ISSN 2347 - 3983
Volume 8. No. 7, July 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter86872020.pdf
https://doi.org/10.30534/ijeter/2020/86872020

Implementation of Service Oriented Architecture Using Web API & SOMA in E-commerce Web Application

Samuel[1], Abba Suganda Girsang[2]
[1]Computer Science Department, BINUS Graduate Program - Master of Computer Science,
Bina Nusantara University, Jakarta, Indonesia 11480
samuel009@binus.ac.id
[2]Computer Science Department, BINUS Graduate Program - Master of Computer Science,
Bina Nusantara University, Jakarta, Indonesia 11480
agirsang@binus.edu

ABSTRACT

The purpose of writing this paper is to produce an online buying & reseller send point system design with Ditoko.com (one of the e-commerce in Indonesia) business-to-customer (B2C) application based on Service Oriented Architecture (SOA) using the API Gateway Integration. API Gateway can help to solve complexity problems with n-to-n integration, where n is the number of integrated applications. Implement Service Oriented Modeling and Architecture (SOMA) methodology as a guide in implementing SOA-based solutions. SOMA has seven parts, namely Business Modeling and Transformation, Solution Management, Identification, Specification, Realization, Implementation and Deployment, Monitoring and Management. API Latency Test Tools as a tool to measure the performance of each service connected to the B2C Ditoko.com web application. Data is analyzed by comparing service usage with the concept of API Gateway and n-to-n service integration where n is the number of integrated applications. The results obtained indicate that using the API-based SOA architecture has a lower time to access each services/pages than using a service with n-to-n integration (monolithic architecture). So that it can improve service quality performance.

Key words : SOA, Monolithic, API, SOMA, Microservice, Software Architecture

1. INTRODUCTION

The rapid growth of the e-commerce market in Indonesia at this time cannot be doubted. With the number of internet users in 2018 reaching 112 million people or more than 40% of the population in Indonesia. Data from the MOCIT shows that the value of e-commerce transactions in 2017 reached IDR 325 trillion and will increase by 30% - 40% in 2018 [1]. Because of very rapid growth and intense competition, every business person must have a business strategy that is superior to being able to survive in the competition. One of the business people & team who participated in e-commerce competition is Ditoko.com. Founded in August 2018, Ditoko.com is a platform that offers services and helps consumers / buyers who like to shop for goods via online. Promotion & Marketing is done through social media like Instagram & Facebook.

As a platform that helps in purchasing / shopping for goods via online, Ditoko.com is expected to be able to provide convenience for online shopping players to do online shopping easily & quickly, of course, also supported by the addition of useful features on the Ditoko.com application to support the effectiveness of shopping for customers / buyers. However, there are a number of conditions faced by Ditoko.com today including:

1. The location of shipping goods is only concentrated in one place / area (Tangerang), while many Ditoko.com customers / buyers make purchases from outside Jabodetabek even outside of Java, where this raises problems from the long duration of delivery and shipping costs. which is quite high and will certainly be burdensome for the customer / buyer. From this issue, need to be created a reseller send point system to pick the suitable reseller to fulfill the buyer order.
2. The load time performance: With the number of users that has reached 120,238 people, Ditoko.com will build other additional features, where with the current conditions it is very difficult to scale up the Ditoko.com application, and with the addition of new features, it will certainly be burdensome for the Developers at Ditoko.com if they have to load the whole project code every time they need to make changes or additions to the code. Of course this also causes developers to not be able to focus on one particular module / feature that they will develop.

From this issue, need to be developed a new Service Architecture to improve the load time performance of each functions & services inside Ditoko.com Web Application.

Service Oriented Architecture (SOA) & Microservices become the solutions to overcome this. SOA exists as an architecture that can reduce the level of dependency between components of information technology. The SOA & Microservices concept is increasingly being found as an architectural concept for creating modern enterprise applications. One important element of SOA & Microservices is integration. The business processes that occur at Ditoko.com are dynamic and evolving causing several services or features that support business processes. Between functions/services must interact and depend on each other, good integration will make a business that involves more than one system can run smoothly and will certainly improve the efficiency of the Ditoko.com application. And this is done by applying the design and architectural concepts from SOA & Microservices.

2. THEORETICAL BASIS

2.1 Web Service

Web Service is a service available on the Internet [2]. Web Service uses the standard XML / JSON format for sending messages. Web Services are also not bound to specific programming languages or operating systems [3]. Web Services are interfaces that describe collections that can be accessed on the network using the standard XML / JSON format for message exchange. Web Services do specific tasks.

