*ICIC Express Letters*
Volume **16**, Number **9**, September **2022**

ICIC International ©2022 ISSN 1881-803X
pp. 1001–1009

# SERVERLESS MICROSERVICES ARCHITECTURE FOR INDOOR POSITIONING SYSTEM USING BLUETOOTH LOW ENERGY

Hengky Suhendinata and Gede Putra Kusuma

Computer Science Department, BINUS Graduate Program – Master of Computer Science
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
hengky.suhendinata@binus.ac.id; inegara@binus.edu

Received August 2021; accepted November 2021

Abstract. *The development of Indoor Positioning System (IPS) research currently focuses on improving accuracy of indoor positioning systems. The use of complex techniques to improve location accuracy has an impact on the use of resources on smartphones. So offloading techniques are used to overcome the limitations of resources on smartphones by moving computing on smartphones to cloud computing. Serverless is one of the cloud computing technologies that have scalable capabilities by performing auto provision on cloud computing resources. In this paper, we propose a design architecture of serverless microservice for indoor positioning system using Bluetooth low energy. This architecture uses serverless technology with a micro services architecture design that divides services into small parts. The test results of 360 concurrent users per second with multitenant showed availability of serverless microservices is 97.83% with an average response time of 4.63 seconds, which is better compared to availability of serverless monolith of 89.68% with an average response time of 4.82 seconds.*
**Keywords:** Serverless microservices, Indoor positioning system, Bluetooth low energy, Computational offloading, Cloud computing

1. **Introduction.** In the field of indoor positioning systems is widely done by researchers because of the limitations of Global Positioning System (GPS) technology when used indoors. It is influenced by the complexity of the building geometry, human movement and other room signal interference factors that result in GPS signals not being received by their users [1]. Several technologies have been proposed for indoor positioning system such as geomagnetic field [2], ultrasound [3], infrared [4], camera [5] and radio.

Bluetooth Low Energy (BLE) technology in indoor positioning systems provides several advantages such as low deployment costs because it uses internal batteries with low power consumption and positioning processes do not require additional hardware because it only requires a smartphone with BLE capabilities and can be widely used on Android and iOS-based smartphones [6]. Modern smartphones are equipped with sensors (inertial sensors, camera and barometer) and communication modules (WiFi, Bluetooth, NFC, LTE/5G, and ultra-wideband), which can be used on indoor positioning system [7].

Several techniques are used to improve location accuracy on BLE such as fingerprinting, RSSI, channel diversity, Kalman filtering, and weighted trilateration. Combining several techniques has been used to improve accuracy and cost savings [8]. The impact of combining some of these techniques will cause complexity to the computing process in the device causing delays to the indoor positioning system [9]. The current IPS research using BLE focuses on improving accuracy due to fluctuation signal of BLE that is received by smartphone [10].

DOI: 10.24507/icicel.16.09.1001

Cloud computing with mobile device computing offloading techniques can overcome problems with the limitations of computing resources and data storage capacity on mobile devices with larger cloud resources. There are 3 cloud computing services, that is, Infrastructure as a Services (IaaS), Platform as a Services (PaaS) and Software as a Service (SaaS). Many applications are moved to the cloud by utilizing IaaS services and PaaS is a monolithic application where applications are built in a large single codebase consisting of various services so it will be a challenge to scaling according to demand or needs [11]. On cloud computing keeping down service delay is key to improve QoE (Quality of Experience) [12].

The microservices architecture approach is the development of applications into a small and independent suite of services with each service able to process and communicate on its own with a lightweight mechanism. Code changes to a service in microservices will not affect other services and this is not like a monolithic architecture that must rebuild and re-deploy the entire related system so that microservices become more flexible in accommodating changes [13].

In serverless computing, organizations or companies do not need to maintain resources either physically or virtually to run a code. Computer resources are provided by the service provider and only need to pay against the resources that have been used and there is no additional cost for unused resources. Function as a Service (FaaS) is a concept of serverless computing in serverless architecture where application developers can develop, create and run applications without worrying about complexity in the infrastructure used to run application code. FaaS will start within milisecond to run the process individually until it is complete. One of the challenges in FaaS is that users have no control over the resources used so users can only rely on service providers [14].

