Phoenix: International Multidisciplinary Research Journal
Vol 1, No.1, Jan-March, 2020
ISSN: 2583-6897
Special Issue

EVENT-DRIVEN ARCHITECTURE AND API OBSERVABILITY FOR REAL-TIME FINANCIAL TRANSACTION SYSTEMS

Nikita Chawla
Email: nikitachawla83@gmail.com
Independent Researcher, USA

Abstract: This study discusses how event-driven architecture and API observability can help to make financial transactions systems reliable in real-time. The literature discusses the observation of the use of asynchronous event processing and observability structures in enhancing scalability and the visibility of transactions. The research is based on an explanatory research design and secondary data analysis ass it evaluates event streaming and monitoring can transform financial systems through the reactive mode into proactive operations. The results reveal that event-driven architectures lower latency and improve resilience and large volumes of transactions, whereas API observability contributes to improving security, compliance, and end-to-end traceability. The study emphasises the need to adopt clouds, governance and skills training of resilient financial platforms.

Keywords: Event-Driven Architecture, API Observability, Real-Time Transactions, Financial Systems, Distributed Systems, System Resilience

I. INTRODUCTION

A. Background to the Study

Digital transactions, online banking, and the financial technological landscape are ever-expanding throughout the globe and need to be fast, reliable, and precise with the real-time financial transaction systems. The presumption issues of the latency, scaling, and fault tolerance increase with the scale of the transaction load in the monolithic and synchronous traditional architectures [1]. Event-Driven Architecture (EDA) is capable of functioning within a mission-critical financial environment, which is termed asynchronous processing, high throughput, and resiliency. Simultaneously, API observability offers transparency, monitoring, and security, along with operational visibility, the aspects required to attain compliance with the regulations, the identification of fraud, and the trust of customers in the financial ecosystems.

B. Overview

The study explains how an Event-Driven Architecture and API observability can impact ensuring real-time financial transactions processing. EDA provides continuous event streaming, real-time validation, and automatic responses, and observability assists in the monitoring of the work of the system, traces of transactions, anomalies, and ensures that the service is trustworthy [2]. The integration of one with another supports the efficiency of its operation, rise in resiliency of systems, assistance in the law-compliance, and addition of security. The study identifies the areas of architecture, performance benefits, implementation considerations, and real-life details of modern monetary platforms, payment gateways, and electronic banking environments.

C. Problem Statement

The problems associated with financial transaction systems include high levels of latency, bottlenecks, system breakdowns, and subpar visibility of distributed services, amongst others [3]. The traditional architectures are typically incapable of being scaled and simply do not provide end-to-end monitoring, and therefore, it is not simple to detect fraud, cure failure within a limited duration, or even meet compliance requirements. Absence of effective options of architecture and observability ways leads financial institutions to disruption of operation, regulatory penalties, network insecurity, and consumer displeasure. A sense of urgent need, therefore, to adopt architectures and monitoring frameworks that can help in the effective processing of transactions in a more reliable, secure, and real-time environment.

D. Aim and Objectives

The aim is to analysis the application of an Event-Driven Architecture and API observability, in terms of enhancing performance, reliability, and transparency of real-time financial transaction systems. The objectives are: 1. To quantify the scale, fault tolerance, and real-time processing of the Event-Driven Architecture in the financial systems. 2. To investigate the relevance of API observability in observability, security, compliance, and transactions end-to-end visibility. 3. To identify the issues, best practices, and realities to observe in executing EDA and observability in the financial environment.

E. Scope and Significance

The given study will mark the existing digital banking services, payment systems, fintech applications, and change environments of financial transactions, accepting the Event-Driven Architecture and API observability. It describes how they contributed to improving the speed, durability, detection of frauds and transparency of operations [4]. The significance of the research is that it has assisted financial institutions, systems architects, and policy makers to embrace quality architectural design in enhancing reliability, compliance with regulatory requirements, and trustworthiness. It also assists in academic dialogue of relating technological novelty with prominent financial system performance innovation.

