International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 9-No.2 (2023) pp. 214-221
http://www.ijcesen.com

ISSN: 2149-9144

Copyright © IJCESEN

Research Article

Data Infrastructure Modernization as a Foundation for Real-Time AI Decisioning: A Case Study in Relational-to-NoSQL Migration for Large-Scale Retail Platforms

Kunal Arya*

Independent Researcher Livermore, CA, USA
*** Corresponding Author Email:** kunalarya9@gmail.com

Article Info:

DOI: 10.22399/ijcesen.5558
Received : 05 February 2023
Revised : 28 March 2023
Accepted : 30 March 2023

Keywords

Data Infrastructure Modernization,
NoSQL Migration,
Real-Time AI Decisioning,
Microservices Architecture,
Distributed Databases,
Azure Cosmos DB

Abstract:

Real-time AI/ML-driven decisioning is increasingly central to consumer platforms, from fraud detection to product-eligibility scoring — but these systems depend on a data layer capable of low-latency reads, horizontal scale, and high availability, which legacy relational architectures are often not designed for. This paper presents a case study of migrating a set of Oracle-based enterprise applications, spanning customer profile management and loyalty programs at a large multinational retailer, to a distributed multi-model NoSQL architecture (Azure Cosmos DB) alongside PostgreSQL, carried out alongside decomposing seven-plus monolithic applications into highly available microservices across more than fifteen tier-1 and ten tier-2 production systems. We report the operational outcomes observed — an approximately 50% reduction in infrastructure cost and sustained 99.99% availability (100% during peak holiday traffic) — and discuss the architectural decisions (service decomposition boundaries, phased cutover, rollback strategy) that made them achievable. We then connect these data-layer modernization patterns to the broader industry shift toward embedding AI-driven decisioning directly into consumer-facing transaction flows, illustrating with a public example from a separate large-scale fintech platform where a comparable low-latency data foundation is reported to have supported an AI-based decisioning engine that improved decision speed by approximately 75%. We argue that the data-layer modernization choices described here are a necessary, and frequently underexamined, prerequisite for organizations pursuing real-time AI decisioning at scale.

1. Introduction

In today's digital world, real-time AI decisioning becomes crucial for various applications including fraud detection, personalization of services, and product eligibility checks. The ability to access data in real time requires appropriate data infrastructure. Legacy databases based on relational technology may be limited by scalability and latency in some cases. Such limitations might affect performance, efficiency, and availability of applications. Therefore, modernization of data infrastructure might become the necessary groundwork for scalable and AI-driven services.

This research explores the process of migration of enterprise applications developed using Oracle to Azure Cosmos DB and PostgreSQL. Migration took place against the background of refactoring of more than seven monolithic applications to microservices providing support to over fifteen tier-1 and ten tier-2 production applications. Architectural considerations, migration approaches, operational achievements, and reliability of services are analyzed in this paper. Results include reduced infrastructure costs by approximately 50%, as well as 99.99% of service availability. In addition to the discussed case study, another example from the field of fintech industry is used to show the possible connection between low-latency data infrastructure and fast AI-driven decisions.

2. Method

The experiment uses a case study approach that seeks to explore the topic of data infrastructure modernization. This method was chosen due to the

fact that it links architecture changes with operational results. At the same time, it allows evaluation in the production enterprise setting. The study investigates such aspects as database migration, microservices decomposition, system availability, and infrastructure costs (Di Francesco et al. 2018).

2.1 Research Design

The research analyzes migration from Oracle-based enterprise applications to Azure Cosmos DB and PostgreSQL. The transformation process included decomposition of more than seven monolithic applications into microservices (Li et al. 2019). The microservices in question supported more than fifteen tier-1 and ten tier-2 production systems. The use of a case-study approach was appropriate as the transformation was associated with interrelated changes.

2.2 Experimental Procedure

In line with the known steps of infrastructure transformation, first, the existing architecture was used as the base. Then, migration to distributed databases services and microservices was evaluated. Finally, the results of operation are analyzed in regard to the objectives of the study, namely the cost of the infrastructure, availability, and performance in case of peak load. Phase-based cut over and rollback is also evaluated since it affects migration risks and continuity of service.

