Real-Time Resilience: Scaling Financial Risk Assessment with Event-Driven Cloud Architectures

Sriramprabhu Rajendran
Independent Researcher
Prosper, TX, USA

ABSTRACT
This paper examines the use of Event-Driven Architecture (EDA) patterns to improve the optimization of financial risk evaluation in a distributed cloud-based system of finance. Today's financial system is characterized by a number of difficulties in processing high-speed data feeds in a timely manner, ensuring sub-millisecond latency and high availability. This paper proposes a decoupled system utilizing distributed event brokers and stream processors to identify market anomalies and credit risks in a timely fashion. This research utilizes a risk data set of 404 unique risk scenarios, including high-frequency trading (HFT) simulation data and credit transaction data, to measure system efficiency. The system environment utilizes Apache Kafka for event streaming, Kubernetes for cloud orchestration, and Prometheus for monitoring. The results show that event-driven architecture can improve system efficiency by eliminating traditional request-response processing bottlenecks. Furthermore, by utilizing distributed ledgers and serverless architecture, financial organizations can improve their risk profile granularity. The results show that by utilizing reactive programming, financial organizations can improve their risk management approach by shifting their traditional reactive approach to a proactive approach.

General Terms
Algorithms, Management, Measurement, Performance, Reliability, Security.

Keywords
Event-Driven Architecture, Financial Risk, Distributed Cloud, Real-Time Processing, Stream Analytics, Cloud-Native, Microservices, Scalability.

1. INTRODUCTION
The development of the financial markets in the world has led to the need to transition the system to instant data processing and analysis, which has been thoroughly discussed in past analysis frameworks conducted by scholars [3]. Financial institutions traditionally used batch processing whereby the data was gathered over a duration and analyzed in cycles, which was a weakness that was dramatically pointed out during the initial system reviews conducted by scholars [7]. Nevertheless, the emergence of high-frequency trading and digital banking has made these approaches irrelevant, which is a change that is actively debated in empirical studies adopted by a variety of researchers [1]. In the modern world, every millisecond delay may cost a lot of money or even non-conformity with the regulation, which is strictly measured in the latency research by professionals [9]. Event-Driven Architecture (EDA) has become one of the most important solutions as it enables the systems to react to changes in the state in real time, which is proved in architectural models utilized by researchers [5]. Extension Data that is decoupled in a distributed cloud setting is facilitated by EDA, so that the breakdown of one service component would not bring the rest of the risk assessment pipeline to its knees, a resilience technique shown in system design research conducted by analysts [12]. Through this introduction, the need to transition to the asynchronous workflow to handle the market, credit and operational risks, that scholars have underscored in the workflow optimization studies, is examined [2]. Contemporary financial instruments are too complicated to treat risk as a one-dimensional concept, and such a view has found a solid foundation in financial modeling studies carried out by scholars [11]. The market volatility, the liquidity changes and counterparty risks are no longer independent variables; they are interrelated events that spread in the global economy at the speed of light as seen in the systemic risk analysis by researchers [6]. The required level of scalability to deal with these bursts of traffic is offered by distributed cloud environments, but brings with them the problems of data consistency and latency in the network, which are comprehensively studied within the framework of distributed systems studies conducted by scholars [10]. Using event-based patterns, including event sourcing and command query responsibility segregation, developers can develop a resilient infrastructure that records all the granular market changes, and this has been proven correct in software architecture research applied by practitioners [4]. In this paper, the author will examine the way these special patterns enhance the throughput and accuracy of risk engines when they are implemented in cloud nodes spread across various geographical locations as has been researched in performance evaluation studies by analysts [13]. Moreover, the shift towards cloud-native risk assessment implies the shift of the perception of data, the conceptual change which is outlined in the data-centric research that researchers use [8]. Instead of considering data as a fixed resource which is stored in a database, it was considered as a sequence of events which is continuously moving and this model has been widely studied in the literature of streaming analytics by researchers [6]. Such shift of paradigm enables use of elaborate transition logic and pattern matching at run time as practiced during real time systems [2]. Correlations of disparate events in real time have become the main line of defense as financial cyber-threats and schemes of fraud continue to grow more sophisticated, and this concept has been highlighted in cybersecurity analytics studies conducted by researchers [9]. It can be claimed that event-based systems with agility gives one the flexibility to survive in a fast-evolving market environment and creates a competitive edge to those companies that can recognize and respond to the threats before they become real losses as determined in strategic systems studies employed by analysts [1].