II. LITERATURE REVIEW

A. Quantifying Scale, Real-Time Processing and Fault Tolerance in Event-Driven Architecture

Event-Driven Architecture (EDA) is implemented in the modern financial systems as enhancement of their scalability, fault tolerance and processing capacity in real time. Financial transaction platforms are designed to meet extreme conditions in that they process high transaction volumes and 24/7 availability commands [5]. The EDA mainly promotes these requirements by decoupling the system parts using asynchronous event streams as the services can automatically scale without interference to the whole system. There are one of the vital benefits is real-time processing as it allows to verify transactions and refresh balance in real-time. As per industrial standards, real time, event driven systems can decrease the processing lag of transactions by up to 60% as compared to the synchronous architectures [6]. These benefits are measured by quantifying throughput rates and recovery times and system resiliency during peak loads. This could reveal the role of EDA in operational stability and performance in particularly high-stakes financial settings as there a few minutes of delays or failures can result as high financial and reputational costs.

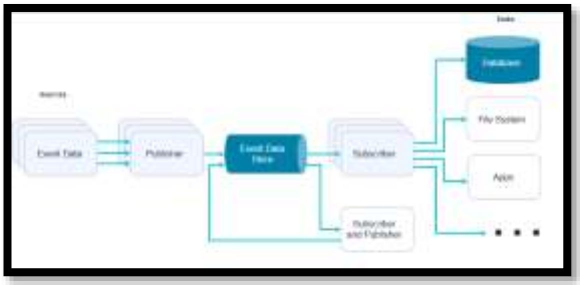

Figure 1: Event-Driven Architecture
[6]

B. Relevance of API Observability for Security and End-to-End Visibility

APIs has purpose to communicate with the internal services layer and third-party platforms, as well as those used by customers, and thus it is a focus of visibility and control. The API observability allows tracking request streams, response latency, error rates, and abnormal access patterns constantly [7]. This allows organisations to identify potential attacks in time. Observability mainly facilitates auditability as use of detailed log traces also it needed to comply with regulatory reporting and inquiry. Transaction visibility is significant as it is a single operation can go through a series of microservices and external integrations. There are more than 70% of failures in the financial system are attributed to the API failures or misconfigurations [8]. This emphasises that API observability can able to enhance transparency and confidence in digital finances and also meets the rigid regulatory and security requirements.

C. Issues, Best Practices, and Practical Realities of Implementing EDA and Observability

There are some challenges that are involved in applying EDA and observability to financial environment. These technologies have advantages also these present challenges of complexity in system design, data consistency, tooling integration, and skills requirement [9]. A general problem is the managing of event duplication and ordering that may give false states of transaction. Equally, logging and monitoring done excessively may add to the infrastructure costs and may cause a data overload, making observability effort less efficient. Good practices entail clear event schemas and using standard observability frameworks as it balances visibility and performance. The implementation of outcomes is also done based on organisational realities as integration of legacy systems and regulatory constraints [10]. These limitations with the best practices of implementation provides a realistic perspective of the execution of EDA and observability. It also stresses on the value of progressive implementation and inter-functional cooperation to maintain sustainable and compliant system evolution.

III. METHODOLOGY

A. Research Design

The present study design is explanatory in nature to comprehend how the issue of Event-Driven Architecture and API observability can lead to the efficiency, reliability, and transparency of the real-time financial transactions systems. It is also geared towards the establishment of relationships, explanation of the impact on the technology, as well as the clarification of how architectural decisions make a difference in performance, security, scalability, compliance, and operational visibility of current financial systems and payment structures.

B. Data Collection and Analysis

The study implied secondary qualitative and quantitative data collection as compiled in the form of academic journals, industry reports, technical whitepapers, financial technology publications, case studies, and regulatory documentation. Research findings are then evaluated by analysis, as well as making a comparison concerning the architectural methods, advantages, and disadvantages of the same, and insights can be drawn. The combination of the existing literature will make it possible to conduct the analysis and find a derived knowledge of the real-time architecture of financial transactions.

