings or economic events.

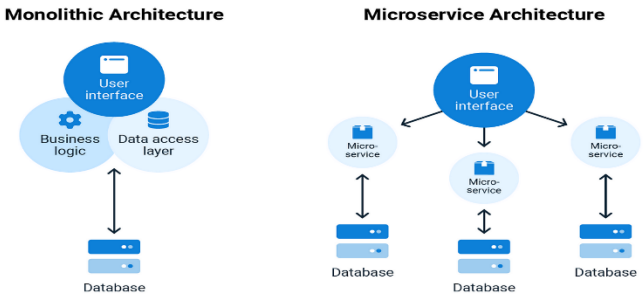

Figure 11: ***Microservices vs Monolithic Architecture***

8.3. Outcome and Key Learnings

Migration to a microservices-based architecture changed the results of it. Many other benefits aided the bank's scalability, like increasing transaction volumes without crashing the system or logging in for an hour due to overloaded servers. It also immediately increased the speed of operational efficiency as transaction processing time

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

was reduced from minutes to seconds during the transition to real-time data processing. Additionally, the use of AI-driven microservices by the bank allowed them to provide more personalized services to their customer and enhanced customer engagement and satisfaction.

The microservices architecture greatly reduced downtime, as updates and maintenance were easier. Unlike a monolithic system, where any part of the system's failure might affect the entire operation, Microservices ensured that the failure of one part of the system could be handled independently without disrupting the rest of the infrastructure—this increased system reliability and overall uptime.

The main lesson learned from this case study is that the change from legacy monolith systems to microservices can bring several wins regarding scalability, flexibility, and efficiency. The effort involved is complicated and involves picking the appropriate technologies so that security is a top priority. Suppose a similar transition is in the course for financial institutions. In that case, it is important that each microservice move in a phased manner where the services have to be integrated seamlessly with other services implemented, and real-time data processing with AI-based functionalities should be put to higher priority to meet current demands. Additionally, a robust security framework should be integrated into the CI/CD pipeline to protect against new threats in the financial sector (Owoade et al., 2024). This bank modernization is why microservices are necessary for financial institutions aiming to boost performance, security, custom, and experience while operating in a dynamic and always-evolving landscape.

9. BEST PRACTICES FOR IMPLEMENTING FINANCIAL MICROSERVICES

9.1 Choosing the Right Tools and Technologies

While implementing financial microservices, picking up suitable tools and technologies becomes key to large-scale performance and security. Apache Kafka is one of the central stream processing technologies that are used in many financial systems. Kafka is a robust and fault-tolerant way of dealing with high volumes of real-time data. Since financial applications require high throughput data and real-time transaction processing, it can be used for financial applications since it can manage the high throughput data. Kafka offers the decoupling of microservices and a convenient data flow across the system, facilitating agility in financial transactions. Large-scale data processing is dependent on Apache Spark. Spark allows us to quickly process huge amounts of real-time data, making financial services applications that need real-time analytics, like algorithmic trading or risk modelling, the perfect place to use this tool. By integrating well with Kafka, Spark provides a solid pipeline for managing and processing data streams at scale and the possibility of directly running a machine learning workload on streaming data.

More organizations opt for microservices architecture, and the cloud-native platforms are becoming their backbone. Financial institutions can pay only for resources used by cloud computing scaling systems. Further, because cloud environments are highly flexible and cost-efficient, the total load can be handled in financial systems in varying load situations that require high availability and fault tolerance. With managed services from major cloud providers like AWS, Microsoft Azure, and Google Cloud, financial organizations can focus on building, deploying, and scaling microservices to the least possible degree (Khan & Chandaka, 2021). In contrast, the cloud provider handles the infrastructure management. In addition, AI models are a key part of financial microservices. Financial institutions can automate decisions, detect fraud, and provide personalized services using machine learning algorithms in microservices. AI in microservices can include integrating AI into microservices that can help with real-time predictive analytics in risk assessment, fraud detection, and market prediction.