Web Services are described using a standard XML / JSON notation format called services description [4]. Web service is used as a facility that provides services (in the form of information or data) to other systems, so that they can interact with the system through the services provided.

Web services store information data in JSON / XML format, so that this data can be accessed by other systems even though different platforms, operating systems, and programming languages.

2.2 REST

Representational State Transfer, abbreviated as REST, is one type of architecture for implementing web services that applies the concept of inter state transfer [5]. State here can be described as a browser request / request to a web page, on the server side will send the current site page state to the browser [6]. Navigating through the URL provided is the same as changing the state of the page.

Just like how REST works, by navigating through HTTP links to do certain activities. It is as if there is a state transfer between one another. HTTP commands that can be used in REST are GET, POST, PUT or DELETE functions. In its application, REST is more widely used in resource-oriented web services. The term for web services that implement the REST architecture is RESTful web service.

2.3 WEB API and JSON

Web API is the program interface of the system that can be accessed via methods and headers on standard HTTP protocol. Web APIs can be accessed from various HTTP clients such as browsers and mobile devices. Web API also has the advantage of using infrastructure that is also used by the web, especially for the use of caching and concurrency [7]. JSON (Java Script Object Notation) is a data exchange format that is lightweight, easy to read and write by humans, and easily translated and generated by computers.

This format is based on part of the javascript programming language. JSON consists of two structures, namely:
- Pairs of names with values. In some languages this is stated as object, record, struct, dictionary, hashtable, keyedlist or associative array.
- List of sorted values (unordered list of values). In most languages, this is stated as array, vector, list or sequence.

2.4 E-commerce

E-commerce or can be called Electronic commerce or e-commerce is the distribution, purchase, sale, marketing of goods and services through electronic systems such as the internet or television, www, or other computer networks. E-commerce can involve electronic funds transfer, electronic data exchanges, automated inventory management systems, and automated data collection systems.

On 2012 in Indonesia, there were around 38 million consumers every month who made purchases or sales online. The number of these transactions increased to 100% from the previous year where only 19 million consumers made monthly transactions [8].

E-commerce can be divided into 3 categories, shown in Figure 1:

1) Business to Business (example: Indotrading)
2) Business to Consumer (example: Bhinneka)
3) Consumer to consumer (example: Tokopedia)

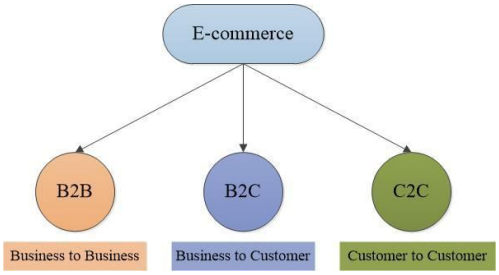

Figure 1: E-commerce Category

2.5 Microservice

Microservice is the latest trend that is growing rapidly in the enterprise world. Although the term microservices has been introduced since 2011, the design pattern of this software / information system architecture is becoming increasingly widely studied and used by the industry after the advent of container-based virtualization technology, namely Docker at the end of 2014 [9]. Many companies, such as Amazon, Netflix, and Guardian, use Microservice Architecture to develop their large and complex application delivery that is continuous development & delivery, which also provides flexibility and diversity in terms of technology [10].

Microservice focuses on a broader area of Service-Oriented Architecture (SOA) and focuses on certain aspects, such as small lightweight service componentization, Agile and DevOps applications for development, use of infrastructure automation with features that are delivered continuously, management decentralized data and decentralized protocols between services [11].

There are many differences between SOA and Microservice Architecture. For example, the service design in Microservice Architecture is driven by a philosophy of sharing nothing to support agile methods and promote isolation and autonomy. Instead, SOA adopts the philosophy of sharing as much & as much as possible for high-level reuse [12].

This microservice architectural style forms a system as a small set of loosely-coupled small services that are isolated in small units that are coherent and autonomous [13]. There are many solutions proposed by researchers to improve the quality of software / information systems in terms of non-functional requirements (scalability, reliability, maintainability and availability). Starting from the use of mirrorring / DRC / cloud / techniques to the algorithm approach to be able to restore system performance after an attack [14].

Microservice, which recently gained popularity among Software Engineering practitioners, offers a slightly different approach. Enterprise information systems that are generally built with a monolithic approach (applications wrapped in a large package, where changes to one part of the program code will have a major effect on other program code) are shifted to a distributed approach. Applications are divided into small parts that function specifically (high cohesion) and do not depend on other program components (loose coupling), with an API interface (Application Programming Interface) [15].