C. Case Studies/Examples

Case Study 1: Event-Driven Architecture for Real-Time Transactions in PayPal

PayPal used Apache Kafka to ensure event streaming on large volumes and consequently carry out asynchronous communication, real-time processing of the transaction, and a reliable flow of information to its dispersed financial services [11]. Kafka helped PayPal to process volumes of transactions with reduced latency, scalability, reduced resiliency, and monitoring. The event-driven model strengthened fraud detection, logging, and operational visibility and supported the availability of the service at any given time. This was an excellent modernisation that has increased the performance and reliability of PayPal in the international payment system.

Case Study 2: Event-Driven Digital Banking Platform in Monzo Bank

Monzo created a digital banking platform, which was powered by cloud-native principles, to support financial transactions by allowing processing events in real time, on the card of both microservices and event-driven principles [12]. The advantage of the architecture was that it real-time notified about transactions, reduced the processing latency, increased the resiliency of the services, and performed automatic health checks of the system. Event streaming also helped in viewing the events, quick detection of faults, and improved customer experience. Monzo has demonstrated that fintech can use event-driven design and observability in order to elicit reliable and scaled banking services.

Case Study 3: Real-Time Global Payment Processing Network in VisaNet

VisaNet is one of the largest real-time commercial transaction networks in the world that supports authorisation, clearing, and settlement of financial transactions across the world. It has built-in risk management and fraud detection, is able to run at extremely high throughput, and offers secure and low-latency payment approvals [13]. It possesses an architecture that emphasises reliability, high availability, resilience, and strong monitoring in a distributed system. VisaNet has been a successful model of scalable real-time financial transaction processing that is supported with high observability and system intelligence.

D. Metrics of Evaluation

The evaluation will be done based on latency in the system, throughput, fault tolerance, scalability efficiency, and the accuracy of the event processing when measuring the performance. The reliability measures entail the error rates, system availability, and recovery time. Measures of security and compliance effectiveness are observability coverage, anomaly detection ability, traceability, and audit ability. The factor of user trust may be considered in terms of stability, transparency, and consistency of the operations in the course of the real-time transactions.

IV. RESULTS

A. Data Presentation

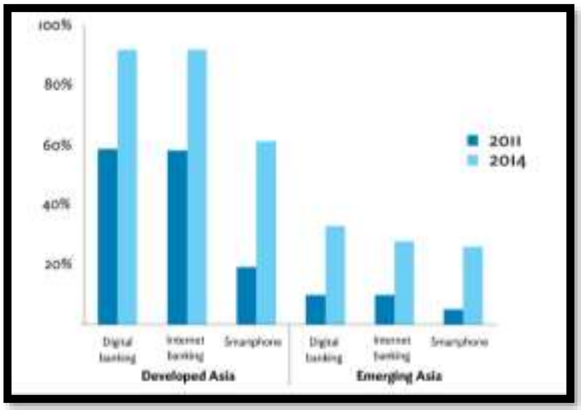

Figure 2: Most Popular Payment Methods
[14]

The graph mainly represents the most frequent payment options according to the survey. The dominance of in-store payments of 40% show that in-store physical point-of-sale transactions are also very important. This is followed by the online payments of 15% are followed by the other online ways with buy buttons at 9% [14]. There are 8% respondents' choses other mobile transfers and P2P transfers is 7% also mobile messenger app is 7%. This indicates efficient development of digital payment ecosystems. The above various real-time payment platforms mainly demonstrate the importance of event-driven architecture and robust API observability in emerging financial transaction systems ***[referred to Figure 2]***.