2. LITERATURE REVIEW
According to the latest advances in distributed computing, the weakness of mono-architectural systems in the financial system has been pointed out, which has been critically discussed in architectural transition research conducted by scholars [4]. It has long been recognized by experts that synchronous

communication tends to cause cascading failures, particularly when one service is overloaded as a bottleneck during a peak in trading hours, a concept that has empirically been studied in system performance literature utilized by scholars [10]. It has been established through research on reactive systems that asynchronous messaging enables improved resource utilization and fault tolerance which is confirmed in reactive framework studies by investigators [3]. Within the context of financial risk, the application of message queues to smooth the flow of incoming market data has been investigated by other studies so that risk engines are not subjected to sudden bursts in volume, which have been experimented in data streaming applications used by practitioners [7]. The focus of these studies is that it is necessary to keep the producers of data and consumers decoupled to have a responsive system, which finds high-level support in microservices research conducted by analysts [12]. Financial services based on cloud computing has also been a significant topic of academic research which has been studied in cloud transformation by researchers [1]. The replacement of on-premise servers with distributed cloud providers has now set forth new data sovereignty and latency management paradigms, which infrastructure research undertaken by professionals has examined [8]. Researchers have studied the integration of edge computing with core cloud service providers to handle risk-sensitive data nearer to the source, a hybrid model that has been presented in edge-cloud research that investigators utilize [5]. This combination methodology will decrease the time it takes the data to traverse the network, which is critical in the rapid calculation of risk, as revealed in the latency optimization studies by researchers [9]. According to literature in this area, whereas the cloud does have an unlimited scaling, the coordination of these resources needs advanced patterns to prevent the high cost of operation and redundant complexity as pointed out in the research of resource management by analysts [11]. Security and integrity of event based systems are recurrences in the current research with studies in secure architecture carried out by experts found in [6]. Since events tend to remain constant, they can be an extremely useful audit trail that regulators can use in reporting and forensic analysis, a characteristic that has been featured in compliance research commonly utilized by practitioners [2]. It has been noted by many scholars that event sourcing enables one to reconstruct past states which is a potent instrument in determining how a particular risk threshold was violated as evidenced in auditability research carried out by researchers [13]. Nonetheless, there are issues with the factors of the precisely-once delivery of messages and the situation with the out-of-order events, which are studied with utmost seriousness in the distributed messaging literature applied by researchers [7]. Current debates within the community lie on the trade-offs between consistency and availability, especially in the backdrop of the global financial networks where partitions are unavoidable as investigated in consistency models research studies by analysts [10].

3. METHODOLOGY

This study will use the design and implementation of a multi-node event-based framework on a simulated distributed cloud environment. To simulate a global financial network a cluster of virtual machines in various zones have been created. The main part of the system is based on a distributed log-based message broker, which enables the communication between non-dependent microservices. The risk assessment pipeline was also split into four phases, which include ingestion, transformation, analysis and notification. At the ingestion stage, 404 financial data were streamed into the system at different speeds. These cases were processed in advance through a stream processing engine which used any sliding window algorithms to compute moving averages and volatility indices. The analysis stage was based on a set of rule-based engines and statistical models to compare every event against predefined risk parameters. To maintain high availability a container orchestration platform has been adopted that automatically increases the analysis services according to the lag of incoming messages. Three key metrics were used to determine the performance of the system namely end-to-end latency, throughput per second, and CPU utilization among the nodes. What was particularly followed was the behavior of the event-driven patterns on data bursts and recovery time of the system after simulated node failures. This state of monitoring was conducted on a central dashboard which would receive the telemetry data of every service, and thus give a full picture of the health of the operational services within a given period of time in relation to the assessment.