Table 5: ***Best Practices for Implementing Financial Microservices***

Best Practice	**Description**	**Example or Tool**
Choosing the Right Tools	Selecting suitable technologies for scalability	Apache Kafka, Apache Spark, MongoDB
Data Encryption	Encrypting sensitive data to prevent breaches	AES-256, TLS

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Best Practice	Description	Example or Tool
Continuous Monitoring	Regularly monitor system performance and health	Prometheus, Grafana, ELK Stack

9.2 Implementing Robust Security Measures

The ability to read and store sensitive data, such as customer information, financial records, and transaction information, is important to the business's customers, so financial microservices must be secured. The first thing to do while securing microservices is data encryption. Securing all data sent on the network requires protocols such as TLS (Transport Layer Security), and financial organizations must implement them. Sensitive data in a database or storage should be encrypted with advanced commands such as AES-256 to prevent intruder access (Ryandika & Prabowo, 2023). Another important practice in solid security is access control, which is very obvious because it is common knowledge. In the financial microservice, many microservices communicate with one another, and each microservice requires a natural and well-defined scope. Therefore, one should implement role-based access control (RBAC) to limit access to certain parts of the system. The granularity of access control controlling sensitive financial data ensures they do not share data with users and services that are not required to access the data other than for operation purposes.

For secure financial microservices, it is important to abide by compliant regulations, like PCI-DSS (Payment Card Industry Data Security Standard) and GDPR (General Data Protection Regulation). Some strict measures need to be taken per these regulations, such as data encryption, data access control, and controls' performance. Suppose a company employs an architecture based on microservices. In that case, it has to ensure that the architecture adheres to these standards – otherwise, there will be penalties, and customers will not trust the company. In addition, compliance as code practices can accelerate the regulatory compliance process by automating controls to safety and privacy standards across the microservice.

It is very important to maintain a secure microservices cycle. This should not depend on where the security attack occurs, whether at the perimeter or during the development and operational phases. To address the feasibility of early vulnerability identification and elimination in the stages of development, security practices such as DevSecOps, which combines security testing within the continuous integration and continuous deployment (CI/CD) pipeline, are implemented.

Table 6: ***Security Measures for Financial Microservices***

Security Measure	Description	Example Technology/Method
Data Encryption	Protecting data at rest and in transit	AES-256 encryption, TLS for data transport
Role-Based Access Control (RBAC)	Limiting access to services based on user roles	AWS IAM, Azure Active Directory
DevSecOps Integration	Incorporating security into the CI/CD pipeline	Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST)

9.3 Continuous Monitoring and Optimization

Continuous monitoring of financial microservices is important once you have deployed them because it is necessary to ensure system performance and that services are working as intended. Financial institutions can identify performance bottlenecks, errors, and potential security breaches so they can act quickly once they are known. Prometheus for metrics collection and Grafana for visualization give insight into maintaining optimal performance. Having the capacity to monitor the health of the microservices allows for proactive intervention to be remediated before being introduced to end users or financial operations. Monitoring is also a necessary (though not sufficient)

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

practice in addition to continuous optimization to ensure that the system stays efficient and scalable over time. Financial systems come with varying loads, and microservices must be built to scale horizontally and vertically. Financial institutions can use auto-scaling features in a cloud environment to ensure that the microservices scale according to demand and perform without overprovisioning resources (Kumar, 2019).

