

Designing microservice architectures for scalability and reliability in e-commerce

I Asrowardi¹, S D Putra¹, E Subyantoro¹

¹ Department of Management of Informatics, Politeknik Negeri Lampung, Jalan Soekarno Hatta, Bandar Lampung, Indonesia

E-mail : imam@polinela.ac.id

Abstract. The rapid growth of ICT is one of the challenges and opportunities in the field of purchasing, selling, and marketing goods and services are better known in the world of ICT as e-commerce. The need for e-commerce requires various strategies to increase revenue and customer satisfaction. E-commerce companies are required to share various services in increasing customer satisfaction. The problem that occurs is that e-commerce is demanded to have reusability services, automated service deployments, and fast service scalability. The best approach now is to use microservice to support current e-commerce needs. This paper discusses the design of a microservice on e-commerce services with the design science research methodology (DSRM) approach. The evaluation process was testing the performance, accessibility, best practices, search engine optimization. The results from the proof of concept microservice architecture for e-commerce already presented in this paper. Performance testing result shown the microservice architecture shows a better 30 percent and 4 percent compare with monolithic architecture for scripting and painting.

1. Introduction

The information and communication technology (ICT) was developing very rapidly, and comprehensively gets the life of the community. The rapid growth of ICT is one of the challenges and opportunities in the field of purchasing, selling, and marketing goods and services are better known in the world of ICT as e-commerce. The progress of ICT is speedy, more than half of Indonesia's population is now connected to the internet, this data is taken from the survey results of the Indonesian Internet Network Providers Association (APJII) which surveyed that in 2017 as many as 143.26 million Indonesians have used the internet. Significant increases can be utilized in the e-commerce field. E-commerce consists of the use of electronic and ICT such as the internet, in promoting access to services to the community. ASEAN is one of the promising markets for e-commerce growth, and it is predicted that e-commerce growth in ASEAN from 2013 to 2017 will reach 25%, this result is still above the market growth in the United States which is only 11% and EU countries which only 10 %. The growth of e-commerce in recent years has become a big business prospect in commerce, according to research by A.T. Kearney with a population of more than 240 million people, Indonesia's e-commerce market in 2013 reached the US\$ 1.3 billion, Indonesia is a potential country in the field of the e-commerce business. The need for e-commerce requires various strategies to increase revenue and customer satisfaction. E-commerce companies are required to share various services in increasing customer satisfaction. The problem that occurs is that e-commerce is demanded to have reusability services,



automated service deployments, and fast service scalability. The best approach now is to use microservice to support current e-commerce needs.

Microservices architectural are new architectural styles that try to solve existing problems in monolithic service architectures. Microservices architecture allows faster application development, and each service can be used and updated separately [1]. Architectural microservices where the system is designed to provide specific and focused services. Although the term microservices has been introduced since 2011, and in 2012, some groups decided that microservices service was the most appropriate service for application development [2]. Microservices in application development is needed because microservices is a cloud that aims to realize small service systems, each of which can be used individually on potentially different platforms and runs in its processes and communicates through lightweight mechanisms such as using the REST API [3]. REST API is a type of architecture for implementing web services that apply the concept of state transfer. The state is described as requesting a web page, on the server side will send the state to the site page. HTTP commands are used in REST and have functioned as getting, post, put or delete, and in their application, REST API is widely used in the web service section. REST API is an excellent choice for developers, their language is independent and compatibility with many platforms [1].

On the other hand, monolithic architecture is a standard way to create architectural applications, over the years using a single code base that makes developers easier and applications relatively small, but the poor development of monolithic architecture slows application development and makes it more challenging to join new developers. In monolithic architectures, the resources of the same machine (in example memory, databases, and files are there are not independently executable [4].

This paper discusses the design of a microservice on e-commerce services. We try to compare microservice and monolithic architecture designs in terms of performance. After getting the results of this test, we hope the microservice approach has a proof of concept and critical challenges encountered to be implemented in e-commerce services. This paper will describe related research to related works, the method developed in the method section part, and conduct an evaluation and performance results in the form of stress testing at the end, namely result and performance evaluation.

