Development of a Prototype for Microservices Architecture and API Gateway Integration in an eLearning Platform

Dwi Ely Kurniawan[1], Afdhol Dzikri[2]

{dwialikhs@polibatam.ac.id[1], afdhol@polibatam.ac.id[2]}

Department of Informatics Engineering, Politeknik Negeri Batam, Indonesia[1,2]

Abstract. The educational access gap between hinterland and urban areas often creates disparities in educational opportunities and economic development. E-learning can reduce the isolation of remote regions by providing access to information and learning. Microservices architecture can enhance efficiency in education delivery. This research aims to develop a prototype of microservices architecture and API Gateway integration in an eLearning platform. The research methodology involves prototype design, data modeling, software development, integration with the eLearning platform, and performance testing. The results of the study demonstrate that the integration of microservices architecture and API Gateway improves platform flexibility and scalability, enabling users in hinterland areas to access online education more effectively, overcoming infrastructure constraints, and opening new opportunities for inclusive and effective education delivery.

Keywords: Microservices, eLearning, API Gateway.

1 Introduction

In the continually evolving digital era, access to quality education is becoming increasingly crucial for the development of society and a country's economy. Information and Communication Technology (ICT) has opened significant opportunities to expand educational access through e-learning platforms [1], [2] . However, in many remote or hinterland regions, accessibility challenges, including limited internet access and inadequate educational infrastructure, remain significant obstacles to realizing the potential of e-learning. Education is a key factor in addressing poverty and social inequality. Hinterland regions often face inequalities in educational access, and e-learning can be a tool to address these disparities [3]. Hinterland areas often grapple with serious issues related to limited internet access. Slow speeds, unstable connections, and even a lack of internet access are problems faced by a substantial portion of the population in these areas. This hampers their ability to access e-learning resources. Schools in hinterland regions often lack adequate infrastructure to support digital learning [4].

ICAE 2023, November 07, Batam, Indonesia

DOI 10.4108/eai.7-11-2023.2342930

The shortage of computers, a robust network, and trained educators pose barriers to effective e-learning implementation. Limited access to computer hardware and software also presents obstacles. Many residents of hinterland regions may not have sufficient devices to participate in online learning.

E-learning can reduce the isolation of remote areas [5] by providing access to information and learning from around the world [6]. This can improve the quality of life and connect hinterland regions with the outside world. Disparities in educational access between hinterland and urban areas often create inequalities in educational opportunities and economic development [7] [8]. Therefore, it is important to seek solutions to overcome these barriers.

Previous research in this field has highlighted some issues and efforts to improve e-learning [9] [10] access in hinterland regions. However, most of this research has yet to address the fundamental issues of limited internet access and inadequate infrastructure [11]. Some studies have identified internet access challenges in hinterland regions [4] [12], but few have developed concrete solutions. Additionally, some research has focused on developing e-learning content suitable for the local context in hinterland regions, but accessibility remains an issue. Initiatives to build network infrastructure in hinterland regions have been attempted, but much more needs to be done to comprehensively address these issues.

Microservices technology can enhance efficiency in education delivery [13] [14]. With this technology, online learning can be tailored to individual needs, resources can be utilized more effectively, and content updates can be made rapidly. Therefore, this research will develop microservices prototypes on the e-learning platform. This research has significant scientific contributions as it aims to address e-learning access issues in hinterland regions through the development of microservices architecture prototypes. By integrating microservices technology, this research aims to improve the efficiency and availability of e-learning platforms in areas with limited internet access.

The results of this research are expected to provide concrete solutions for addressing e-learning access issues in hinterland regions, which can be adopted by governments, educational institutions, and other stakeholders. Additionally, this research will provide deeper insights into how technology can be used to address educational access inequalities in remote areas, with significant social and economic implications. Enabling e-learning access in hinterland regions can trigger economic and social development. Improved educational access can help residents acquire better skills, enhancing job opportunities and their contributions to economic growth. E-learning provides individuals with opportunities to develop their skills and knowledge independently. By offering better e-learning access in hinterland regions, residents can gain skills that can enhance their quality of life. Microservices architecture can enhance efficiency in education delivery, allowing online learning to be tailored to individual needs, resources to be used more effectively, and content updates to be made rapidly.