The concept of microservices is more or less different from the paradigm of its predecessor, namely System Oriented Architecture (SOA). Starting from the use of message communication protocols that are more concise (REST, etc.) compared to SOA which uses XML SOAP a lot, to the design and functional division processes that prioritize the division based on domain functionality in organizations [16].

When the process of developing the program code is complete, and the software is ready to be deployed on the server / cloud, the image docker can be easily deployed on the server, so that the server configuration setting problem becomes no longer a problem [17]. Several studies evaluating Docker performance compared to similar systems (OpenStack, KVM virtualization, etc.) conclude that Docker provides superior performance, with a small size image and lightweight does not burden server performance [18].

2.6 ESB & API Gateway

The Enterprise Service Bus or ESB is a flexible architecture that can be operated for integration between applications and services, this can reduce the number and complexity of interfaces in SOA architecture, improve business processes, integrate heterogeneous applications and also increase the value of data assets. ESB is a combination of traditional middleware technology, including XML and web services, it provides a standard-based message exchange mechanism and other components through adapters and standard interfaces to meet application integration needs for large companies [19].

The service bus approach to integration uses technology that provides buses so that applications can be integrated with one another. Various applications do not communicate with each other directly but rather communicate through SOA backbone middleware. The most distinguishing feature of ESB architecture is the distributed nature of the integration topology [20].

3. PROPOSED METHODOLOGY (USING SOMA METHOD)

3.1 Business Modeling and Transformation

At the stage of Business Modeling and Transformation, identification of initial business processes that are currently running within an organization. Business processes that have been running are modeled, simulated and optimized, then identified using a series of SOMA stages, so that they will form a new business process that is the result of Business Process Re-engineering.

This stage is carried out using several approaches, starting from value chain diagrams that will describe business processes and Business Process Diagrams that are used to describe the details of running business processes and business processes that are changed through Business Process Re-engineering, and at this stage will be elaborated more details related to the use case of each actor in an application system.

3.2 Solution Management

At this stage, the Initiate Project Management Activities process, Select Solution Templates and Patterns, and the Conduct Method Adoption Workshop are carried out. This stage is carried out the consolidation of the solution by defining what solution is used, the applications involved in it, and the part that participates in the solution completion.

3.3 Identification

At this stage it starts from looking at observations from Fulfillment by Amazon and gathering every feature / function available on the Ditoko.com web application. From the results of observations and collection of web services then enter the Goal Service Modeling stage to sharpen the focus of making microservice. After getting the Goal Service Modeling, then go to the next step, Process Decomposition to identify the process in the system to be made. Decomposition is carried out on business processes that have been previously defined in Business Process and Transformation to produce microservices components to be used.

3.4 Specification

At this stage the specifications include high-level design and detailed specification of the service components to be used. By using the identification results that have been carried out in the previous stage, the design of the composition of services and also the process of exchanging data between services is carried out at this stage.

3.5 Realization

The realization phase functions to connect various predetermined services with related components to be implemented in implementation. In the realization process, the Microservice Reference Architecture is used as a guide to determine and describe the layers needed to form a Microservice system.

3.6 Implementation

During the implementation phase of the Ditoko.com application from SOMA, service, functional and technical components that realize services, components, and flows are built, generated, and integrated. In some cases, the existing assets are refactoring and repaired so that they can be used to realize a service. In the implementation phase, build, produce, and assemble service, functional, and technical components that create services, components, and flow.

Use a variety of template solutions that include support for custom development, integration of packaging applications, design, and composite business services. The implementation phase is expanded by the activities and tasks of this solution template for various types of solutions in the overall solution. Depending on the type of solution, additional activities and tasks from the appropriate solution template are executed to create and update the design as needed.

3.7 Deployment, Monitoring and Management

As in the implementation phase, this phase is also enhanced by the activities and tasks of the solution templates for various types of solutions. In addition, modeling using SOMA provides support for monitoring and managing business processes and monitoring performance in the production environment as part of this phase. SOA-based solutions will benefit from using descriptive guidelines provided as part of SOMA.

The final stage of the SOMA methodology is the deployment of all services to the production level after a user acceptance test (UAT) is performed where the user performs the overall testing and gives a sign that the system process is running smoothly. After deployment services, it is necessary to monitor the process and performance to determine the stability and availability of the Ditoko.com application system.

4.IMPLEMENTATION AND ANALYSIS PERFORMANCE

4.1 Business Modeling & Transformation

Through the Value Chain Diagram in Figure 2 represented a series of activities carried out by Ditoko.com to produce products and services in the form of the Ditoko.com website application & include features that support the provision of services / services to purchase products, as well as e-services other commerce. Where all the series of processes that will lead to margins that will generate revenue & profits for the company.