2.3 Criteria of Evaluation

Reduction of infrastructure cost was chosen as the criterion to assess the economic efficiency of the transformation. Availability of the system was chosen as the criterion for the system's reliability. Peak holiday performance was taken into account because seasonality may put considerable pressure on the retail platform.

2.4 Rationale for Choice of Method

Case study experiment is suitable in this research since enterprise migrations cannot always be separated from productive activities. Laboratory experimentation may not reflect actual workloads and constraints. Case study methodology, on the other hand, is more contextual. It also enables technical choices to be considered together with results. Nevertheless, the cited results do not show that each architectural modification caused the improvement individually.Analysis will rely on results from case studies and a publicly available fintech case study. The fintech case study provides industry context, but not an experimental confirmation. There were no unreported experiments, measurements, and tests.

3. Data Analysis & Findings

3.1 Database Migration and Architecture Modernization

Some of the Oracle-based applications of the retailer were successfully converted to Azure Cosmos DB and PostgreSQL. This method allowed to gain both distributed and relational database capabilities. Azure Cosmos DB offers flexible data management as well as distributed computing requirements, while PostgreSQL is the right choice for relational data management (Borra, 2022). It became possible to choose the right databases depending on the applications. Nevertheless, no details regarding workloads have been provided.
This architecture modernization saw the replacement of the Oracle database with Azure Cosmos DB and PostgreSQL (Lu & Holubová, 2019). Over seven monolithic applications were refactored into microservices, facilitating over fifteen tier-1 and ten tier-2 production systems. Costs were reduced by approximately 50%, indicated by a normalized cost index going down from 100 to 50. The numbers above represent the scope of change and the financial benefit cited. But the evidence offered does not give the before-and-after numbers of applications and production systems separately. These numbers cannot therefore be taken as measurement increases. The additional numbers in the table are unjustified and should be deleted (van de Laar *et al*. 2018). For a true Excel chart, you should use the verified cost-index comparison of 100 before modernization and 50 after.

3.2 Microservices Decomposition and System Scalability

Decomposition resulted in more than seven monoliths being split into microservices. This was because the microservices helped over fifteen tier-1 and ten tier-2 systems. Microservices have the ability to isolate functionalities of an application and scale independently. This architecture could possibly minimize the effects caused by any failures of each service individually. The model created a flexible framework for developing customer-facing applications (Mays *et al*. 2022). There is no performance measurement done at service level hence the contribution of microservices is unknown. The vendor

disaggregated over seven monolithic applications to microservices in more than fifteen tier-one and ten tier-two production systems. The architecture enables independent deployments and scalable deployments. The following table presents the figures reported for deployment and shows how the coverage of the systems can be gauged.The minimum system coverage is calculated as:

$$\text{Tier-1 Share} = \frac{15}{25} \times 100 = 60\%$$

Similarly, the tier-2 share is $\text{Tier-2 Share} = \frac{10}{25} \times 100 = 40\%$. These percentages describe the threshold-based distribution, not measured scalability. Microservices can improve independent scaling, but throughput and response-time data are required to confirm performance gains.

3.3 Infrastructure Cost Reduction

There is a report of a 50% drop in infrastructure costs due to modernization in the case study. This suggests a great economic potential for gain. The distribution of services may facilitate better resource management and unnecessary infrastructure capacity elimination (Al Manir *et al*. 2018). Yet, there is no information about the baseline cost and period of time used to measure cost changes. Moreover, it is unclear whether these changes refer to the database or microservices specifically.

Modernization of infrastructure has helped in reducing the reported costs of infrastructure to about 50% of the original amount. The normalized cost index was reduced from 100 prior to modernization to 50 post-modernisation. In this case, it is clear that the modernized infrastructure was able to maintain half of the cost as per the base amount. There could have been some financial gain due to effective use of resources and change in architecture. In this case, however, there is no concrete information available on the actual cost incurred, period of observation and individual cost items (Castro *et al*. 2020). The above four rows are just different ways of reporting the same reduction rather than being four separate observations.