2. Related works

2.1. Object-Oriented Programming

Concepts in Object Oriented Programming (OOP) programming see the efficiency and benefits of OOP for software development and maintenance, so OOP is indispensable for learning high-level programming languages based on OOP, such as C++ and Java. But in learning the concept of structured programming many graphical and visual programming tools have been developed to facilitate system development, such as scratch, blockly, and others. Most OOP developments adopt a text-based programming approach with OOP concepts, such as the relationship between classes, objects, methods, and concepts. OOP is a concept in programming that is so important for the development of modern software. The concept of OOP is based on the concept of objects. Thus, object-based concepts are useful for software design and development. OOP programming has six essential concepts to support OOP features, namely Class, Object, Method, Encapsulation, Inheritance, and Polymorphism. In OOP, a class defines abstract data formats (Attributes) and procedures (Method). An abstract class will be realized by an object, which is an instance of a class and can execute methods associated with related attributes during run-time on the computer. Details of the attributes and methods specified will be encapsulated to class. Therefore, the installed object can use and run this method directly and communicate with each other through the method passing approach. in addition, methods with various definitions can be executed dynamically according to the type of method call appropriate for the polymorphism feature. Classes (derivatives) can be defined from existing classes using the inheritance approach for efficient software development [5].

2.2. Benefit and challenges implemented microservice

Microservices for e-commerce was interested reasons. The e-commerce services that simplify and accelerate deployment through modularity because each microservice is responsible for a small portion of all application functionality. Both microservices services can take advantage of heterogeneity programming languages and frameworks, because they only require a standard cross-application API, usually through the Remote Procedure Calls (RPC) or REST API. The three microservices services simplify development, accuracy, and performance because bugs can be isolated for specific components, where problem-solving often involves end-to-end service, so microservices service fits into a centralized data-based model, with each microservice being accommodated in single container [6].

Architectural microservices are cloud-based software systems and differ fundamentally from monolithic system architectures. Architectural microservices have a communication mechanism to work in conjunction with other services. This service will be easily implemented using a programming language due to different frameworks, platforms, and databases. Architectural microservices are generally integrated with DevOps techniques for automated system development through continuous integration. Application monitoring also allows architecture microservices to get feedback to facilitate system improvements, thereby shortening the development cycle. Some companies, such as Netflix and Amazon, have adopted microservices architecture to build software systems that can be scaled and efficiently in the maintenance process [7]. The advantages of architecture microservices according to [8] are:

- Microservices make the on-boarding process easier.
- It provides flexible data storage.
- Microservice activates polyglot programming or programming language that is easy to understand.
- Can handle problems that occur fault tolerance or fault tolerance and fault isolation.
- Microservice provides scalability, the ability of the system to handle the growth in the amount of data and concurrency without having an impact on system performance.
- Make it easy to monitor security.

While the shortcomings of architecture microservices namely:

- Microservices are more complex.
- Microservices require cultural change.
- Microservices are more expensive.
- Service is alert to security challenges.

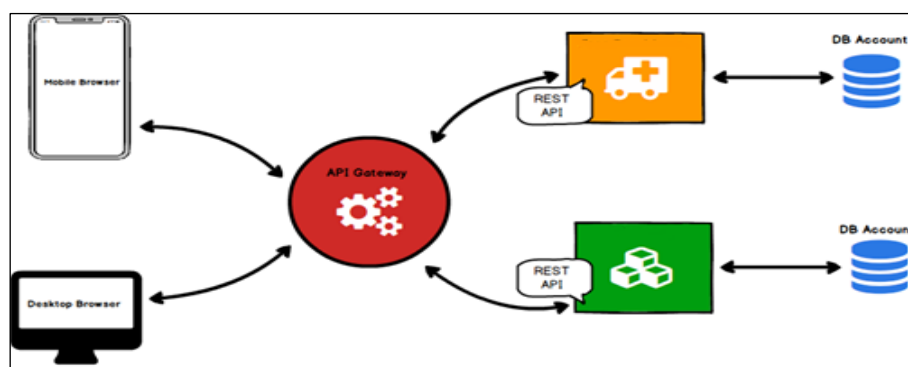


Figure 1. Model implementation using microservice.