2 Research Method

The research method utilizes a prototyping approach to develop an eLearning platform design by integrating microservices architecture and API Gateway [15], and then testing the developed eLearning platform. The main steps in this research methodology can be seen in Figure 1.

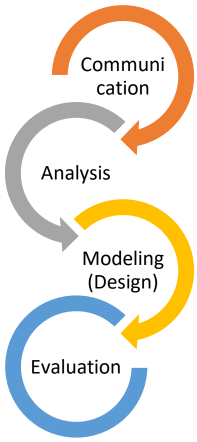

Fig. 1. Research Method

The first step is to engage in user communication and conduct comprehensive literature research on relevant topics. This encompasses literature on microservices architecture, e-learning, challenges of internet access in hinterland regions, and related technological solutions. The goal of this step is to understand the conceptual framework, related research, and recent trends. Next, analyze the requirements in designing a microservices architecture prototype that will be used in the development of the eLearning platform, including infrastructure modeling, service definition, and integration planning with the API Gateway, as well as configuring the API Gateway to manage requests and responses.

Based on the knowledge from literature research and requirement analysis, the next stage commences with the development of the microservices architecture prototype. This includes modeling, infrastructure design, and software development tailored to the needs. The microservices architecture prototype will be implemented on a small scale for the eLearning Prototype to measure performance, through testing aspects such as access speed, scalability, and interoperability.

The collected data will be analyzed to assess the extent to which the prototype has successfully integrated microservices architecture and API Gateway into the eLearning platform. The analysis will encompass data modeling and comparison with previous baseline data (before prototype implementation). Evaluation will be conducted to assess the impact of prototype development on e-learning access.

3 Results and Discussion

Every microservice has functionality related to specific tasks within the overall application. Microservices architecture can be used to build a more flexible, easily updatable, and scalable eLearning platform. A microservice specifically handles course management, including course creation, editing, and settings. This service will have an Application Programming Interface (API) that allows interaction with other components within the system. This microservice is responsible for user management, which includes registration, authentication, authorization, and user profile management. Each user has an account accessible through the API. To manage learning content, a dedicated microservice can be provided for uploading, storing, and managing learning materials such as videos, documents, or images. Users can access the content through the interface provided by this service. A microservice can also handle communication tools between students and instructors, such as live chat, discussion forums, or email notifications, enabling better interaction within the eLearning platform. Additionally, to track student progress, a microservice can manage data related to completed assignments, exam results, and other statistics. This will provide the necessary information for monitoring and reporting on student progress.

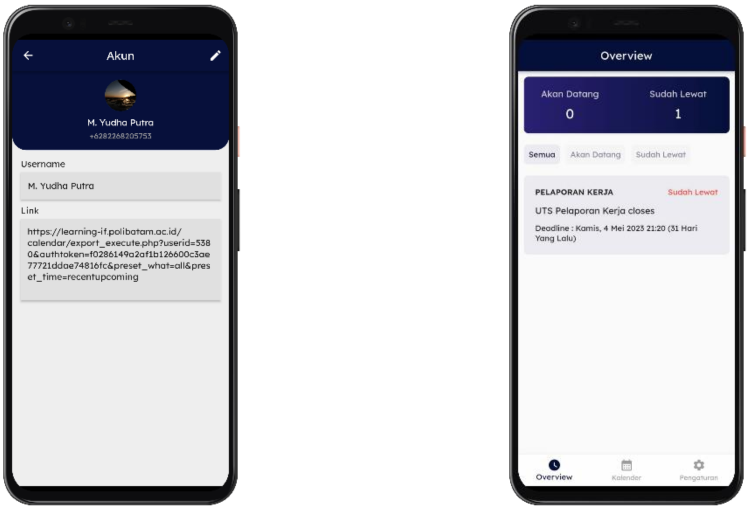

Fig. 2. Microservices in User Notification Delivery

Figure 2 represents a microservice that can handle the delivery of notifications and alerts to users, for example, when there are course updates or impending assignments, based on each user's API. By using a microservices architecture, each of these services can be developed, updated, and managed independently. This allows for faster feature releases or improvements without affecting the entire eLearning platform. Furthermore, in case of a surge in usage or scalability needs, it's easy to increase the number of copies of the corresponding services without modifying the entire platform. This provides great flexibility in the development and management of the eLearning platform.