The focus of research on indoor positioning systems today is heavily on the use of techniques to improve the accuracy of indoor location determination. There has been no research focusing on the challenges that will be faced as a result of the use of complex techniques to improve accuracy. Application of complex techniques will certainly have an impact on computing processes that require greater mobile resources. Therefore, this research will focus on the right design architecture for implemention of indoor positioning system using BLE with moving computing process on smartphone to cloud servers to provide good scalability and availability.

This paper is organized as follows. After the Introduction section, Section 2 goes through some related works. Section 3 gives out the system architecture that is proposed to provide good scalability and availability on cloud. Section 4 shows the evalution process for analyses of the architecture. Section 5 shows the evalution results and discussion. Section 6 will conclude the study.

2. **Related Works.** In this section, some related work that has been done by previous researchers related to the indoor positioning system and serverless microservices methods will be discussed.

2.1. **Indoor positioning system.** Alexander and Kusuma conducted a positioning study in a 6 m $\times$ 4 m room using a fingerprinting approach combined with machine learning which was tested on the following methods such as ANN (Artificial Neural Network), MLR (Multiple Linear Regression), RF (Random Forest Regression) and SVR (Support Vector Regression). From the test results, the SVR method is the best algorithm among other methods with an average error rate of 134.92 cm [15].

Ghozali and Kusuma researched by combining machine learning methods with RSS fingerprinting and tested it in a room measuring 21 m $\times$ 12 m. In this study, comparisons were made on the three methods as the CNN model, MLP (Multi-Layer Perceptron) model, and the weighted sum model. The test results show that the CNN method gives the best results with an average error of 167.49 cm [16].

Lie and Kusuma conducted research using a new method called the coarse-to-fine algorithm by combining coarse estimation with the proposed fine-tuning. In the coarse estimation using weighted k-nearest neighbor to predict the user's location, then the prediction results will be fine-tuned using the delta rule. The results of the tests carried out in a 6 m $\times$ 4 m room resulted in an average error of 87.40 cm and a 21 m $\times$ 12 m room an error of 153.85 cm [17]. This method produces a smaller error rate when compared to the 2 previous studies.

For the research that has been done by previous researchers above, it shows that the various methods proposed have succeeded in increasing the accuracy of determining the location in the room. In this paper, we will use the fingerprinting algorithm (Weighted-KNN) because it has a fairly good performance and is often used as a benchmark in several studies.

2.2. **Serverless & microservices.** Viggiato et al. surveyed 122 professionals to find out about the advantages and challenges of micro services. From the survey results, it can be seen that micro services have advantages such as independent deployment, easy to scale, maintainability, and are not dependent on one technology stack but have challenges such as complexity in distributing transactions, testing the entire system, and service failures [18].

Akherfi et al. analyzed computational offloading in mobile computing and used several approaches to achieve offloading. All of the approaches taken have the same goal, namely increasing the smartphone's ability to save energy, reducing response time, and minimizing the execution cost [19].

Kratzke conducted a review of cloud applications and the evolution of monolith application deployment via micro services to a serverless architecture. From the results of the review, there are two trends. First, cloud computing and the evolution of its application architecture can be seen as a stable process for optimizing the utilization of computing resources in the cloud. Second, the development of resource utilization over time from architectural evolution is about how the application is created and deployed [20].

Microservices which is one of the cloud computing technologies where the computing process at the end user is transferred to the cloud. With the existence of microservices, the computing process that is usually done on mobile devices will be moved to the cloud that has greater resources (Offloading). With offloading techniques the use of resources and delay response on mobile devices can be reduced. Serverless architectures allow the provision of services to be more dynamic and application developers can focus more on application development than setting up cloud infrastructure as it is taken over by cloud providers. In addition, serverless has advantages in terms of scalability, namely the ability of the server to set the scale of resources according to the number of users. Resources will be scaled to zero if not in use and will be added according to the concurrent user. In this paper, it will be tested on monilith architecture and microservices that will be deployed on serverless (AWS Lambda).