Microservices architecture is an alternative service that is more scalable and more flexible in the microservices architecture; information systems are designed to provide services that are more specific and focused. Big problems are broken down into several small solutions and arranged in one service, where each service has its own tasks. By using this approach, the information system will have several

services that can be managed independently, and this will make it easier for the system to adapt to changing needs [9]. One most important thing before migrating the system is always considering maintaining risk adequately [10] and should to manage all the services [11]. The implementation model that we applied to the microservices-based web application e-commerce is shown in Figure 1.

3. Methodology

We propose a Design Science Research Methodology (DSRM) approach to this study. The research design framework uses research methods that combine research design paradigms namely behavioral science and design science. Behavioral science as developing and verifying theories that explain and predict business needs. On the other hand, design science will produce an output that is useful in solving a problem. Behavioral science and design science are combined and can be used in a system design research framework that can address the needs of business organizations were presented in Figure 2.

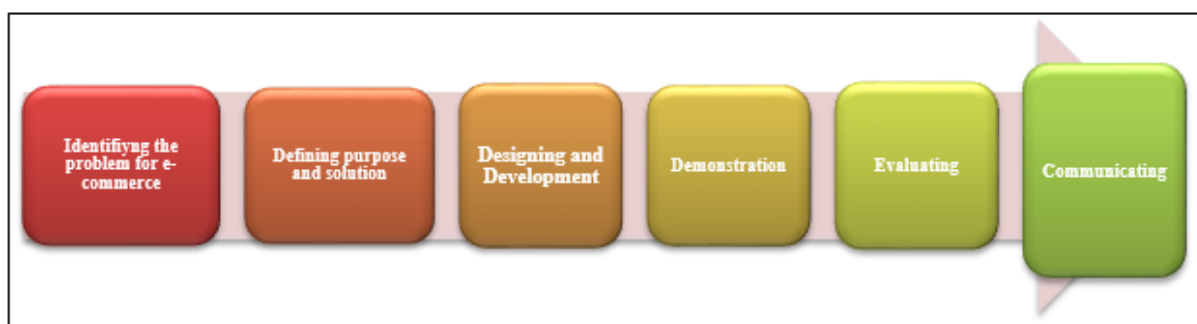


Figure 2. Research method.

4. Results and performance evaluation

4.1. Controller

The parameter designed in this study is the application of taking orders on an e-commerce system. Following is the controller class table that is in the taking order application and then will be explained in each controller class. The variable controller and method are shown in Table 1.

4.2. Performance testing monolithic and microservice architecture

The evaluation phase is to find out the mistakes in the application developed and to get test data such as determining performance, accessibility, best practices, search engine optimization. Performance testing is a non-functional testing technique used to determine system parameters in terms of response and stability in a load of a website or application. Accessibility testing is testing conducted to test access or ease of users in accessing and using web applications. Best practices are tests conducted to determine the parameters of the best web application performance. Search Engine Optimization (SEO) aims to obtain data optimization on a web application on the top position on a search engine when the web application that we build has been published. The performance testing result between microservices and monolithic architecture are shown in Table 2.

The comparison between microservice and monolithic performance was shown in Table 1. The microservice architecture shows more loading times and access to the system compared to the monolithic architecture, which is 62% and 67%. However, in the aspect of scripting and painting, microservice performance shows a better 30% and 4%. The time difference in the loading aspect is caused by web service performance and network connectivity.

Table 1. Variable controller and method.

System Needs	Method	URL
Creating customer data	POST	/home/createaccount
Showing item data	GET	/home/menu1
Make data items	POST	/home/menu1create
Changing item data	POST	/home/{id}/menu1update
Erasing item data	GET	/home/{id}/menu1delete
Showing category data	GET	/home/category
Add category data	POST	/home/categorycreate
Change category data	POST	/home/{id}/categoryupdate
Delete category data	GET	/home/{id}/categorydelete
Display data outlets	GET	/home/menu2
Add data outlets	POST	/home/menu2create
Change data outlets	POST	/home/{id}/menu2update
Removing data outlets	GET	/home/{id}/menu2delete
Showing location data	GET	/home/location
Add location data	POST	/home/locationcreate
Change location data	POST	/home/{id}/locationupdate
Clear location data	GET	/home/{id}/locationdelete
Add items to cart	POST	/home/addtocartandverif/{id}
Make an order	POST	/home/createorder/{id}
Make a transaction	POST	/home/checkout/{id}
To send proof of payment	POST	/home/{id}/payvalidation

Table 2. Performance testing result.