The microservices architecture does not limit how small or large a microservice can be, but rather focuses on separating specific functions within an application. A microservice should be designed to perform a single task or functionality well on a smaller scale compared to a monolithic application. The primary goal of using microservices is to break down the application into smaller parts that can be managed, updated, and improved independently.

Communication between services can be done through protocols like HTTP/HTTPS, message protocols, or application programming interfaces (APIs). The illustration of communication between services in the context of the eLearning web can be described as follows.

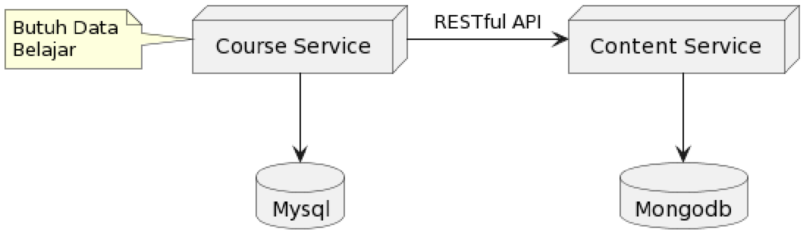

Fig. 3. Communication Between Services

Figure 3 shows that communication through a Restful API can be used, for example, to access user data, course data, or student progress data without the need to directly access the database. This helps ensure data security, better error management, and separation of responsibilities between the application layer and the database layer. Communication between these services can be done through various mechanisms, including HTTP requests (usually with methods like GET, POST, PUT, DELETE), the use of application programming interfaces (APIs), or even message protocols like MQTT or Apache Kafka if real-time communication is required [16] [17]. Each microservice can work independently and communicate with each other to provide the full features of the eLearning platform, enabling scalability, flexibility, and better maintenance in a distributed environment. Implementing an API Gateway is a common practice to control and manage access to various microservices within the application. The API Gateway serves as an intermediary layer between clients (such as user devices, applications, or client software) and the microservices behind it.

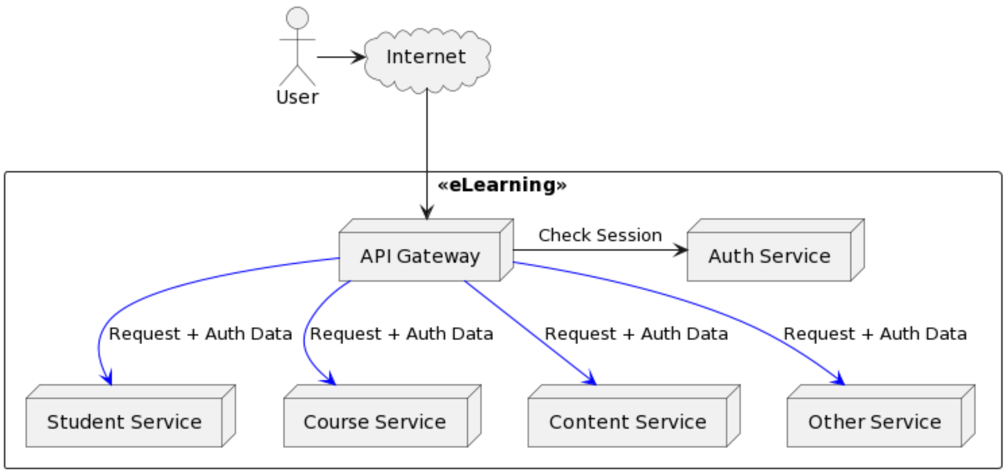

Fig. 4. API Gateway and Auth Service as Middleware

Figure 4 shows integration with the Auth Service, which enables efficient and secure management of authentication and authorization in the eLearning web application. Furthermore, it allows users to have a better and safer experience when using the eLearning platform. The proper use of middleware in each microservice within the eLearning web architecture can enhance the overall functionality, security, and performance of the application. However, it's important to remember that the use of middleware also requires good management and maintenance to prevent it from becoming a source of issues.