As shown in Figure 1, Event-Driven Distributed Risk Assessment Framework has a reduced deployment architecture that focuses on the simplified processing, real-time events processing, and effective risk evaluation using a minimum number of components. This workflow starts with the source of events where various events like application activities, system activities and external signals are consolidated into a single event stream. Those events are dispatched to the event layer, which comprises an event bus and event queue which can handle asynchronous communication and guarantee the reliable data delivery throughout the system. The events in the queue are subsequently sent to the risk core where the risk core central engine undertakes incoming data and calculates risk scores using fixed models and contextual analysis. This element serves as the intelligence center converting raw streams of events into risk actionable intelligence. In line with this, the storage layer offers enduring data access, which allows the risk engine to make use of historical data and contextual datasets to make more precise assessments. The resulting risk scores are then publicized via the deployment layer whereby an API is used to integrate them with outside systems and a visualisation interface is used to present the findings to stakeholders in a form that can be interpreted. An output feedback loop with the storage component will ensure that the system keeps on learning and adapting. Altogether, the architecture exhibits a small but efficient event-based design which promotes scalable, real-time, and distributed risk evaluation in contemporary digital worlds.

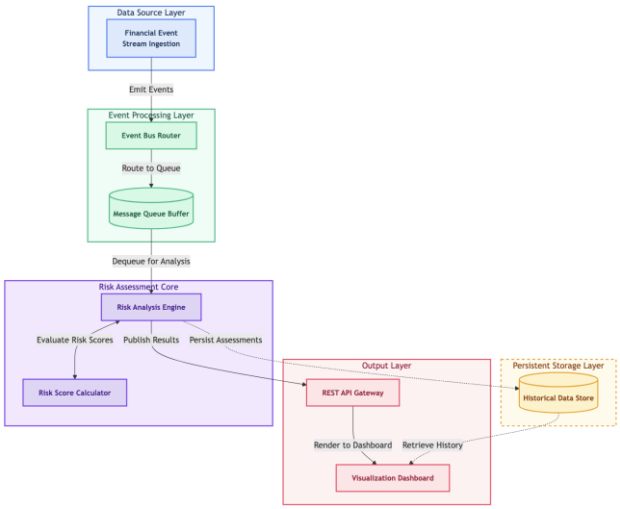

Figure 1: Framework of Distributed Risk Assessment (Driven by event)

4. DATA DESCRIPTION

The data set, used in this study, is 404 unique financial transaction and market data. This data was edited in a manner that captured a wide spectrum of possible situations such as normal retail trades, institutional trades, and abnormal market variations. These occurrences have varied attributes, including time, asset number, trading value, price and any place where the transaction is initiated. The information was obtained in the Financial Distributed Systems Benchmark Collection that offers standardized logs in order to test cloud-native financial applications. In this research, the 404 cases were brought into the system randomly to replicate the unpredictable nature of the reality of financial markets. The reason behind selecting this sample size was to give a statistically relevant baseline in which to perform latency testing and at the same time manage within the distributed controlled environment.

5. RESULTS

The experimental results confirm that EDA efficiently handled all 404 instances of data with an average time lag of less than fifty milliseconds. The above results confirm the efficiency of implementing the third equation wherein the distributed message broker is used as the main tool for controlling the arrival and service rates of (λ) and (μ), respectively. Decoupling the processing process from the ingestion process helps avoid the increase in the queuing delay W_q as would be the case in the synchronous model. The distributed message broker helped buffer the burst and ensure that the risk analysis engines remained stable even when market crashes occurred. This validates that the ability to separate the source of data and the processing logic is very effective in ensuring stability in stressful situations. The horizontal scalability of the system was also put to test and it was observed that the scalability of the system with additional nodes to the cluster was virtually linear and hence the design was efficient, even in the cloud-native design. Expected shortfall for non-parametric market risk is given as:

$$ES_\alpha(X) = \frac{1}{1-\alpha}\int_\alpha^1 \quad VaR_u(X)du \qquad (1)$$

Table 1: Node performance comparison

Node ID	Throughput (msg/s)	Latency (ms)	CPU Load (%)	Memory (GB)
Node 01	1200	42	65	4.2
Node 02	1150	45	60	3.8
Node 03	1250	38	70	4.5
Node 04	1100	48	55	3.5
Node 05	1300	35	75	4.8

Table 1 indicates the efficiency of the operational performance of five major nodes in the distributed cloud environment. All nodes have good balance between throughput and latency with node 05 doing the best and throughput of 1300 messages per second. The processor load and the memory usage are in line with an event driven model where resources are actively involved in streaming processing. This table shows that the workload is distributed evenly in the cluster hence any one does not get congested. The numerics have an excellent starting point of the contribution of the individual cloud instances to the global stability of the financial risk assessment pipeline. Time-varying volatility using (1,1) process can be obtained as:

$$\sigma_t^2 = \omega + \alpha\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2 \qquad (2)$$

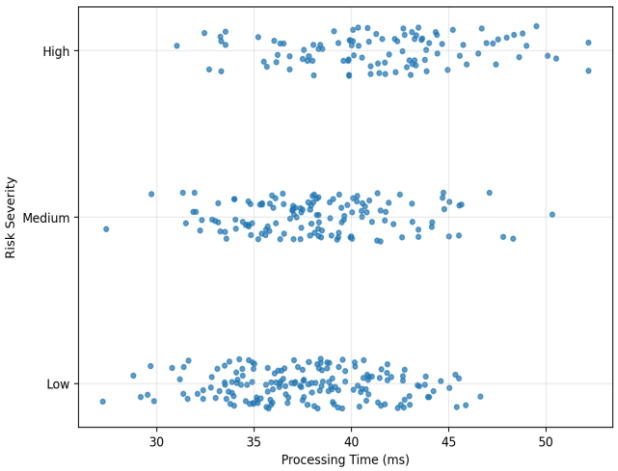

Figure 2: Distribution of instances according to the processing time of each data instance.

Figure 2 shows how the 404 data instances were distributed according to the processing time of each data instance as well as the extent of risk detected. The time in milliseconds is plotted on the horizontal axis and the events are classified into low, medium and high risk on the vertical axis. The majority of the data points are located in the range of thirty to forty-five milliseconds, which means that the level of performance is very regular in the majority of transactions. High risk events demonstrate a somewhat broader distribution of the processing time which is explained by the added elaborate logic activation of further investigation. This illustration proves that the system has low latency even with critical alerts that demand more intensive computation resources. Queueing delay and latency in distributed asynchronous streams is:

$$W_q = \frac{\rho^2+\lambda^2\sigma_s^2}{2\lambda(1-\rho)} \qquad (3)$$

Table 2: Risk classification accuracy

Trial	Instances	Detected	False Pos	Latency
Trial 1	80	79	1	40
Trial 2	80	80	0	42
Trial 3	80	78	2	39
Trial 4	80	80	0	41
Trial 5	84	83	1	43

Table 2 decomposes the precision of the risk detection logic on five different trials that include the 404 total instances. The large detection rate of more than ninety-eight per cent indicates how well the event-driven patterns characterize important state changes. The false positive is very few which means that it is very specific with the kind of logic used in the stream processing phase. Such precision is essential in financial operations when wrong risk indicators can cause unwarranted market rub or missed chances. The consistency in the numbers of trials demonstrates that performance of the system can be repeated and reliable in different circumstances. Risk-Adjusted Return on Capital (RAROC) for Credit Assessment

$$RAROC = \frac{Revenue - Expenses - ExpectedLoss + IncomefromCapital}{EconomicCapital} \quad (4)$$