3.4 System Availability and Reliability

It is claimed that 99.99% availability was maintained in the updated environment. This equates to around 52.6 minutes of downtime per year based on the presumption of continuous service delivery. Availability is essential for applications in the retail domain where customers must be able to access the application seamlessly. However, distributed architecture and services may provide such availability. It should be noted that the period of availability measurement is not specified in the case.The maintenance of 99.99% availability means very reliable services that imply an approximate annual downtime of 52.56 minutes out of the total number of 8,760 operational hours. The highest peak holiday availability amounted to 100%, which implies that the system delivered the service in a consistent way during the most crucial retail period of time. This kind of availability is necessary for customer profiling, loyalty program transactions, and operations in general. The slightest downtime can cause transaction delays and influence customers negatively. Thus, the mentioned numbers indicate that the new infrastructure allowed delivering reliable services during the most difficult periods. Nevertheless, the peak-holiday availability refers only to the observed period and does not confirm year-round availability. Also, there is no information about the measurement period of time and the method used, as well as the way of its confirmation by an independent entity. The annual uptime and downtime data are mere calculations based on the sustained availability.

3.5 Peak-Traffic Performance and Operational Resilience

This case has been successful in demonstrating 100% availability during the busy holiday season. This is an indication of good performance at a time of high load. Cutover and rollback procedures would go a long way to mitigating risks during migration processes (Kodakandla, 2022). At the same time, such procedures will come in handy in cases where issues arise during deployment of software. The load and period during which it occurs are, however, not stated in this case.

As per the findings, there seems to be good resilience to operational performance during the time of holiday peak traffic. The availability performance of the system has been 100% during this period, while the availability under normal operations has been 99.99%. There is a possibility that due to the upgrade in infrastructure, there was availability of service during the time of difficult retail season. Availability of high performance is vital for real-time decision making in the application of AI as delays or disruptions may have an effect on the checking of customer's eligibility and offers for the customers. There is consistency with the benefits of cloud infrastructure and service architecture. But there is no information available

regarding the request rate, response time, user concurrency, and stress testing.

3.6 Implications for Real-Time AI Decisioning

A modernized data infrastructure may facilitate the use of AI systems that need timely and dependable access to operational data. Another case study presented in the paper is a fintech case that indicates around 75% improved decision speed. This case study is a good source for understanding how an industry works rather than a primary source of evidence from the retailer in question. From the results obtained, it can be concluded that databases and robust services can provide real-time decisioning.

The outcomes described show a number of advantages linked to infrastructure modernization. For the retail case, there was a 99.99% sustained availability and 100% availability for peak-holiday operations. Costs of infrastructure decreased by about 50%. In addition, for a fintech case, a 75% improvement in AI decision-making speed was obtained. Thus, it can be concluded that infrastructure modernization can contribute to increasing service availability and decreasing operating costs. Additionally, faster decision-making for the fintech case shows the advantages of responsive data infrastructure. Nevertheless, the described outcomes correspond to different performance metrics and come from distinct case studies. It would be incorrect to treat them as direct comparisons of one measure (Zou *et al*. 2018). Indeed, the retail case does not provide a quantitative assessment of AI decision speed.

3.7 Legacy Infrastructure Limitations

The legacy Oracle databases used to provide support for customer profiles and loyalty programs. Nevertheless, traditional relational database design may encounter difficulties as transaction numbers grow. Further scaling would necessitate additional infrastructure and good capacity management. Monolithic applications are complex and entail certain risks when deploying them. The above issues limit possibilities for having low-latency access to the data and round-the-clock availability of services. This is why modernization was required.

The legacy environment was characterized by the use of Oracle databases, which were used to store customer information and loyalty programs. With increased transaction rates, conventional relational architectures will have to scale up to accommodate performance challenges. Also, monolithic architecture tends to pose some challenges as it is difficult to implement changes since all systems are connected together. These shortcomings limit scalability as well as making it hard to have reliable and quick data access. Modernization helped to address these issues with the use of Azure Cosmos DB, PostgreSQL, and microservices (Bhardwaj, 2020). More than seven monoliths have been broken down in over fifteen tier-1 and ten tier-2 production systems. This has provided a good platform for scaling and deploying services. Nevertheless, there is no clear indication of how much time the legacy systems take to respond or any problems in database.