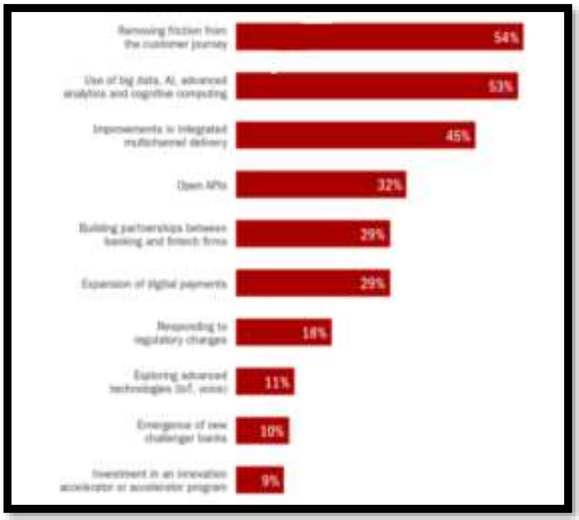

Figure 3: Major Trends in the Modern Retail Banking
[15]

The graph identifies major trends in the modern retail banking and enhancement of digital transactions. The ability to remove friction in the customer journey is prioritised the most at 54% then application of big data, AI, and modernised analytics at 53% [15]. Amazon experiences improved integrated multichannel delivery of 45% as show the value of the common customer experiences across platforms. The focus on increasing ecosystem-based innovation is supported by open APIs 32% and bank-fintech company partnerships of 29% [15]. The above trends justify that event-driven architecture and API observability are necessary to support real-time data processing and dependable monitoring of contemporary financial transactions systems ***[referred to Figure 3]***.

B. Findings

The research shows that financial transactions may has various channels and these are becoming very diverse and real-time with in-store payments at 40 percent dominant [14]. This diversity mainly adds some complexity over systems and as it provides scalability to high-volume flows of asynchronised transactions. Moreover, customer frictionless journeys (54%), big data and artificial intelligence (53%), and other priorities of the retail banking industry underscore the need to process data in real-time and have the system responsiveness [15]. The focus on open APIs, fintech collaboration and the trend of multichannel delivery further highlights the relevance and significance of efficient API observability among financial networks.

C. Case Study Outcomes

Case Study 1: Event-Driven Architecture for Real-Time Transactions in PayPal

- This facilitated real-time and world-scale transaction processing, with the effect of greatly decreasing the latency also managing heavy volumes of transactions without the loss of service.
- Decoupled service provides better system resilience and availability as enable the isolation of faults and steady flow of transactions [11].

Case Study 2: Event-Driven Digital Banking Platform in Monzo Bank

- This enables real-time notifications of transactions with low latency also enhances the responsiveness of the service and its high availability across micro services [12].
- There are efficient observability and automatic health checks that allow prompt fault detection and an easier customer banking experience.

Case Study 3: Real-Time Global Payment Processing Network in VisaNet

- This provides globally at scale with high availability in terms of payment authorisation and settlement also delivered at massive scale with high availability and resilience.
- There are advanced risk management, fraud detection and high observability as these makes global Visa payment network achievable in real time and reliable working of global Visa payments network [13].

D. Comparative Analysis

Author	**Focus**	**Key Findings**	**Literature Gap**
[5]	Event-driven architectures in geospatial IoT systems	Describes that event driven architectures enhance real time processing, interoperability and scalability of spatial data infrastructures.	Weak attention to financial transaction systems and API-level observability in high-frequency systems.

[6]	Fault tolerance in large-scale dataflow architectures	Introduces architectural mechanisms to develop resilience, reliability, and fault tolerance of high-volume distributed systems.	Does not look into actual financial flow and observability needs.
[7]	EDA tools for IC implementation and circuit design	Provides an investigation on the use of EDA to aid the efficiency, accuracy and optimisation of integrated circuit implementation.	Application to event-driven systems based on software and financial spheres.
[9]	EDA for IC system design, verification, and testing	Demonstrates the EDA effect on enhancing system verification, test reliability and design consistency.	Concentrated on hardware systems as opposed to real-time digital transaction platforms.
[10]	Cloud readiness of the EDA industry	Determines opportunities and pitfalls of cloud computing adoption in the EDA environments.	No mention made on API observability or cloud-native event-driven financial architecture.

Table 1: Comparative Analysis

(Source: Self-developed)

As the table shows, current literature is a clear proponent of event-driven and fault-tolerant architecture, with little evidence of real-time financial systems and to full API observability ***[referred to Table 1]***.