Ensuring consistency in user interface (UI) design across the entire frontend can be a challenging task. Each frontend may have different appearances, resulting in inconsistent user experiences. Effective permission and security management are required for each frontend, which can add complexity, especially if multiple frontends need to access the same data with different permissions. To address this issue, the selection of Backend for Frontend (BFF) technology is necessary [18]. Backend for Frontend (BFF) is an architectural pattern used in the development of microservices-based applications, including in the context of eLearning web architecture. The main idea behind BFF is that each user interface (UI) has a specific backend designed to meet the needs of that UI. In the context of eLearning web architecture, BFF can be used to deliver a more efficient and optimized user interface tailored to the different features and views of eLearning.

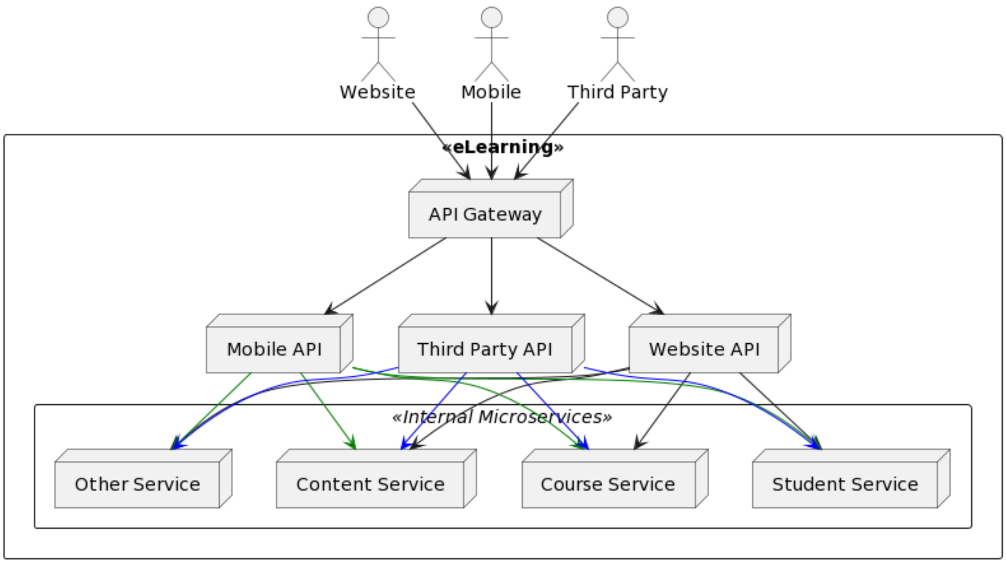

Fig. 5. Backend for Frontend

Figure 5 shows an illustration of BFF in the context of elearning implementation. Various views may be required, including the homepage, course page, user profile page, and others. Each view may require access to different data, different business logic, and different presentation. With BFF, it is possible to have a dedicated backend for each of these views. Therefore, BFF enables the optimization of the performance of each view by designing a backend that suits its needs. For example, if the homepage requires a large amount of data, you can design its BFF to efficiently fetch and serve the data without burdening other views. Additionally, BFF can separate development tasks between the frontend and backend more clearly [19]. The frontend development team can focus on the design of the user interface and user interactions, while the backend development team can focus on business logic and data retrieval. BFF can also handle authentication and authorization specifically for each view. For instance, it can control who can access specific views and provide the appropriate level of permissions.