To achieve system observability, organizations should put in place practices, like distributed tracing, to follow the requests as they go through multiple microservices. There are tools like Jaeger and Zipkin that will give you visibility on the lifecycle of your request, which would help you identify any latency issues and optimize performance. Log aggregation tools such as ELK Stack (Elasticsearch, Logstash, and Kibana) help in easier troubleshooting and faster incident resolution (Şuştic et al., 2022). As part of the development pipeline, performance testing should be performed to optimize financial microservices further. Performance tools, such as Apache JMeter, can simulate high traffic volumes to test what happens to the microservices once they are hit so hard by real-time financial transactions. In order to implement financial microservices, adequate technologies need to be selected, a strong security layer should be put in place, and optimization should be a never-ending process. With the emergence of cutting-edge tools like Kafka, Spark and cloud-native technologies, and on the back of security and monitoring systems, financial institutions are successfully transitioning their systems to account for the needs of the digital economy. By following these best practices, these microservices architecture can be secure and scalable, high performing and, therefore, better services to their customer and staying in competition in an increasingly complicated financial landscape.

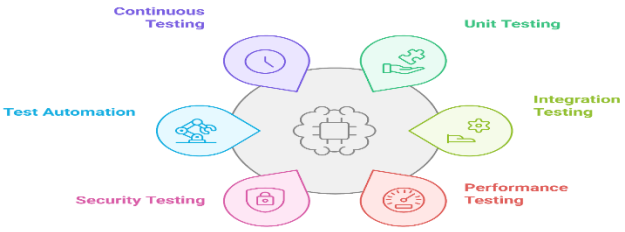

Figure 12: Technical challenges and solutions in AI Agent deployment

10. FUTURE TRENDS IN FINANCIAL MICROSERVICES AND AI

From technological advances, the landscape of financial services is evolving extremely fast. At financial institutions, which are still evolving to some form of modernization, microservices, artificial intelligence (AI), machine learning (ML), Blockchain, and cloud-native architectures are shaping the road ahead.

10.1 Advancements in AI and Machine Learning

Having already deployed in the financial sector, artificial intelligence and machine learning have much to offer but still have a long way to go. The point was made that, as the microservice architecture proves to be the cornerstone of financial applications, AI and ML must evolve and integrate within these microservices architecture very deeply. The most important advancement is that more and more companies are using large language models (LLMs) like GPT and BERT that can read and process tremendous amounts of textual data in full motion. Automating such decision-making, risk assessment, and customer service processes greatly speeds up the time in question and potentially lessens human resources utilization in these activities.

AI will continue to evolve through one of the major areas: predictive analysis. With more data collected by financial institutions via microservice, the patterns can be analyzed, and machine learning models can forecast future market trends. This predictive capability is especially key for financial institutions that deal with frequency trading, portfolio management and risk mitigation. Financial services will turn on predictive analytics in conjunction with machine learning, becoming a cornerstone decision-making process, allowing makey to make better and quicker decisions (Dhanagari, 2024).

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

More adoption of smart contract automation in the financial sector can also be expected. Self-executing contracts whose terms are designed in lines of code are called smart contracts. When financial institutions use microservices, these contracts will be AI-powered and phoned into the overall financial systems. It is anticipated that smart contracts would render intermediaries less necessary and facilitate financial transactions, including but not limited to insurance, securities trading, and cross-border payments. Decisions will also be automated and improved through deep learning. Deep learning models analyzing vast datasets will draw patterns that do not appear to exist using them. In real-time, these models will detect fraud, monitor credit scores, and monitor compliance.

10.2 The Rise of Blockchain and Decentralized Finance

Blockchain technology and decentralized finance (DeFi) as a source of financing for various projects, communities, and start-ups are revolutionizing the traditional financial ecosystem. It is widely known that Blockchain provides better security, transparency, and immutability, which plays a vital role in financial microservices. Financial institutions can improve their operations' efficiency, security, and traceability by combining Blockchain with the microservices architecture. The integration of Blockchain brings data immutability to an organization. Transparency, trust, and accountability in financial systems increase the level of trust from the users because no one can change or escape the transactions made in the Blockchain (Rane et al., 2023). For example, Blockchain's cons in real-time are that it proves the authenticity of transactions, decreases fraud, and increases confidence in the system. In addition, this technology facilitates faster and more secure transactions at financial institutions, for instance, in cross-border payments, asset tracking, and supply chain financing.