Performance	Microservices Times	Monolithic
Loading	283.6 ms	175.0 ms
Scripting	1180.4 ms	1705.0 ms
Rendering	479.0 ms	377.7 ms
Painting	45.2 ms	47.1 ms
System	790.0 ms	471.5 ms
Accessibility	66	66
Best Practice	79	79
SEO	91	91

4.3. Stress testing

To ensure that the microservice and the required APIs are able to serve customers, we do stress testing. It aims to get the API available, fast, safe and reliable. The performance testing result for some iteration is shown in Table 3.

Table 3. Performance testing result.

Variable	Iteration (300)	Iteration (500)	Iteration (1000)
Client average (all)	1270 ms	1080ms	1593 ms
Max duration	24115 ms	24956 ms	26501 ms
Client average (best)	825 ms	754 ms	1251 ms
Min duration	441 ms	434 ms	473 ms
Server average (all)	1268 ms	1078 ms	1591 ms
Server average (best)	823 ms	752 ms	1249 ms
Total duration	381006 ms	539878 ms	1592785 ms
Stress	295	495	995
Warmup	5	5	5
Error	1	13	30

The result was shown in Table 3 is a comparison test microservice architecture test between 300, 500, and 1000 iterations. The iteration test shows that the microservice architecture in e-commerce is able to serve peak data requests of 22.000 data request requests. This is tested on stress testing with a total data testing of 300 as shown in Figure 2, 500 iterations as shown in Figure 3, and 1000 iteration times as shown in Figure 4.

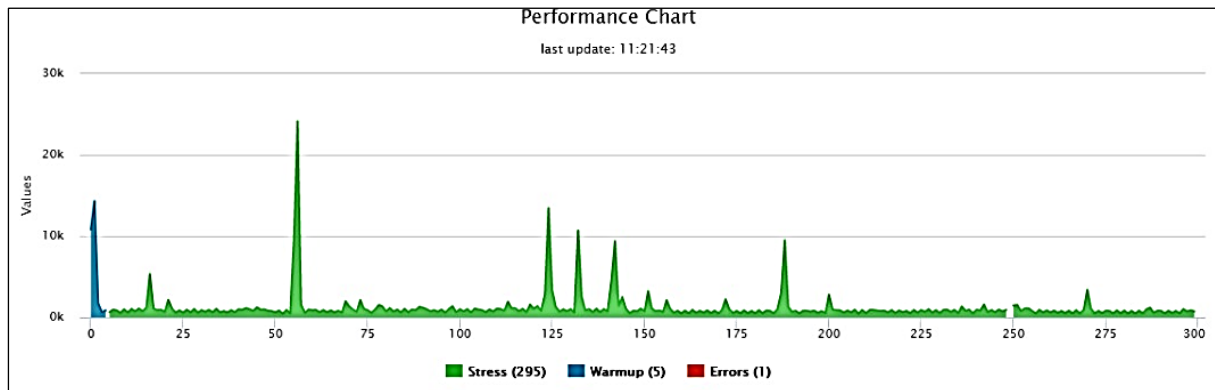


Figure 3. Testing with 300 data test iterations.

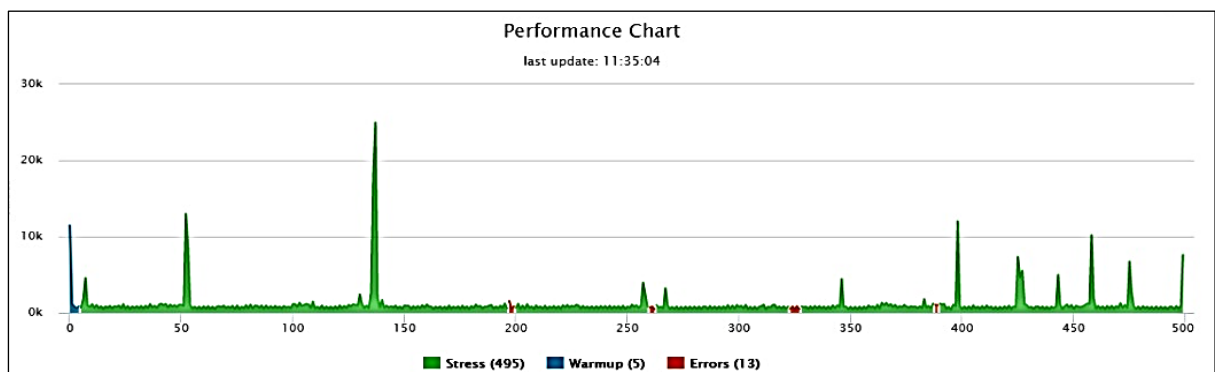


Figure 4. Testing with 500 data test iterations.