3. **Proposed Architecture.** Figure 1 shows a system design with a serverless monolith architecture to provide services for mobile clients who use the BLE indoor positioning system. Each service will be received by one API Gateway for all tenants which will then be processed by one Lambda server with one database. All various services are provided by web services in a single large Lambda code. The smartphone will receive data signal input from Bluetooth low energy and be sent to the cloud server for further processing into a form of coordinates that can be recognized and displayed into a map on the application. The standard communication API Gateway is simple web services.

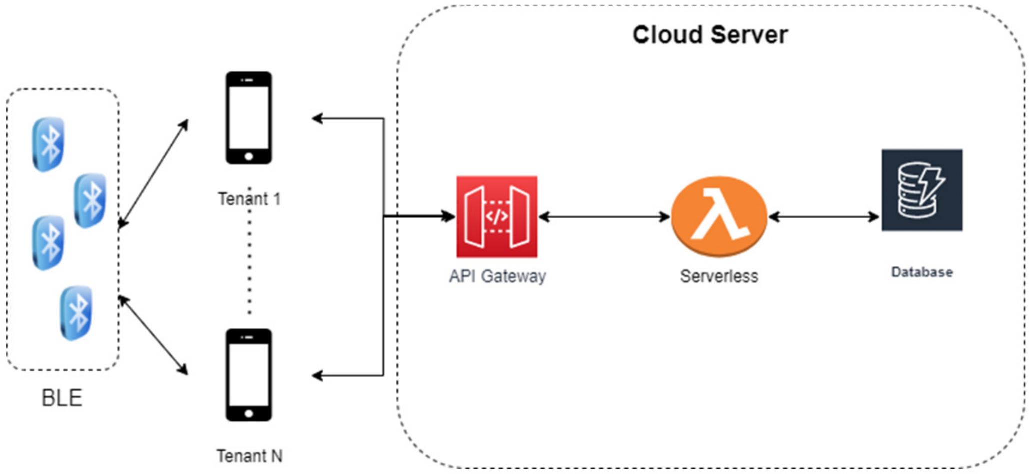


FIGURE 1. Serverless monolith architecture design

Figure 2 shows the design of the serverless microservices architecture that will provide services for indoor positioning system using BLE that all request from smartphone will go through the API Gateway on each tenant. Each API Gateway, Lambda function and database are designed separately from other tenants. Services for each tenant will also be separated into smaller services according to their respective functions. The separation of services to be smaller will be separated by Lambda function that can be directly inputted the programming code. The API communication standard used is the same as the design in Figure 1, namely simple web services.