There is a challenge to the traditional finance models with the rise of DeFi, which operates outside centralized financial institutions. Decentralized finance platforms- peer-to-peer lending, decentralized exchange, and automated market makers- are breaking the traditional finance model of delivering and spending financial products. Integrating these decentralized platforms into financial institutions' networks can be facilitated by microservices, allowing financial institutions to innovate within financial ecosystems and simultaneously avoid losing security and compliance. The combination of Blockchain and microservices allows financial institutions to work more efficiently and transparently by entertaining the decentralized reality of Blockchain. This combination of a bank and an asset manager is a major driver of innovation in financial services.

10.3 The Future of Cloud-Native Financial Architectures

Because the cloud-native economy is ever-evolving, financial microservices will be forever altered. The architectures of cloud-native applications are highly scalable, flexible, and robust, traits that financial institutions desire. As such, serverless computing has become one of the major cloud-native systems trends, allowing financial services to scale resources automatically in response to demand with no manual intervention. By reducing operational costs, serverless platforms enable you to increase the efficiency and scalability of microservices-based systems. Another important trend is the rising popularity of edge computing, meaning that data is processed closer to the point where it is generated instead of solely dependent on central data centres. Using edge computing, financial institutions can process transactions, analyze data, and provide services in real-time and for remote and unserved areas. The other advantage is that it works well in high-frequency trading, fraud detection, and real-time customer interactions, where latency matters greatly.

Financial systems are also entering more and more into AI cloud platforms. Machine learning models improve financial microservices' performance, security, and predictability with these platforms. Financial institutions can deploy fast AI models for various tasks, such as risk management, fraud detection, and personalized customer services on cloud-based AI platforms. The architecture driven by cloud, AI, and machine learning will enable financial institutions to build agile, scalable, and secure financials that compute to the real-time market dynamics. Financial systems will be built in the future by integrating microservices, AI, Blockchain, and cloud-native technologies (Raj et al., 2022). These technologies help the rise of automation, efficiency, and security in financial transactions and enable institutions to create the required competitive environment. However, as these technologies mature, financial institutions will have the opportunity to capture clients' needs and other regulatory changes.

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

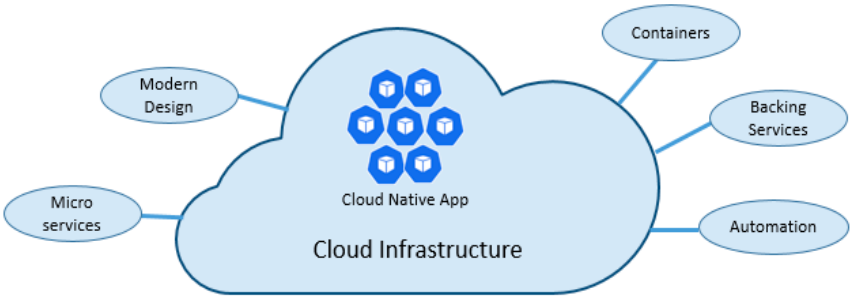

Figure 13: cloud-native-the-future-of-application-development

11. CONCLUSION

In an increasingly digital and very quick market, the financial sector has to transform from legacy monolithic systems to microservices for its own managed survival. In the past, these very successful traditional monolithic architectures have been used. Today, they present enormous difficulties with scaling up to meet the continually rising requirements of financial institutions: they are unscalable, very expensive to keep, and highly vulnerable to cyber threats. This solution is more flexible, scalable, and secure than that microservices architecture, where this starts by dividing a huge, and one-heavy monolithic system into a set of independent little services that may be scaled and updated separately. The change in this enhances the institution's ability to immediately adjust to the requirements of market conditions, customers, and regulatory demands. Adopted microservices have real-time data processing ability and can be achieved using tools like Apache Kafka and Spark. As a quote from the current market, big data claims that high-frequency trading, portfolio management, and risk assessment cannot count. Traditional financial institutions can still take advantage of the power of handling and analyzing enormous amounts of data in real-time. Further improvement of this capability is achieved by incorporating artificial intelligence (AI) and large language models (LLMs) into microservices, bringing decision-making to automation, predictive analytics and personalized services. Microservices and AI-driven technologies are cornerstones of the modern financial system, where organizations can deliver faster, smarter, and safer customer services.