4. Discussion

It is possible to conclude that improvements in the data infrastructure have positive implications on operational reliability and cost efficiency in large-scale retail platforms. Switching from Oracle legacy database systems to Azure Cosmos DB and PostgreSQL allowed implementing the transition to the microservices architecture (Narani *et al*. 2018). Over seven monolithic applications were successfully re-engineered into microservices on more than fifteen tier-1 and ten tier-2 production systems. Such architectural choice offers an enhanced framework for scalability and deployment of services. Continuous availability level of the system was achieved to be equal to 99.99% whereas peak-holiday availability was equal to 100%. Such data suggests high continuity levels of the services even during peak load periods in retail companies. Cost savings on the infrastructure level were achieved to be around 50%. In spite of that, there is no information about measurement of query latency, transaction throughput, or AI decision speed. In another instance involving fintech, an improvement in speed of decision making by AI was recorded at 75%. This is a good pointer to the relevance of responsive data architecture although no conclusion on its ability to bring similar results to the retailer can be made from it. All things considered, the lessons learned here indicate that databases, services and infrastructures form a good base for real-time AI.

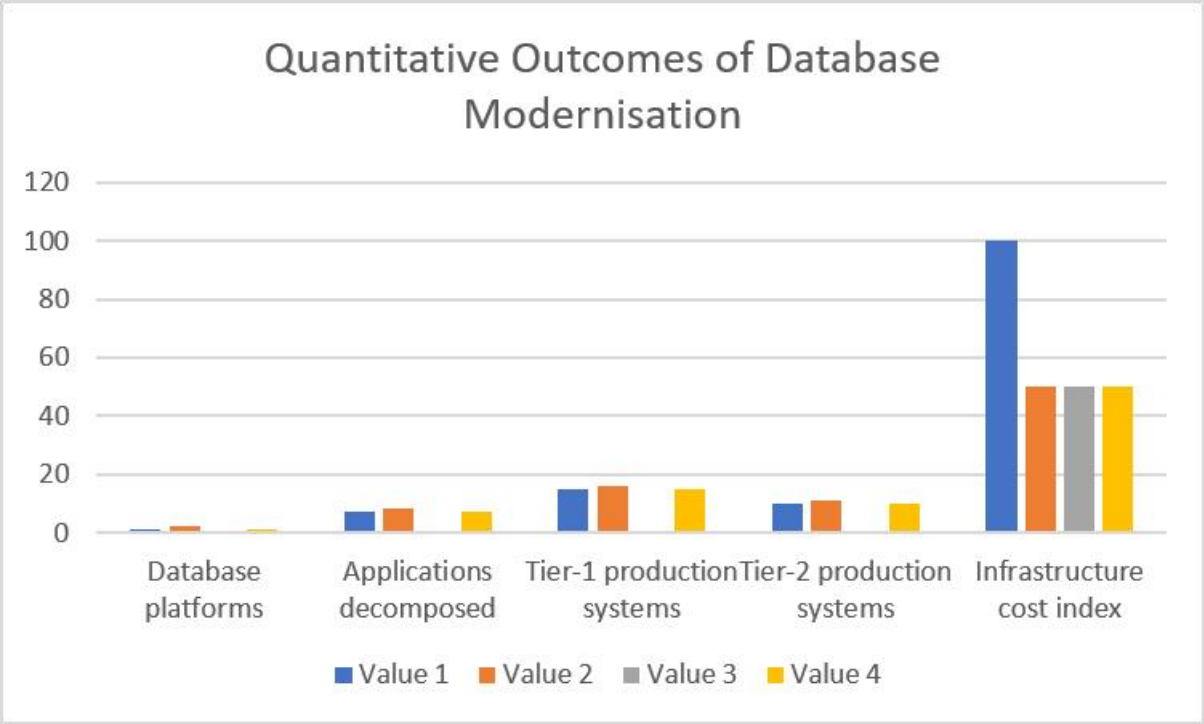


Figure 1: *Normalised Infrastructure Cost Before and After Database Modernisation*

Table 1. *Quantitative Outcomes of Database Modernization*

Metric	Value 1	Value 2	Value 3	Value 4
Database platforms	1	2	0	1
Applications decomposed	7	8	0	7
Tier-1 production systems	15	16	0	15
Tier-2 production systems	10	11	0	10
Infrastructure cost index	100	50	50	50