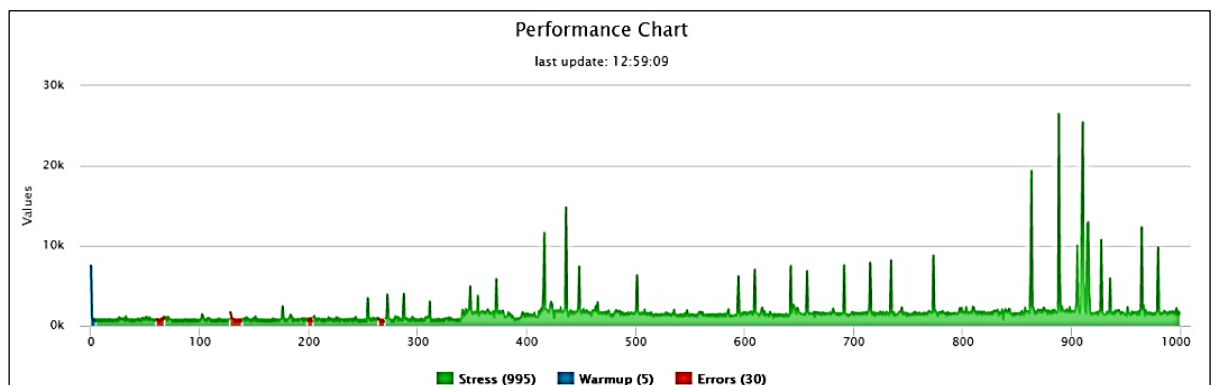


Figure 5. Testing with 1000 data test iterations.

5. Conclusions

In this section, the results from the proof of concept microservice architecture for e-commerce already presented. The performance testing of the attachment microservice transition from monolithic are discussed together. The biggest challenge for implementing microservice architecture is migration with deployment artifacts and details. The research finding microservice architecture shows more loading times and access to the system compared to the monolithic architecture, which is 62% and 67%. The time difference in the loading aspect is caused by web service performance and network connectivity. Furthermore, due to the existence of the number of developer teams working on monolithic systems, any changes need much coordination. These issues related migration process can be meet at development life cycle. Before migrating from monolithic to microservices architecture can be made to create a modular monolith. Migrating from the legacy to components inside the monolith has significant challenges and considering the security problem.

6. References

- [1] Miika K 2017 *Transforming Monolithic Architecture Towards Microservice Architecture* (Finland: Helsingin Yliopisto)
- [2] Sourabh S 2016 *Mastering Microservices with Java* (United Kingdom: Packt Publishing Ltd)
- [3] Balalaie A 2016 Microservices *IEEE Software* **33** 42–52
- [4] Bucchiarone A, Dragoni N, Dustdar S, Larsen S T and Mazzara M 2018 *IEEE Software* **35** 50-55
- [5] Su J M and Hsu F Y 2017 *IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)* **6**
- [6] Gan Y and Delimitrou C 2018 *IEEE Computer Architecture Letters* **17** 155-158
- [7] Fan C Y and Ma S P 2017 *IEEE International Conference on AI & Mobile Services (AIMS)* 109–12
- [8] Phil W 2019 *Microservices Disadvantages & Advantages – Tiempo Development* (USA: Tiempo Development)
- [9] Li S, Zhang H, Jia Z, Li Z, Zhang C, Li J, Gao Q, Ge J and Shan Z 2019 *Journal of Systems and Software* **157**
- [10] Putra S D, Sutikno S, Rosmansyah Y and Asrowardi I 2014 *Conference on ICT For Smart Society (ICISS)*
- [11] Asrowardi I, Putra S D, Subyantoro E and Daud N H M 2018 *IT IJECE* **8** 4023