Figure 2: Value Chain Diagram for Web Application Ditoko.com

The series of processes / activities that take place at the Ditoko.com company is divided into 2 major parts, namely the main activities (primary activities) and support activities, where the main activities consist of:

1. Inbound Logistics
Inbound Logistics is an activity to prepare storage of raw material products, in this case renting warehouses and providing a system for carrying out inventory control & stockists to ensure that stock of product materials or materials is maintained and can meet the manufacture of products that are ordered by a customer Ditoko.com. In this process scheduling / scheduling for delivery of incoming goods from producers to the Ditoko.com warehouse is also carried out.

2. Operations
Like business in general, Ditoko.com requires operational processes to support day-to-day activities in supporting business activities. Operational activities carried out include the process of printing the address of Ditoko.com's customers, the process of finishing the product, and followed by the product packaging process.

3. Outbound Logistics
In Outbound Logistics there is an activity to prepare the storage of the final product that is ready for consumption by the customer, in this case renting the warehouse and providing a system for carrying out inventory control & stockist to ensure that the stock of the final product is maintained and can fulfill customer orders Ditoko.com. In this process scheduling / scheduling for delivery of goods from the Ditoko.com warehouse is carried out directly to Ditoko.com customers.

4. Marketing & Sales
To support the sustainability of Ditoko.com e-commerce business and as the largest source of cash-in at Ditoko.com, marketing & sales activities are required. In marketing and sales activities will focus on funneling.

5. Service
In service activities, Ditoko.com's customer service team will focus on answering questions & doubts, as well as feedback from Ditoko.com prospective customers and also be ready to serve complaints & complaints submitted by the customer, thus building customer confidence to always be loyal to buy through the application Ditoko.com

4.2 Solution Management
At the stage of initate project management activities, it is determined which parts will take part in the completion of the project in accordance with the allotted time. At this stage also carried out data collection on business processes that are running, as well as technical documentation about the running system.

At the stage of determining the solution template begins by looking at business processes that are running and after finding the problem encountered is the lack of space for the placement of each service and with the addition of the latest service, the send point reseller product that will be implemented on the Ditoko.com web application. The process of making a container for each of these services needs to be developed and integrated in an existing system to increase the availability of the system depicted in the figure. 3 below

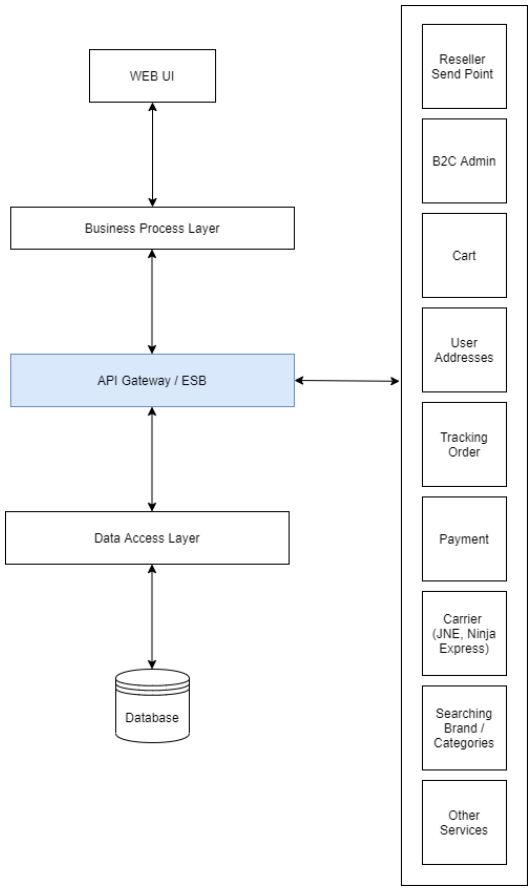

Figure 3: Solution Architecture for Web Application Ditoko.com

4.3 Identification

Process Decomposition is a top-down analysis technique of business domains and business process modeling to identify services, components, and flows contained in Ditoko.com's B2C web application system. From the results of the business process reengineering carried out to integrate the B2C web application system Ditoko.com, the addition of flow and services needed to support changes in business flow are decomposed to produce service candidates