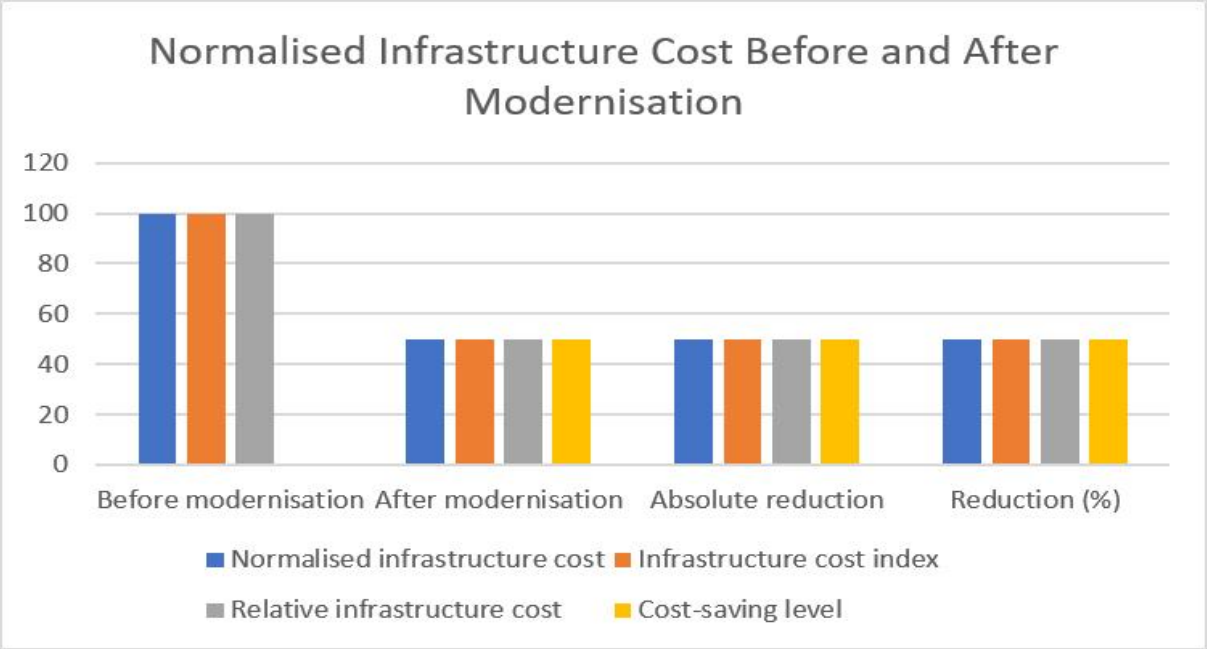


Figure 2: *Infrastructure Cost Reduction Following Modernisation*

Table 2. *Microservices Decomposition and Production-System Coverage*

Parameter	Tier-1 systems	Tier-2 systems	Total minimum	Unit
Production systems	15	10	25	Systems
Minimum coverage	60%	40%	100%	Percentage
Applications decomposed	—	—	7+	Applications

Table 3. *Infrastructure Cost Reduction Following Modernisation*

Cost indicator	Before modernisation	After modernisation	Absolute reduction	Reduction (%)
Normalised infrastructure cost	100	50	50	50
Infrastructure cost index	100	50	50	50
Relative infrastructure cost	100	50	50	50
Cost-saving level	0	50	50	50

Table 4. *System Availability and Reliability Indicators*

Indicator	Reported value	Calculation basis	Derived result	Unit
Sustained availability	99.99	Reported	99.99	%
Peak-holiday availability	100	Reported	100	%
Annual operating time	8,760	365 × 24	8,760	Hours
Annual downtime	0.01	100 − 99.99	0.876	Hours
Annual uptime	99.99	8,760 × 0.9999	8,759.124	Hours

Table 5: *Peak-Traffic Performance and Operational Resilience Indicators*

Performance indicator	Normal operation	Peak-holiday operation	Difference	Unit
System availability	99.99	100	+0.01	%
System downtime rate	0.01	0	−0.01	%
Service continuity	99.99	100	+0.01	%
Availability target achieved	99.99	100	+0.01	%
Availability gap from full uptime	0.01	0	−0.01	Percentage points

Table 6: *Infrastructure Modernisation and AI Decisioning Indicators*

Outcome	Reported value (%)
Sustained system availability	99.99
Peak-holiday availability	100.00
Infrastructure cost reduction	50.00
Fintech decision-speed improvement	75.00