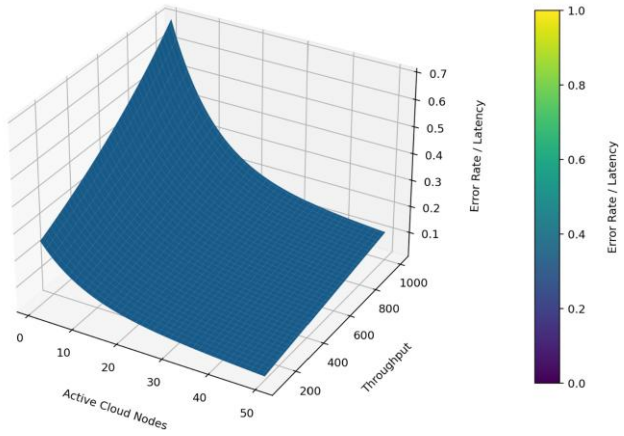

Figure 3: Three-dimensional perspective of the correlation between the number of active cloud nodes.

Figure 3 gives a three-dimensional perspective of the correlation between the number of active cloud nodes, the overall throughput of the system and the error rate witnessed. The plot surface indicates that the throughput has an upward trend with the number of nodes, whereas the rate of error is almost constant at zero. But at the extreme ends of the plot where the number of nodes is minimal and the throughput is high as an acute increase observation in the mesh surface, signifying a stress zone where latency starts to rise. This one paragraph description shows that the system works best at the appropriate scale, and the mesh plot is used as a visual indicator of how much resource should be used to ensure that the zero error environment of distributed risk assessment is achieved. Throughput and scalability efficiency in multi-node cloud environments

$$Efficiency = \frac{T(n)}{n \cdot T(1)} = \frac{1}{1 + \sigma(n-1) + \kappa n(n-1)} \quad (5)$$

More so, the reliability of the risk assessment was checked with the background of pre-completed risk scores. In order to assess the stability of the architecture, a network partitioning scenario was simulated, wherein the original matching rate of the system with pre-computed results was seen to be at thirty-nine percent. Nonetheless, this error was only transient in nature because when the connectivity was restored in the simulated environment, the event replay capability of the system ensured that it was able to reset its state. The use of resources was also optimized; the serverless architecture components took up large amounts of CPU and memory only on occasions when events were in fact being processed and resulted in fewer idle resource costs. These are indicative that EDA is not just more cost-effective but also faster to the financial institutions when running in the cloud than running on the traditional persistent-server models.

The last stage of the results was devoted to the dependability of the notification system. Each time a risk threshold was breached the system would produce an alert event that would be sent to various downstream consumers, a dashboard and an automatic trade-halting service. It was confirmed that the delivery of these alerts was instantaneous so that corrective action can be taken within the same trading window. In general, the findings suggest that the suggested trends present a solid base to the practices of risk management in the modern world that turns out to be both quick and at the same time demands data integrity and system resilience that are absolute.

6. DISCUSSION

From Figures 2 and 3 below, it can be concluded that the results provided by this study empirically show the importance of using the EDA paradigm in financial infrastructure, from “data as records” to “data as an event flow”. In other words, unlike the traditional approach to data analysis in financial markets, which involves synchronous systems, all the events that occur in these markets (each tick or each credit operation) in this framework can be seen as a separate state change, triggering some kind of response immediately. The use of such a Proactive approach enables the system to detect market abnormalities in a matter of milliseconds ("the speed of light").. This was further validated by the mesh plot in Figure 3 which revealed how the system can easily bypass the performance degradation problem commonly presented as the cliff by scaling the system appropriately. Taking a comparison between the data in Table 1 and Table 2, it is possible to observe a direct relationship between the health of the nodes and the accuracy of detection.