V. DISCUSSION

A. Interpretation of Results

The findings demonstrate that event-driven structure and API observability are the key facilitators of dependability and performance in today financial systems. The case studies indicate that PayPal, Monzo and VisaNet organisations can implement event-driven designs with high observability as these can enable the achievement of real-time processing. The distribution of payment methods also illustrates the growing transactional diversity in the in-store, online, and mobile channel, hence the necessity to have scalable and traceable systems [14]. The trends in the industry that highlight frictionless customer experiences, analytics based on AI, open APIs and fintech are stimulating the need to have real-time data flow and end-to-end visibility. These results mainly indicate that the use of event-driven architecture and strong API observability helps financial institutions to cope with complexity and improve customer experience [15]. It also helps in compliance with regulatory requirement also supports in operational resilience for dynamic digital payment.

B. Practical Implications

The results allow different operational implications of the financial institutions and fintech organisations. Event driven architecture enables the companies to manage a large volume of transactions over a variety of payment systems with reduced latencies, enhanced scalability and fault resistance [14]. The increased API observability enables organisations to follow transactions across the board and ensure regulatory and security measures are adhered to. This helps to respond to incidents quicker, increase fraud detection and enhance system visibility. The findings also suggest that banks need to invest in cloud-native systems and embedded monitoring devices in order to support real-time operations. Event-driven systems can enhance the reliability of services and efficiency of the operations in complex and interdependent financial ecosystems.

C. Challenges and Limitations

There are a number of challenges and limitations that influence the implementation of event-driven architecture and API observability within business finances. These architectures create complexity in the system and demand specialised abilities, event design care, and good governance mechanisms to ensure such phenomena as event duplication and inconsistent data are averted. The combination of event-driven systems and legacy banking systems is not easy and may delay the execution schedule. Observability tools have the potential to produce high amounts of logs and metrics that can be costly to the infrastructure and potentially overwhelm information without a proper control mechanism in place. Furthermore, it is difficult to secure, protect, and regulate the data over distributed event streams and especially in transactions across different borders. It is also limited by dependency on secondary data and some case studies which are not comprehensive of organisational restrictions and implementation failures in various financial institutions.

D. Recommendations

According to the findings, event-driven architecture is to be implemented gradually by financial institutions based on the type of impact of the service on the business to reduce risk and complexity [6]. Distinct event standards, idempotent processing and robust governance structures ought to be implemented in order to determine the consistency and reliability of data. Companies need to invest in end-to-end API observability technologies which combine logging, metrics, and distributed tracing to obtain complete visibility of the transaction [8]. There need to upskill the teams as management of cloud-native and event-driven technologies to facilitate successful implementation. Further improvement of the system resilience, scale, or customer experience may be achieved through continuous testing and collaboration with fintech partners.

VI. CONCLUSION AND FUTURE WORK

This study focused on event-driven architecture and API observability that can be used to support a reliable and real-time system of financial transactions. The results show that event-driven designs enhance scalability and processing speed of transaction over various payment channels also API observability increases end-to-end visibility, security and regulatory compliance. The case studies of PayPal, Monzo, and VisaNet mainly demonstrate that the major financial organisations managed to utilise asynchronous processing and robust monitoring to ensure high availability and customer confidence. The discussion also clarifies that the expansion of payments and data-driven decision-making make systems more complex and thus real-time event management. The research indicates that the combination of event-driven architecture and strong observability frameworks contribute to the operation resilience and the financial institution can adapt well to the digital world that changes swiftly.

Further studies in the future on ways of assessing empirically using primary data maintained by financial institutions the extent of performance enhancement, cost, and implementation problems would be interesting. Cross-regional comparisons that could be based on the regulation areas may improve information in terms of compliance-based architecture selection. Additional studies can take a look at how AI-based analytics can be combined with event streams to improve fraud detection and autonomous system control.