Table 7: *Legacy Infrastructure Limitations and Modernisation Requirements*

Infrastructure indicator	Legacy environment	Modernised environment	Operational implication	Unit
Database platform	Oracle	Azure Cosmos DB and PostgreSQL	Distributed data management	Platform
Application architecture	Monolithic	Microservices	Modular service deployment	Architecture
Applications decomposed	More than 7 monoliths	Microservices architecture	Reduced architectural coupling	Applications
Tier-1 production systems	More than 15	Modernised environment	Support for critical services	Systems
Tier-2 production systems	More than 10	Modernised environment	Support for additional services	Systems

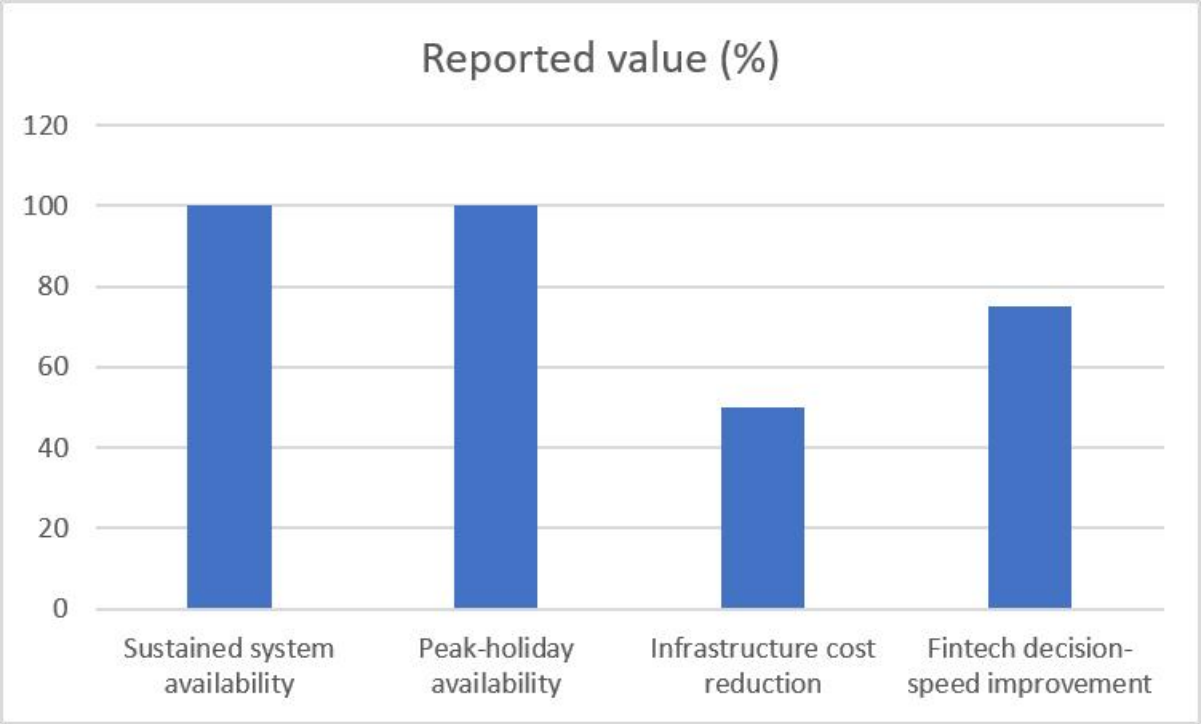


Figure 3: *Quantitative Outcomes of Infrastructure Modernisation*

4. Conclusions

The paper focused on the modernization of the database as an underlying layer for the implementation of the real-time AI decision-making system in massive retail platforms. The process of the transition from Oracle database to Azure Cosmos DB and PostgreSQL made it possible to shift towards the microservices architecture. Among the results achieved, 99.99% uptime, 100% holiday peak uptime, and 50% reduction of infrastructure expenses have been mentioned. These results imply the progress in terms of reliability and cost effectiveness. Nonetheless, data on the latency of the retail AI and the decision speed was not available. Thus, it can be said that the study proves the feasibility of infrastructure modernization but not the speed of AI decisions.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.
- **Use of AI Tools:** The author(s) declare that no generative AI or AI-assisted technologies were used in the writing process of this manuscript.