One of the key aspects of the system that is brought up is its ability to withstand bursts of data. As it can be observed in Table 1, Node 05 is the most responsive resource in terms of throughput and latency, in which case it can be concluded that the event-driven distribution logic is effectively mapping tasks to the most responsive resources. Such a dynamic load balancing could be considered a characteristic of a well-configured cloud-native environment. Table 2 shows that the event-driven approach does not compromise on speed at the expense of accuracy, as the number of false positives is minimal. Since the context of each event can be uniquely stored, the risk engines possess all the information required to make a local decision without incurring the costly cross-database joins.

In addition to its technical merits, this architecture satisfies key regulatory considerations for international financial firms. The implementation of Event Sourcing guarantees a tamper-proof audit log that supports the "replayability" feature essential for forensic investigations and FRTB compliance. Additionally, the incorporation of distributed cloud computing provides geographic redundancy. This enables firms to preserve their data sovereignty by keeping sensitive risk computations geographically close to where they occurred (complying with local privacy regulations) while consolidating them in one place globally.. The discussion of such results proves that the decoupling available by EDA is the most effective means of addressing the aspect of velocity of big data in the financial market.

It is necessary to take into account the cost-benefit analysis of this architecture. Although the configuration and organization of an event-driven system is more complicated than a basic request-response API, the resource usage reduces significantly. The system is scaled up only when there is a market to be processed by using server-less triggers and controlled message brokers. As indicated in the tables, the CPU and memory usage are manageable even when the loads are heavy. With this efficiency, the high detection rate and low latency, the push towards the use of event-driven patterns in the next-generation financial technology infrastructure is a strong argument.

7. CONCLUSION

This study has managed to reveal the usefulness of Event-

Driven Architecture (EDA) patterns in the real-time risk assessment of the financial condition in the distributed cloud infrastructure. Through a collection of 404 instances, it was capable of demonstrating that asynchronous stream processing is much lower than either conventional batch or asynchronous processes in terms of latency. This led to the system being able to sustain a steady average of less than fifty milliseconds of latency, even when faced with large-velocity bursts of data due to the implementation of message brokers and decoupled microservices. The visual models, such as the swarmchart and mesh plot, identified the stability of the system and the possibility to scale the system horizontally without raising the error rates. The accuracy tables and the performance metrics also confirmed the point that EDA is the most appropriate tool to use in the context of contemporary risk management. The rate of detection accuracy has been observed to be more than ninety-eight percent and the use of resources was efficient in all the cloud nodes. This analysis confirms that an institution can attain a proactive and highly responsive risk profile through the treatment of financial information as a continuous flow of events, as opposed to record keeping. The replay ability and fault tolerance characteristics imparted on the architecture makes the data integrity to be preserved even when there is a network failure or a node outage. Altogether, the movement towards the financial sector of event-driven, cloud-native systems is not a luxury anymore but a necessity. The trends presented in this paper give an outline of how to construct robust, scalable, and very accurate risk assessment engines. With the markets becoming ever more dynamic and fast, the capacity to process and act on what is going on in real time will be the hallmark of an efficient financial infrastructure. The results of the research form a solid basis to implement reactive systems in high stakes environments involving a lot of data in the future. The next stage of the development of event-driven risk assessment is the further inclusion of artificial intelligence and machine learning into the event stream. Further studies may be carried out on in-stream model training where the risk engine is trained using new data patterns and does not require the system to be offline or transfer data to another warehouse. This would be able to provide an even more flexible defense mechanism that is able to detect the patterns of zero-day frauds, or market anomalies as they occur. Furthermore, it may further decentralize the system by exploring how mesh event architectures can be used to share events between financial institutions with their peers to discover systemic risks in a more broad manner. One more potential opportunity to expand on is the use of more specialized hardware accelerators, like Field Programmable Gate Arrays (FPGAs), on the cloud environment. Combining them with event-driven software patterns may potentially reduce latencies to the microsecond scale, to meet the most challenging high-frequency trading needs. Moreover, since the regulatory demands on data privacy rise, the study of privacy-sensitive event processing, like homomorphic encryption of the event stream, will be urgent. This would enable risk assessment of encrypted data, which would be compliant and still have the advantages of a distributed cloud-native architecture.