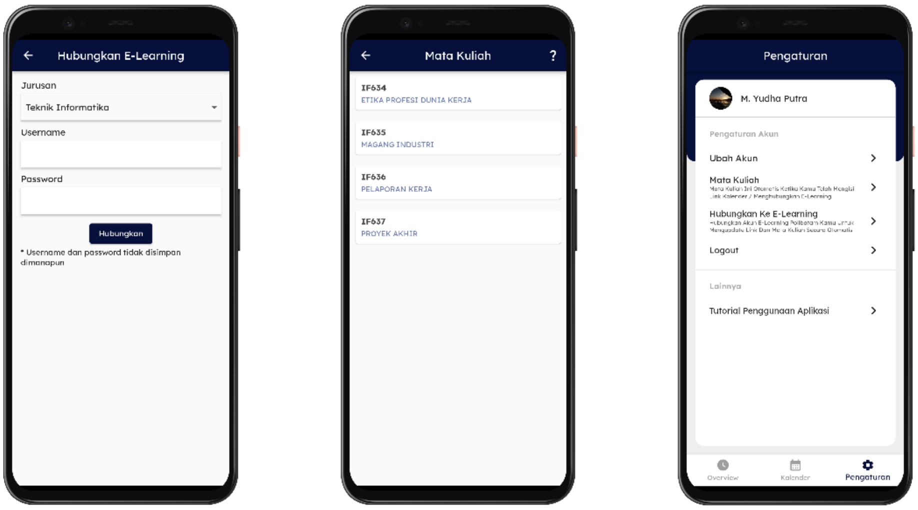

Fig. 6. Mobile eLearning Access

Figure 6 is the frontend view for displaying user login, courses, and course settings. The BFF (Backend for Frontend) can manage scalability more efficiently by adding or removing backend components specific to the user's request and traffic. BFF can also be used to provide additional services that may be needed by specific views, such as search, notifications, or messaging delivery. After the prototype is created, the next step involves evaluating the performance of the available device resources and network under minimal specifications.

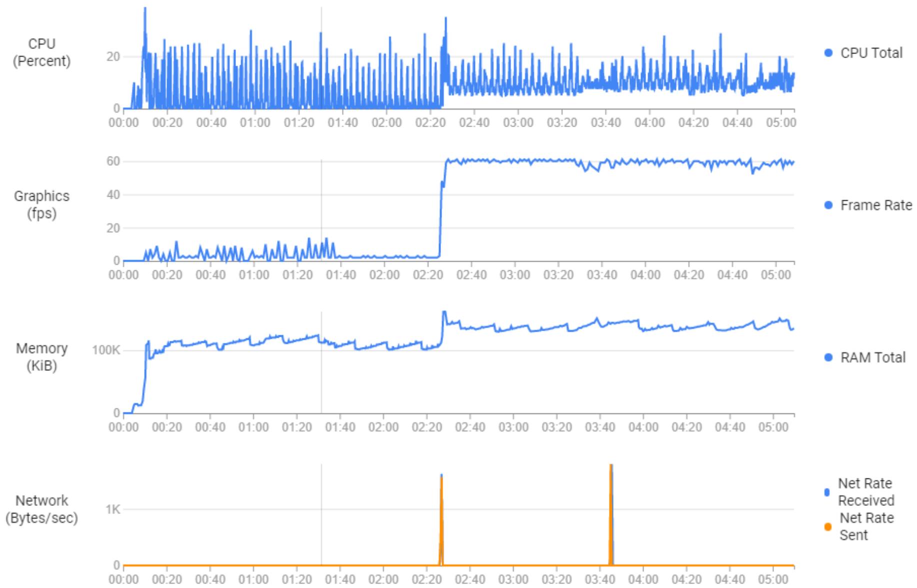

Fig. 7. Mobile Device Resource Performance

Figure 7 depicts the performance of mobile device resources when accessing the Front-End platform of eLearning. On mobile devices, the CPU is utilized at an average rate of only about 20%. The application's graphics run smoothly, achieving frame rates of 55 to 60 FPS after the application starts. The highest RAM usage observed is approximately 161 MB. Therefore, it can be concluded that accessing the eLearning platform does not require significant resources and runs stably on devices with minimal specifications.