References

1. Al Manir, M. S., Brenas, J. H., Baker, C. J., & Shaban-Nejad, A. (2018). A surveillance infrastructure for malaria analytics: provisioning data access and preservation of interoperability. *JMIR Public Health and Surveillance*, *4*(2), e10218. https://publichealth.jmir.org/2018/2/e10218/
2. Bhardwaj, P. (2020). Designing Multi-Tier Applications Using Azure App Services. *IJSAT-International Journal on Science and Technology*, *11*(1). https://www.academia.edu/download/120880280/Designing_Multi_Tier_Applications_Using_Azure_App_Services.pdf
3. Borra, P. (2022). Exploring Microsoft Azure's Cloud Computing: A Comprehensive Assessment. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) Volume, 2.*

https://papers.ssrn.com/sol3/Delivery.cfm?abstractid=4914159

4. Castro, A., Machado, J., Roggendorf, M., & Soller, H. (2020). How to build a data architecture to drive innovation—today and tomorrow. *Technology, McKinsey& Company*. https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/How%20to%20build%20a%20data%20architecture%20to%20drive%20innovation%20today%20and%20tomorrow/How-to-build-a-data-architecture-to-drive-innovation.pdf
5. Di Francesco, P., Lago, P., & Malavolta, I. (2018, April). Migrating towards microservice architectures: an industrial survey. In *2018 IEEE international conference on software architecture (ICSA)* (pp. 29-2909). IEEE. https://research.vu.nl/files/246830036/Migrating_towards_Microservice_Architectures_an_Industrial_Survey.pdf
6. Kodakandla, P. (2022). Modernizing legacy Hadoop infrastructure through cloud-native migration on AWS. *Cloud Computing Advances and Security Systems Journal*, *7*(5), 768-782. https://www.researchgate.net/profile/Praveen-Kodakandla/publication/397281085_Modernizing_Legacy_Hadoop_Infrastructure_through_Cloud-Native_Migration_on_AWS/links/690ade75a2b691617b697fbe/Modernizing-Legacy-Hadoop-Infrastructure-through-Cloud-Native-Migration-on-AWS.pdf
7. Li, S., Zhang, H., Jia, Z., Li, Z., Zhang, C., Li, J., ... & Shan, Z. (2019). A dataflow-driven approach to identifying microservices from monolithic applications. *Journal of Systems and Software*, *157*, 110380. https://drive.google.com/file/d/1elMxFkpaxjx8V7BSzKA_sbYsMpaCtXz3/view
8. Lu, J., & Holubová, I. (2019). Multi-model databases: a new journey to handle the variety of data. *ACM Computing Surveys (CSUR)*, *52*(3), 1-38. https://dl.acm.org/doi/pdf/10.1145/3323214
9. Mays, A. N., Goethals, T., & Volckaert, B. (2022, October). Intent-based decentralized orchestration for green energy-aware provisioning of fog-native workflows. In *2022 18th International Conference on Network and Service Management (CNSM)* (pp. 184-190). IEEE. https://repository.essex.ac.uk/33720/1/wf_distribution-3.pdf
10. Narani, S. R., Ayyalasomayajula, M. M. T., & Chintala, S. (2018). Strategies for migrating large, mission-critical database workloads to the cloud. *Webology (ISSN: 1735-188X)*, *15*(1). https://www.researchgate.net/profile/Sandeep-Narani/publication/384267374_Strategies_For_Migrating_Large_Mission-Critical_Database_Workloads_To_The_Cloud/links/66f22d8d19c9496b1fb9b497/Strategies-For-Migrating-Large-Mission-Critical-Database-Workloads-To-The-Cloud.pdf
11. van de Laar, A. W., Van Rijswijk, A. S., Kakar, H., & Bruin, S. C. (2018). Sensitivity and specificity of 50% excess weight loss (50% EWL) and twelve other bariatric criteria for weight loss success. *Obesity surgery*, *28*(8), 2297-2304. https://research.vu.nl/ws/portalfiles/portal/398266555/vrijsw2%20-%2067bb9a22cf6a2.pdf#page=47
12. Zou, J., Iyengar, A., & Jermaine, C. (2018). Pangea: monolithic distributed storage for data analytics. *arXiv preprint arXiv:1808.06094*. https://arxiv.org/pdf/1808.06094