The key to deploying a successful microservice is cloud-native infrastructure. The scalability, flexibility and low costs of cloud platforms allow financial institutions to implement a microservices architecture that is robust, protected and efficient. Blockchain's integration with decentralized finance (DeFi) advancements further boosted the transparency and security of financial transactions while introducing new avenues for innovation in the financial industry. The availability of generative AI microservices has not only expanded the possible automation of financial processes (from customer service to risk management) but, in so doing, also contributed to improving operation efficiency and customer satisfaction.

The financial industry constantly evolves, and institutions must be modernized continuously to follow new tech developments and changing market dynamics. The one thing to learn is that the journey towards adopting microservice is not easy. To make this big shift happen, financial institutions must properly plan the transition by integrating legacy data, communication between cross systems and security concerns. Additionally, the migration involves overcoming resistance to change in the organization, upskilling teams, and allocating resources in the best possible way. Swapping to an architecture that uses a microservices-based approach has many more benefits. Usually, microservices allow financial institutions to scale and innovate faster and more securely by isolating each service's risk while minimizing the slowdown's damage. Additionally, integrating AI and real-time data processing enables financial systems to take more informed and time-bound decisions to achieve an advantage over the market.

As the financial industry trend continues to move toward more complex data-oriented and pave the way to more profitable, agile and scalable financial systems, researchers expect the demand for such systems to increase. For financial institutions, the future of finance implies modernizing their technology; adopting technologies like microservices, AI, cloud-native infrastructure, and blockchain will be a great preparation for these institutions. Being at the forefront of this transformation journey is essential to staying clear of the competition, meeting customer expectations, and succeeding in the long term in an increasingly digitized world. Converting legacy systems into

Research Article

microservices is not just about improving technology. It is also about being a choice designed to give financial institutions the ability to succeed in a rapidly changing environment. The ability to explore the power of microservices, real-time data processing, and the use of AI technologies in the infrastructure support of financial institutions makes them open to responding to future challenges and opportunities, allowing them to achieve increased agility, responsiveness, and reliability.

REFERENCES;