Next, to assess network performance, the evaluation is conducted using the Network Profiler application as a tool for developers to monitor and analyze network performance while the application is running. The Network Profiler can simulate various network conditions, such as slow or unstable connections, to test the data communication performance in the Front-End application under different situations.

Table 1. Network Performance Upload and Download Speed Results

Network	Upload		Download	
	Bandwidth	**Delay**	**Bandwidth**	**Delay**
Wifi	360.4 Mbps	1 ms	585.5 Mbps	1 ms
3G	340 kbps	110 ms	790 kbps	110 ms
2G	210 kbps	450 ms	250 kbps	410 ms

Table 1 provides information about how much data can be uploaded and downloaded simultaneously (upload and download speeds) and how quickly data can travel between the device and the network (latency) for each type of network tested. Latency is the time it takes for data to travel back and forth between the device and the server. The lower the latency, the more responsive the application will feel, with a latency of less than 100 ms being considered responsive [20] [21]. The information in Table 1 suggests that the performance of the Front-End application or data communication in microservices varies under different network conditions. For Wifi, the latency is very fast, only 1 ms; for 3G, it is moderate at 110 ms, and for 2G, it is slower at 410 ms. Therefore, it can be concluded that Wifi is recommended for data communication in microservice services.

4 Conclusion

This research aims to address the challenges of e-learning access in remote areas, proposing the use of microservices technology and API Gateway, and highlighting the importance of customized user interfaces through Backend for Frontend (BFF) technology. Microservices technology is utilized to manage various aspects of course management, user management, content, communication, and student progress monitoring within the e-learning platform. The use of API Gateway is employed to control and manage access to these microservices. Additionally, Backend for Frontend (BFF) technology is implemented to optimize the user interface (UI) to cater to various features and displays within the e-learning platform, including websites, mobile devices, and third-party applications. BFF enables scalability management and the provision of additional services tailored to specific UI needs. Performance measurements of the Front-End prototype indicate that the application can be accessed efficiently, with resource utilization such as CPU at approximately 20%, smooth graphics performance, and limited RAM usage (around 161 MB). In terms of network performance and data communication, Wifi is considered the best option due to its very low latency (only 1 ms) and high download speed (585.5 Mbps).

Acknowledgments

This research is a Research Assignment Grant for the Program Directorate of Research and Community Service (DRPM) in 2023, the Applied Research, Information and Communication Technology through Politeknik Negeri Batam.

References

[1] Y.-C. Chang, J.-W. Li, and D.-Y. Huang, "A personalized learning service compatible with moodle e-learning management system," *Applied Sciences*, vol. 12, no. 7, p. 3562, 2022.

[2] P. Campanella, "LMS: Benchmarking ATutor, Moodle and Docebo," in *2022 20th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, IEEE, 2022, pp. 85–90.

[3] R. Bhaumik, "Widening disparities in educational access during COVID-19: A deepening crisis," *The impact of COVID19 on the international education system. IGNOU, New Delhi, India. DOI: https://doi. org/10.51432/978-1-8381524-0-6_10*, 2020, Accessed: Oct. 06, 2023. [Online]. Available: https://www.academia.edu/download/82898442/10-1014.pdf

[4] H. Widyastuti, D. Mulyaningtyas, and U. Brajawidagda, "E-learning readiness in Hinterland of Batam," in *The Second International Conference on Knowledge creation and Intelligent Computing. Bali: KCIC*, 2013.

[5] A. Jafar *et al.*, "Readiness and challenges of e-learning during the COVID-19 pandemic era: A space analysis in Peninsular Malaysia," *International Journal of Environmental Research and Public Health*, vol. 20, no. 2, p. 905, 2023.

[6] J. Roy, "E-Learning Teaching: Barriers for Isolated Learners," *E-Learning-Teaching Strategies and Teachers' Stress in Post Covid-19*, p. 33, 2021.

[7] Y. Rokhayati, N. Z. Jannah, S. Irawan, and D. E. Kurniawan, "Decision Determination of Hinterland Selection Using Analytical Network Process," in *2019 2nd International Conference on Applied Engineering (ICAE)*, 2019, pp. 1–5. doi: 10.1109/ICAE47758.2019.9221825.