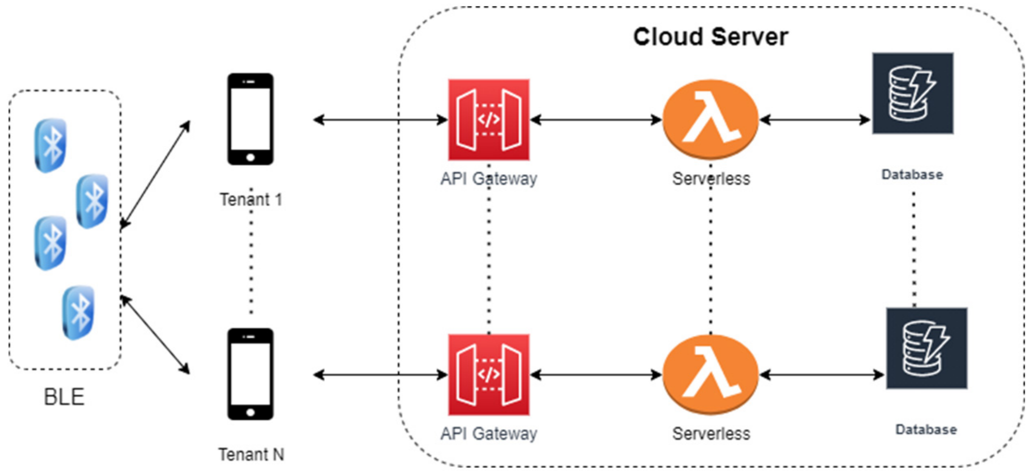


FIGURE 2. Serverless microservices architecture design

Flow diagram of the BLE application is described by Figure 3. The smartphone will be connected to 12 BLE that have been installed in the room where then RSSI and Beacon Id data will be collected by the smartphone and sent to the cloud server to be processed by calculation of fingerprinting algorithm (Weighted-KNN). Global coordinates will be returned to the user after the data is processed. Then, the map coordinates will be shown.

Cloud server will receive requests from users containing 12 RSSI BLE information that is read by the smartphone. All the information will then be processed by the cloud server to calculate the distance from the user's location to each reference point. Reference Point (RP) and Training Point (TP) are used to calculate the distance between RSSI using

Formula (1).

$$RSSI\ Distance_{(TP,RP)} = \sqrt{\sum_{i=4}^{12} \left(RSSI_i^{TP} - RSSI_i^{RP}\right)^2} \tag{1}$$