[1] Ali, T. (2024). *Next-generation intrusion detection systems with LLMs: real-time anomaly detection, explainable AI, and adaptive data generation* (Master's thesis, T. Ali).
[2] Allioui, H., & Mourdi, Y. (2023). Exploring the full potentials of IoT for better financial growth and stability: A comprehensive survey. *Sensors*, *23*(19), 8015.
[3] Blessing, E., Saleh, M., & Jason, H. (2024). Big Data in Finance: Data processing and storage solutions for handling large financial datasets.
[4] Boppiniti, S. T. (2019). Machine learning for predictive analytics: Enhancing data-driven decision-making across industries. *International Journal of Sustainable Development in Computing Science*, *1*(3).
[5] Chatterjee, P. (2023). Cloud-Native Architecture for High-Performance Payment System.
[6] Chavan, A. (2022). Importance of identifying and establishing context boundaries while migrating from monolith to microservices. Journal of Engineering and Applied Sciences Technology, 4, E168. http://doi.org/10.47363/JEAST/2022(4)E168
[7] Cristofaro, T. (2023). *Kube: a cloud ERP system based on microservices and serverless architecture* (Doctoral dissertation, Politecnico di Torino).
[8] Dhanagari, M. R. (2024). MongoDB and data consistency: Bridging the gap between performance and reliability. *Journal of Computer Science and Technology Studies, 6*(2), 183-198. https://doi.org/10.32996/jcsts.2024.6.2.21
[9] Dhanagari, M. R. (2024). Scaling with MongoDB: Solutions for handling big data in real-time. *Journal of Computer Science and Technology Studies, 6*(5), 246-264. https://doi.org/10.32996/jcsts.2024.6.5.20
[10] Dupont, B. (2019). The cyber-resilience of financial institutions: significance and applicability. *Journal of cybersecurity*, *5*(1), tyz013.
[11] Engelmann, A., & Schwabe, G. (2018). Enabling workers to enter Industry 4.0: A layered mobile learning architecture.
[12] Goel, G., & Bhramhabhatt, R. (2024). Dual sourcing strategies. *International Journal of Science and Research Archive*, 13(2), 2155. https://doi.org/10.30574/ijsra.2024.13.2.2155
[13] Karwa, K. (2024). Navigating the job market: Tailored career advice for design students. *International Journal of Emerging Business*, *23*(2). https://www.ashwinanokha.com/ijeb-v23-2-2024.php
[14] Kaur, S., Kumar, K., Singh, J., & Ghumman, N. S. (2015, March). Round-robin based load balancing in Software Defined Networking. In *2015 2nd international conference on computing for sustainable global development (INDIACom)* (pp. 2136-2139). IEEE.
[15] Khan, O. M. A., & Chandaka, A. (2021). *Developing Microservices Architecture on Microsoft Azure with Open Source Technologies*. Microsoft Press.
[16] Khurana, R. (2020). Fraud detection in ecommerce payment systems: The role of predictive ai in real-time transaction security and risk management. *International Journal of Applied Machine Learning and Computational Intelligence*, *10*(6), 1-32.
[17] Konneru, N. M. K. (2021). Integrating security into CI/CD pipelines: A DevSecOps approach with SAST, DAST, and SCA tools. *International Journal of Science and Research Archive*. Retrieved from https://ijsra.net/content/role-notification-scheduling-improving-patient
[18] Koya, S. R. M. (2024). *Microservice Architecture for Social Media Data Collection, Analysis, and Dashboarding* (Master's thesis, University of Arkansas at Little Rock).
[19] Kumar, A. (2019). The convergence of predictive analytics in driving business intelligence and enhancing DevOps efficiency. International Journal of Computational Engineering and Management, 6(6), 118-142. Retrieved from https://ijcem.in/wp-content/uploads/THE-CONVERGENCE-OF-PREDICTIVE-ANALYTICS-IN-DRIVING-BUSINESS-INTELLIGENCE-AND-ENHANCING-DEVOPS-EFFICIENCY.pdf