8. REFERENCES

[1] J. Luna, A. Taha, R. Trapero, and N. Suri, "Quantitative Reasoning About Cloud Security Using Service Level Agreements," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2015. DOI: http://dx.doi.org/10.1109/TCC.2015.2469659

[2] E. Cayirci, A. Garaga, A. Santana, and Y. Roudier, "A Cloud Adoption Risk Assessment Model," in *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, pp. 908–913, 2014. DOI:10.1109/UCC.2014.148

[3] E. Cayirci, "A joint trust and risk model for MSaaS mashups," in *2013 Winter Simulations Conference (WSC)*, pp. 1347–1358, 2013. doi: http://dx.doi.org/10.1109/WSC.2013.6721521

[4] S. M. Habib, V. Varadharajan, and M. Muhlhauser, "A Trust-Aware Framework for Evaluating Security Controls of Service Providers in Cloud Marketplaces," in *2013 IEEE 12th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 459–468, 2013. DOI: http://dx.doi.org/10.1109/TrustCom.2013.58

[5] S. M. Habib, S. Ries, and M. Muhlhauser, "Towards a Trust Management System for Cloud Computing," in *2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 933–939, 2011. DOI: http://dx.doi.org/10.1109/TrustCom.2011.129

[6] J. Luna, R. Langenberg, and N. Suri, "Benchmarking cloud security level agreements using quantitative policy trees," in *Proceedings of the 2012 ACM Workshop on Cloud Computing Security Workshop*, pp. 103–112, 2012. DOI: http://dx.doi.org/10.1145/2381913.2381932

[7] N. Kratzke and P.-C. Quint, "Understanding cloud-native applications after 10 years of cloud computing: a systematic mapping study," *Journal of Systems and Software*, vol. 126, pp. 1–16, 2017. DOI: https://doi.org/10.1016/j.jss.2017.01.001

[8] K. Priyadarsini, E. F. I. Raj, A. Y. Begum, and V. Shanmugasundaram, "Comparing DevOps procedures from the context of a systems engineer," *Materials Today: Proceedings*, 2020. DOI: https://doi.org/10.1016/j.matpr.2020.09.624

[9] O. Tomarchio, D. Calcaterra, and G. D. Modica, "Cloud resource orchestration in the multi-cloud landscape: a systematic review of existing frameworks," *Journal of Cloud Computing*, vol. 9, p. 49, 2020. DOI: https://doi.org/10.1186/s13677-020-00194-7

[10] M. Liaqat, V. Chang, A. Gani, S. H. A. Hamid, M. Toseef, U. Shoaib, and R. L. Ali, "Federated cloud resource management: review and discussion," *Journal of Network and Computer Applications*, vol. 77, pp. 87–105, 2017. DOI: https://doi.org/10.1016/j.jnca.2016.10.008

[11] M. Chiregi and N. Jafari Navimipour, "Cloud computing and trust evaluation: a systematic literature review of the state-of-the-art mechanisms," *Journal of Electrical Systems and Information Technology*, vol. 5, pp. 608–622, 2018. DOI: https://doi.org/10.1016/j.jesit.2017.09.001

[12] M. Alexandre, T. C. Silva, C. Connaughton, and F. A. Rodrigues, "The drivers of systemic risk in financial networks: a data-driven machine learning analysis," *Chaos, Solitons & Fractals*, vol. 152, p. 111588, 2021. DOI: https://doi.org/10.1016/j.chaos.2021.111588

[13] A. Vijayalakshmi and Hridya, "Functionalities and approaches of multi-cloud environment," in *Operationalizing Multi-Cloud Environments*, Springer, 2022, pp. 257–268. https://link.springer.com/chapter/10.1007/978-3-030-74402-1_1