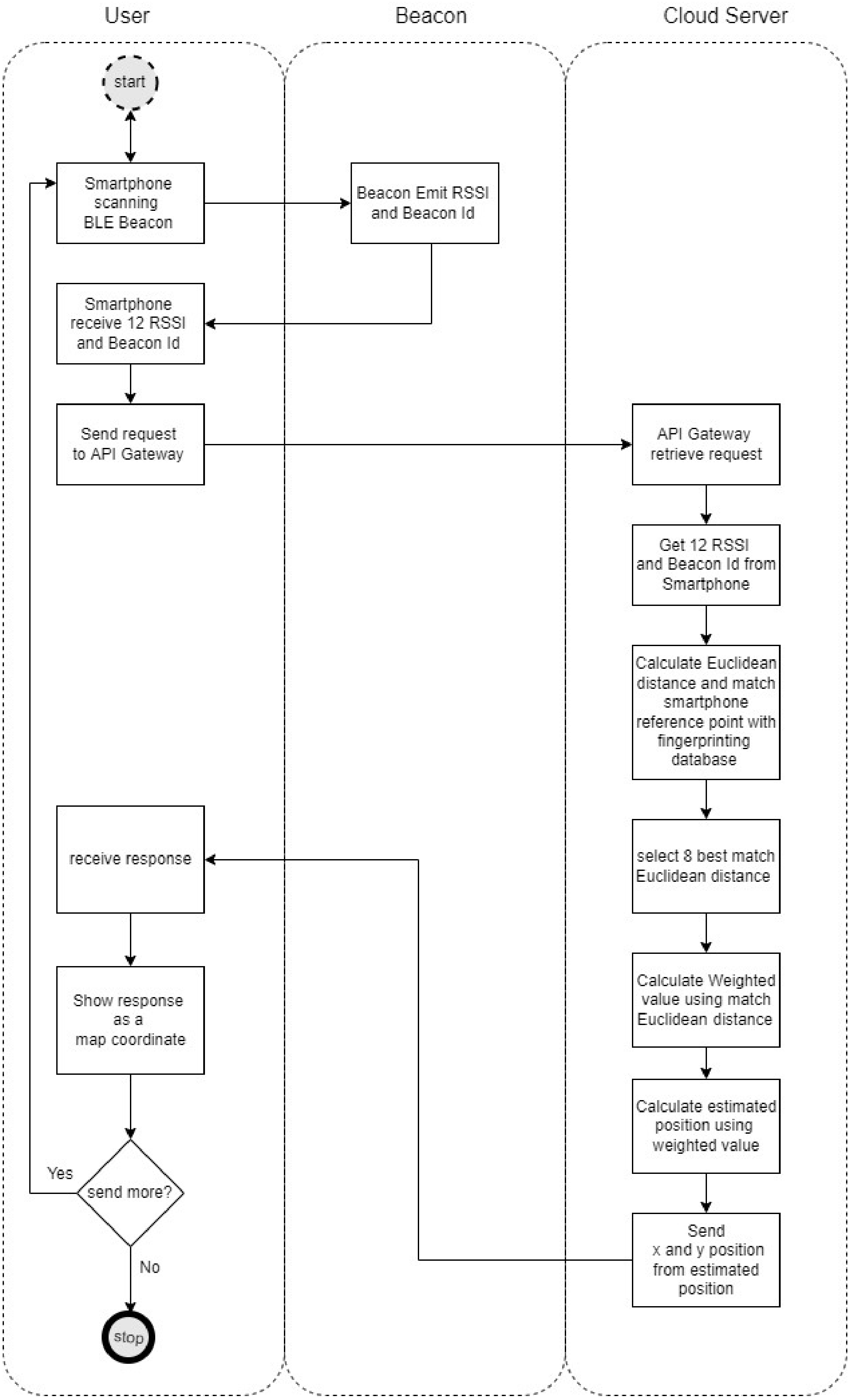


Figure 3. BLE indoor positioning system flow diagram

After getting the RSSI distance value will be calculated weighted sum ($W_i$) by looking for 8 reference points closest to the user. Weighted sum is used to obtain the coordinates $x$ and $y$ prediction results.

$$W_{ij} = \frac{1}{RSSI\ Distance_{(TP,RP)}} \tag{2}$$

$$W_i = \sum_{i=1}^{8} W_{ij} \tag{3}$$

$$(X, Y) = \sum_{i=1}^{8} (W_i \times X_i), \sum_{i=1}^{8} (W_i \times Y_i) \tag{4}$$

4. **Evaluation Process.** Figure 4 describes the stages of the evaluation process that implement the design of serverless monolith architecture and serverless microservices. Both architectures are implemented into the AWS cloud server (Amazon Web Services). The architectural design that has been implemented will be configured in accordance with the testing requirements in Table 1. The load test uses JMeter application with the number of concurrent users tested in accordance to test script. The test will be conducted on both architectures from 2 to 5 tenants and will be averaged for analyzing and concluding the test result.