[20] Lee, J. (2020). Access to finance for artificial intelligence regulation in the financial services industry. *European Business Organization Law Review*, *21*(4), 731-757.
[21] Lorignon, F., Gossard, A., & Carboni, M. (2020). Hierarchically porous monolithic MOFs: An ongoing challenge for industrial-scale effluent treatment. *Chemical Engineering Journal*, *393*, 124765.
[22] Mazzara, M., Dragoni, N., Bucchiarone, A., Giaretta, A., Larsen, S. T., & Dustdar, S. (2018). Microservices: Migration of a mission critical system. *IEEE Transactions on Services Computing*, *14*(5), 1464-1477.
[23] Nabeel, M. (2017, June). The many faces of end-to-end encryption and their security analysis. In *2017 IEEE international conference on edge computing (EDGE)* (pp. 252-259). IEEE.
[24] Nyati, S. (2018). Revolutionizing LTL carrier operations: A comprehensive analysis of an algorithm-driven pickup and delivery dispatching solution. International Journal of Science and Research (IJSR), 7(2), 1659-1666. Retrieved from https://www.ijsr.net/getabstract.php?paperid=SR24203183637
[25] Ogunwole, O., Onukwulu, E. C., Joel, M. O., Adaga, E. M., & Ibeh, A. I. (2023). Modernizing legacy systems: A scalable approach to next-generation data architectures and seamless integration. *International Journal of Multidisciplinary Research and Growth Evaluation*, *4*(1), 901-909.
[26] Owoade, S. J., Uzoka, A., Akerele, J. I., & Ojukwu, P. U. (2024). Enhancing financial portfolio management with predictive analytics and scalable data modeling techniques. *International Journal of Applied Research in Social Sciences*, *6*(11), 2678-2690.
[27] Owoade, S. J., Uzoka, A., Akerele, J. I., & Ojukwu, P. U. (2024). Cloud-based compliance and data security solutions in financial applications using CI/CD pipelines. *World Journal of Engineering and Technology Research*, *8*(2), 152-169.
[28] Raj, P., Vanga, S., & Chaudhary, A. (2022). *Cloud-Native Computing: How to design, develop, and secure microservices and event-driven applications*. John Wiley & Sons.
[29] Raj, P., Vanga, S., & Chaudhary, A. (2022). *Cloud-Native Computing: How to design, develop, and secure microservices and event-driven applications*. John Wiley & Sons.
[30] Raju, R. K. (2017). Dynamic memory inference network for natural language inference. International Journal of Science and Research (IJSR), 6(2). https://www.ijsr.net/archive/v6i2/SR24926091431.pdf
[31] Rane, N., Choudhary, S., & Rane, J. (2023). Blockchain and Artificial Intelligence (AI) integration for revolutionizing security and transparency in finance. *Available at SSRN 4644253*.
[32] Ryandika, D. G., & Prabowo, W. A. (2023, November). Two-stage encryption for strengthening data security in web-based databases: AES-256 and RSA integration. In *2023 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)* (pp. 486-492). IEEE.
[33] Salah, T., Zemerly, M. J., Yeun, C. Y., Al-Qutayri, M., & Al-Hammadi, Y. (2016, December). The evolution of distributed systems towards microservices architecture. In *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)* (pp. 318-325). IEEE.
[34] Sardana, J. (2022). Scalable systems for healthcare communication: A design perspective. *International Journal of Science and Research Archive*. https://doi.org/10.30574/ijsra.2022.7.2.0253
[35] Sardana, J. (2022). The role of notification scheduling in improving patient outcomes. *International Journal of Science and Research Archive*. Retrieved from https://ijsra.net/content/role-notification-scheduling-improving-patient
[36] Singh, V., Unadkat, V., & Kanani, P. (2019). Intelligent traffic management system. *International Journal of Recent Technology and Engineering (IJRTE)*, *8*(3), 7592-7597. https://www.researchgate.net/profile/Pratik-Kanani/publication/341323324_Intelligent_Traffic_Management_System/links/5ebac410299bf1c09ab59e87/Intelligent-Traffic-Management-System.pdf
[37] Şuştic, R. D., Moraru, A., Rus, A. B., & Dobrota, V. (2022). Performance evaluation of ELK stack versus Graylog as Open-Source log management tools. *Acta Technica Napocensis*, *62*(1), 17-22.
[38] Wang, Y., Kadiyala, H., & Rubin, J. (2021). Promises and challenges of microservices: an exploratory study. *Empirical Software Engineering*, *26*(4), 63.
[39] Wolfart, D., Assunção, W. K., da Silva, I. F., Domingos, D. C., Schmeing, E., Villaca, G. L. D., & Paza, D. D. N. (2021, June). Modernizing legacy systems with microservices: A roadmap. In *Proceedings of the 25th International Conference on Evaluation and Assessment in Software Engineering* (pp. 149-159).

Research Article

[40] Xu, B. (2018, June). Research on Countermeasures for Internet Financial Risk Management. In *2018 International Conference on Sports, Arts, Education and Management Engineering (SAEME 2018)* (pp. 65-72). Atlantis Press.

[41] Yarygina, T., & Bagge, A. H. (2018, March). Overcoming security challenges in microservice architectures. In *2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)* (pp. 11-20). IEEE.