VII. REFERENCE LIST

[1] Nasim, M., Nguyen, A., Lothian, N., Cope, R. and Mitchell, L., 2018, April. Real-time detection of content polluters in partially observable Twitter networks. In *Companion Proceedings of the The Web Conference 2018* (pp. 1331-1339).

[2] Erik, S. and Emma, L., 2018. Real-Time Analytics with Event-Driven Architectures: Powering Next-Gen Business Intelligence. *International Journal of Trend in Scientific Research and Development*, *2*(4), pp.3097-3111.

[3] Zhelev, S. and Rozeva, A., 2019, November. Using microservices and event driven architecture for big data stream processing. In *AIP Conference Proceedings* (Vol. 2172, No. 1, p. 090010). AIP Publishing LLC.

[4] Megargel, A., Shankararaman, V. and Reddy, S.K., 2018. Real-time inbound marketing: A use case for digital banking. In *Handbook of Blockchain, Digital Finance, and Inclusion, Volume 1* (pp. 311-328). Academic Press.

[5] Rieke, M., Bigagli, L., Herle, S., Jirka, S., Kotsev, A., Liebig, T., Malewski, C., Paschke, T. and Stasch, C., 2018. Geospatial IoT—The need for event-driven architectures in contemporary spatial data infrastructures. *ISPRS International Journal of Geo-Information*, *7*(10), p.385.

[6] Weis, S., Garbade, A., Fechner, B., Mendelson, A., Giorgi, R. and Ungerer, T., 2016. Architectural support for fault tolerance in a teradevice dataflow system. *International Journal of Parallel Programming*, *44*(2), pp.208-232.

[7] Lavagno, L., Scheffer, L. and Martin, G. eds., 2018. *EDA for IC implementation, circuit design, and process technology*. CRC press.

[8] Coneksion, 2018. 10 *Reasons Why Integration Projects Fail*. Available at: https://www.coneksion.com/blog/10-reasons-why-integration-projects-fail. [Accessed On: 18th November, 2019].

[9] Scheffer, L., Lavagno, L. and Martin, G. eds., 2018. *EDA for IC system design, verification, and testing*. CRC press.

[10] Sehgal, N., Acken, J.M. and Sohoni, S., 2016. Is the EDA industry ready for cloud computing?. *IETE Technical Review*, *33*(4), pp.345-356.

[11] Highscalability.com, 2016, *how-paypal-scaled-to-billions-of-transactions-daily-using-ju* Available at: https://highscalability.com/how-paypal-scaled-to-billions-of-transactions-daily-using-ju/ [Accessed on: 10th September 2019]

[12] Thefinancialbrand.com, 2018, *fintech-banking* Available at:https://thefinancialbrand.com/news/fintech-banking/digital-banking-future-mondo-monzo-challenger-75087 [Accessed on: 15th November 2019]

[13] Onafriq.com, 2019, *visa-and-mfs-africa-bring-digital-payments* Available at: https://onafriq.com/press/article/visa-and-mfs-africa-bring-digital-payments [Accessed on: 23rd December 2019]

[14] Printecgroup, 2019. *The rise of digital payments and how they are changing the financial industry*. Available at: https://blog.printecgroup.com/the-rise-of-digital-payments-and-how-they-are-changing-the-financial-industry. [Accessed On: 20th November, 2019].
[15] The Financial Brand, 2016. *Retail Banking Trends and Predictions for 2017*. Available at: https://thefinancialbrand.com/news/banking-trends-strategies/2017-top-banking-trends-predictions-forecast-digital-fintech-data-api-62858. [Accessed On: 19th November, 2019].
[16] Goli, A. K. R. THE ROLE OF SRE IN ACHIEVING OPERATIONAL RESILIENCE IN CLOUD-BASED ENTERPRISES.
[17] Chintale, P. (2020). Designing a secure self-onboarding system for internet customers using Google cloud SaaS framework. IJAR, 6(5), 482-487.
[18] Goli, S. R. (2020). Energy Sector Modernisation: DevOps and Cloud as Catalysts for Operational Efficiency. *Available at SSRN 5741642*.