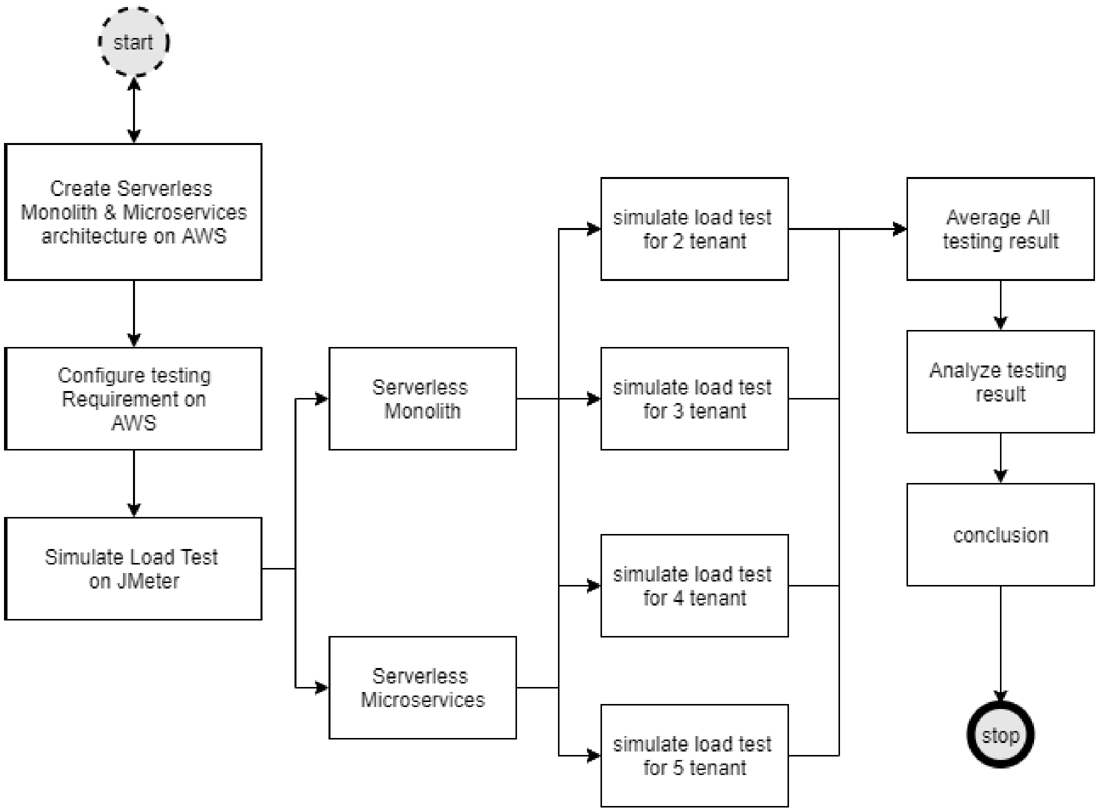


FIGURE 4. Evaluation process flow

Testing requirements in Table 1 are tested on both architectures with equally large memory comparisons, 3-second service time out configuration and database configuration using auto provision up to 40,000 with 5 initial value reserved for write and read.

5. **Evaluation Results.** Multitenancy testing is carried out starting from 2 tenants to 5 tenants. This multitenancy test will be averaged with 10 trials for each tested tenant scenario starting from 2 to 5 tenants. Then the test results from 2 tenants to 5 tenants

Table 1. Serverless architecture testing requirement

| Tenant | Architecture | Memory | Time out services | Database |
|---|---|---|---|---|
| 2 | Monolith | 256 MB | 3 s | Auto Provision (Reserved: Write/read (5), provision up to 40,000) |
| | Microservices | 2 × 128 MB | | |
| 3 | Monolith | 384 MB | | |
| | Microservices | 3 × 128 MB | | |
| 4 | Monolith | 512 MB | | |
| | Microservices | 4 × 128 MB | | |
| 5 | Monolith | 640 MB | | |
| | Microservices | 5 × 128 MB | | |

Table 2. Evaluation results with average error rate, response time and availability

| Load (Request/s) | Average error (%) | | Average response time (s) | | Availability (%) | |
|---|---|---|---|---|---|---|
| | Serverless monolith | Serverless microservices | Serverless monolith | Serverless microservices | Serverless monolith | Serverless microservices |
| 20 | 0.01% | 0.00% | 0.33 | 0.62 | 99.99% | 100.00% |
| 80 | 0.02% | 0.01% | 0.29 | 0.54 | 99.98% | 99.99% |
| 160 | 0.16% | 0.14% | 0.59 | 0.85 | 99.84% | 99.86% |
| 240 | 1.83% | 0.63% | 1.47 | 1.22 | 98.17% | 99.37% |
| 300 | 7.33% | 1.49% | 3.66 | 3 | 92.67% | 98.51% |
| 360 | 10.32% | 2.17% | 4.82 | 4.63 | 89.68% | 97.83% |