[8] D. E. Kurniawan and A. Fatulloh, "Clustering of Social Conditions in Batam, Indonesia Using K-Means Algorithm and Geographic Information System," 2017.

[9] D. A. Bauer, B. Penz, J. Mäkiö, and M. Assaad, "Improvement of an existing microservices architecture for an e-learning platform in stem education," in *AICT 2018: The Fourteenth Advanced International Conference on Telecommunications*, IARIA, 2018.

[10] O. Tsilingeridis and A. Karakasidis, "MILMS: A Microservices-based Learning Management System," in *2020 IEEE International Conference on Big Data (Big Data)*, Atlanta, GA, USA: IEEE, Dec. 2020, pp. 5843–5845. doi: 10.1109/BigData50022.2020.9378285.

[11] A. Garad, A. M. Al-Ansi, and I. N. Qamari, "THE ROLE OF E-LEARNING INFRASTRUCTURE AND COGNITIVE COMPETENCE IN DISTANCE LEARNING EFFECTIVENESS DURING THE COVID-19 PANDEMIC," *Jurnal Cakrawala Pendidikan*, vol. 40, no. 1, Art. no. 1, Feb. 2021, doi: 10.21831/cp.v40i1.33474.

[12] N. Z. Janah, Y. Rokhayati, D. E. Kurniawan, and M. F. Muvariz, "Electronic School Books Dissemination Application for Batam Hinterland Schools," *Advanced Science Letters*, vol. 24, no. 12, pp. 9739–9744, Dec. 2018, doi: 10.1166/asl.2018.13128.

[13] A. Milovanović, "Microservice architecture in E-learning," *E-business technologies conference proceedings*, vol. 1, no. 1, Art. no. 1, Sep. 2021.

[14] Y. Artamonov, I. Golovach, and V. Zymovchenko, “Use analysis of microserves in e-learning system with multi-variant access to educational materials,” *Technology audit and production reserves*, vol. 4, no. 2(60), Art. no. 2(60), Jul. 2021, doi: 10.15587/2706-5448.2021.237760.

[15] S. Gadge and V. Kotwani, “Microservice architecture: API gateway considerations,” *GlobalLogic Organisations, Aug-2017*, 2018.

[16] Srijith, K. B. R, G. N, and A. M. R, “Inter-Service Communication among Microservices using Kafka Connect,” in *2022 IEEE 13th International Conference on Software Engineering and Service Science (ICSESS)*, Oct. 2022, pp. 43–47. doi: 10.1109/ICSESS54813.2022.9930270.

[17] D. Jaramillo, D. V. Nguyen, and R. Smart, “Leveraging microservices architecture by using Docker technology,” in *SoutheastCon 2016*, Mar. 2016, pp. 1–5. doi: 10.1109/SECON.2016.7506647.

[18] S. Alkhodary, “The Evaluation of Using Backend-For-Frontend in a Microservices Environment”, Accessed: Oct. 06, 2023. [Online]. Available: https://lup.lub.lu.se/student-papers/record/9086847/file/9086853.pdf

[19] P. Rattanukul *et al.*, “Microusity: A testing tool for Backends for Frontends (BFF) Microservice Systems.” arXiv, Feb. 22, 2023. Accessed: Oct. 06, 2023. [Online]. Available: http://arxiv.org/abs/2302.11150

[20] D. Bhamare, M. Samaka, A. Erbad, R. Jain, and L. Gupta, “Exploring microservices for enhancing internet QoS,” *Trans Emerging Tel Tech*, vol. 29, no. 11, p. e3445, Nov. 2018, doi: 10.1002/ett.3445.

[21] M. R. S. Sedghpour *et al.*, “HydraGen: A Microservice Benchmark Generator,” in *2023 IEEE 16th International Conference on Cloud Computing (CLOUD)*, IEEE, 2023, pp. 189–200. Accessed: Oct. 09, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10254991/