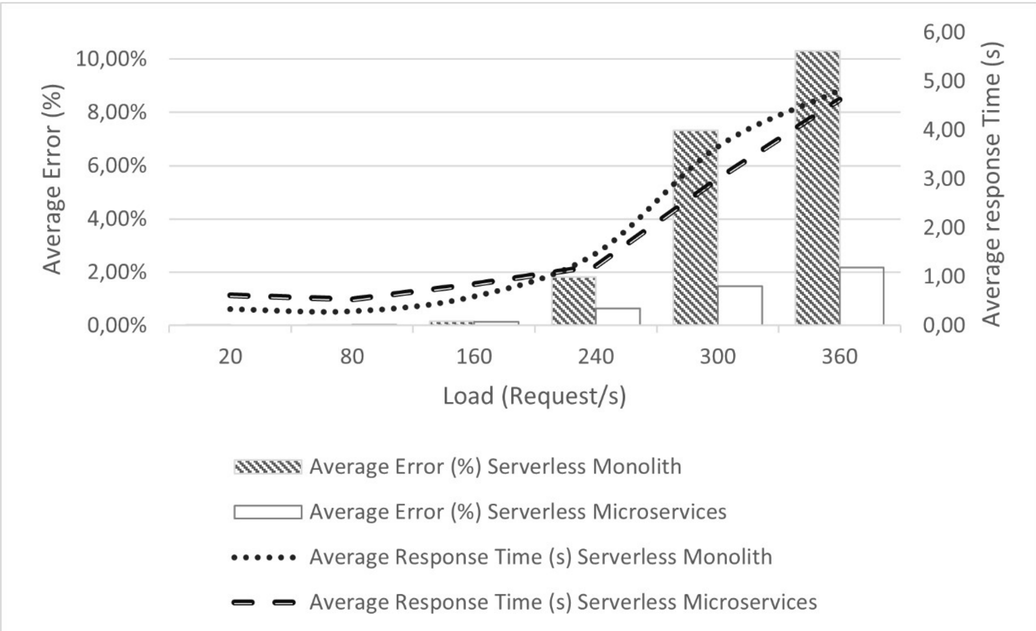


Figure 5. Average testing result with 2 to 5 tenants

will calculate the average error rate and response time. The results of the multi tension test can be seen in Table 2.

Figure 5 describes the relationship between the average error and the average response time for each architectural design. The smaller the average error value obtained means the better availability of the architectural design being tested. Response time indicates how quickly a service responds to requests made. The smaller the response time value from the test results, the faster the architectural design response. In the multitenancy concurrent user scenario of 360 per second, serverless microservices have an availability of

97.83% with an average response time of 4.63 seconds. This is better than the serverless monolith which only has an availability of 89.68% with a response time of 4.82 seconds.

6. **Conclusion.** In this study, the authors proposed a design of serverless microservices architecture that can be applied to indoor positioning systems using Bluetooth low energy. The proposed architectural design has succeeded in providing good availability and scalability in dealing with increasing concurrent users and supporting multi-tenants. Serverless microservices managed to accommodate 360 multi-tenant concurrent users per second with an availability of 97.83% and an average response time of 4.63 seconds. While serverless monolith only has an availability of 89.68% with an average response time of 4.82 seconds. In addition, the serverless microservices architecture has scalability capabilities by auto-provisioning on AWS Lambda and DynamoDB. For future research it can be continued from the client side by looking at the impact of resources used on smartphones.

## REFERENCES

[1] M. M. Din, N. Jamil, J. Maniam and M. A. Mohamed, Indoor positioning: Technology comparison analysis, *International Journal of Engineering and Technology*, vol.7, no.14, pp.133-137, DOI: 10.14419/ijet.v7i2.14.12813, 2018.

[2] C. Medina, J. C. Segura and A. L. Torre, Ultrasound indoor positioning system based on a low-power wireless sensor network providing sub-centimeter accuracy, *Sensors*, vol.13, no.3, pp.3501-3526, 2013.

[3] C. Lee, Indoor positioning system based on incident angles of infrared emitters, *The 30th Annual Conference of IEEE Industrial Electronics Society (IECON2004)*, Busan, Korea, 2004.

[4] Y. N. Naggar, A. H. Kassem and M. S. Bayoumi, A low cost indoor positioning system using computer vision, *International Journal of Image, Graphics & Signal Processing*, vol.11, no.4, pp.8-25, 2019.

[5] H. Huang, D. H. Lee, K. Chang, W. Li and A. T. Dev, Development of mobile platform for indoor positioning reference map using geomagnetic field data, *Computers and Electrical Engineering*, vol.68, pp.557-569, 2018.

[6] Z. Zuo, L. Liu, L. Zhang and Y. Fang, Indoor positioning based on Bluetooth low-energy beacons adopting graph optimization, *Sensors*, vol.18, no.11, pp.1-20, DOI: 10.3390/s18113736, 2018.

[7] S. Tomažiè, Indoor positioning and navigation, *Sensors*, vol.21, no.14, DOI: 10.3390/s21144793, 2021.

[8] V. C. Paterna, A. C. Augé, J. P. Aspas and M. A. P. Bullones, A Bluetooth low energy indoor positioning system with channel diversity, weighted trilateration and Kalman filtering, *Sensors*, vol.17, no.12, DOI: 10.3390/s17122927, 2017.

[9] L. Mainetti, L. Patrono and I. Sergi, A survey on indoor positioning systems, *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM2014)*, pp.111-120, DOI: 10.1109/SOFTCOM.2014.7039067, 2014.

[10] G. P. Kusuma and M. M. K. Lie, A review of indoor positioning system techniques using Bluetooth Low Energy, *ICIC Express Letters*, vol.13, no.12, pp.1139-1147, DOI: 10.24507/icicel.13.12.1139, 2019.

[11] T. Lorido-Botran, J. Miguel-Alonso and J. A. Lozano, A review of auto-scaling techniques for elastic applications in cloud environments, *Journal of Grid Computing*, DOI: 10.1007/s10723-014-9314-7, 2014.

[12] X. Wang, H. Ni, R. Han and X. Huang, Trade-off between service delay and power consumption in edge-cloud computing, *International Journal of Innovative Computing, Information and Control*, vol.14, no.6, pp.2011-2024, DOI: 10.24507/ijicic.14.06.2011, 2018.

[13] C. M. Aderaldo, N. C. Mendonça, C. Pahl and P. Jamshidi, Benchmark requirements for microservices architecture research, *Proc. of 2017 IEEE/ACM 1st International Workshop on Establishing the Community-Wide Infrastructure for Architecture-Based Software Engineering (ECASE2017)*, pp.8-13, DOI: 10.1109/ECASE.2017.4, 2017.

[14] B. Jambunathan and K. Yoganathan, Architecture decision on using microservices or serverless functions with containers, *Proc. of the 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT2018)*, pp.1-7, DOI: 10.1109/ICCTCT.2018.8551035, 2018.

[15] I. Alexander and G. P. Kusuma, Predicting indoor position using Bluetooth Low Energy and machine learning, *International Journal of Scientific and Technology Research*, vol.8, no.9, pp.1661-1667, 2019.

[16] R. P. Ghozali and G. P. Kusuma, Indoor positioning system using regression-based fingerprint method, *International Journal of Advanced Computer Science and Applications*, vol.10, no.8, pp.231-239, DOI: 10.14569/ijacsa.2019.0100829, 2019.
[17] M. M. K. Lie and G. P. Kusuma, A fingerprint-based coarse-to-fine algorithm for indoor positioning system using Bluetooth Low Energy, *Neural Computing and Applications*, vol.33, pp.2735-2751, DOI: 10.1007/s00521-020-05159-0, 2020.
[18] M. Viggiato, R. Terra, H. Rocha, M. T. Valente and E. Figueiredo, Microservices in practice: A survey study, *arXiv.org*, arXiv: 1808.04836, 2018.
[19] K. Akherfi, M. Gerndt and H. Harroud, Mobile cloud computing for computation offloading: Issues and challenges, *Applied Computing and Informatics*, vol.14, no.1, pp.1-16, DOI: 10.1016/j.aci.2016.11.002, 2018.
[20] N. Kratzke, A brief history of cloud application architectures, *Applied Sciences*, vol.8, no.8, pp.1-25, DOI: 10.3390/app8081368, 2018.