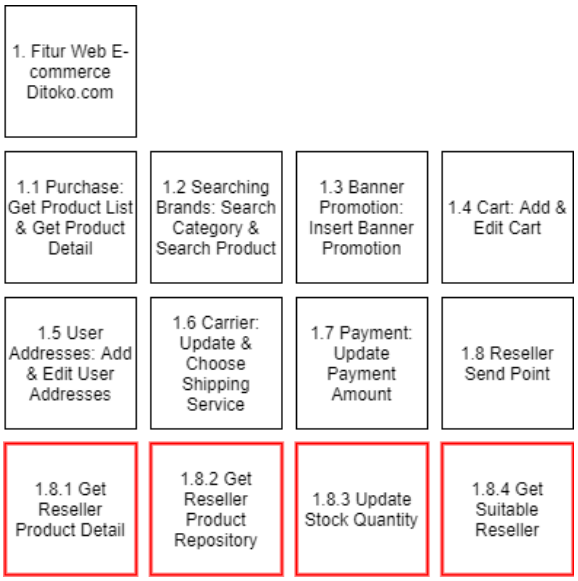

Figure 4: Process Decomposition for Services inside Web Application Ditoko.com

Based on the results of the process decomposition, in Figure. 4, the send point reseller system & online shopping of Ditoko.com which will be integrated with various services are divided into several features, starting from the first box, namely 1. Ditoko.com web features such as Purchase, Searching, Banner, Cart , User Addresses, Carriers, and Payment. Then, the detail service will be the main focus of adding reseller send point services. This reseller send point is divided into 4 main services, including getting a list of product data sold by resellers, getting product details sold by resellers, updating stock quantity, and selecting resellers in accordance with customer requests. This reseller send point will become a series in the online shopping process on the Ditoko.com website, consumers / buyers who buy products through Ditoko.com will get products from the Ditoko.com reseller closest to the buyer's location.

At the stage of rationalize services defined services needed for the online shopping system integration process on the B2C website Ditoko.com. Then refactoring is carried out so that basic services that have an equation are logically classified as a service. To define the web service needed by the current application where integration focuses on adding a Reseller Send Point system to the online shopping system of the Ditoko.com application as a back end of the Reseller Send Point feature and implemented on the Bok Ditoko.com web application. As for Figure 5 explains the service portfolio of the two processes as follows

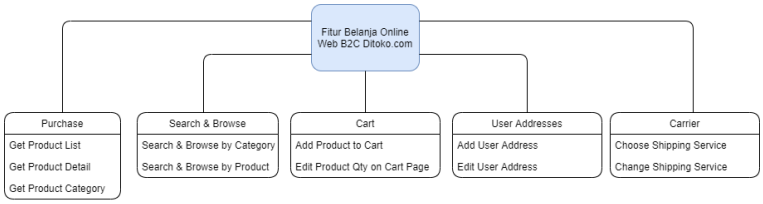

Figure 5: Service Identification Management for Web Application Ditoko.com

4.4 Specification

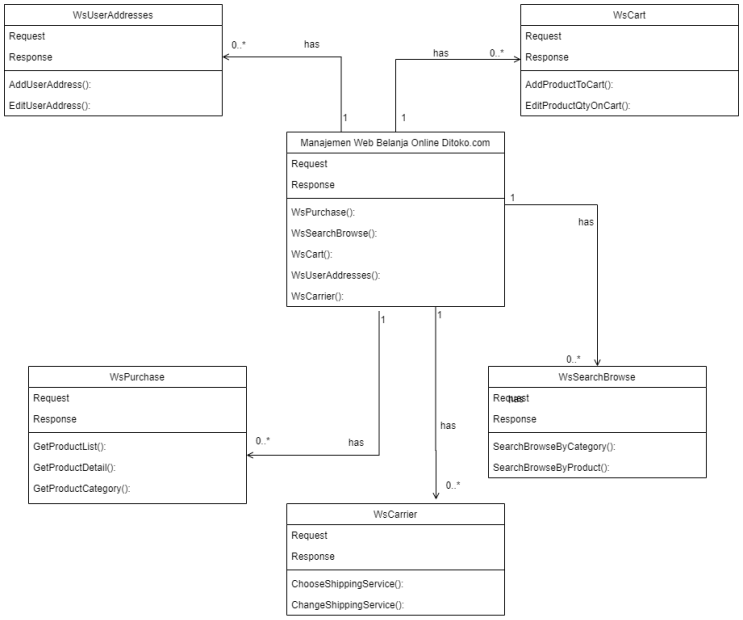

Figure 6: Domain Model Web for Web Application Ditoko.com

Figure 6 describes the domain model of Ditoko.com online shopping web management such as a Purchase service to display product lists and product categories on the home page, Search & Browse to search & browse by product & product category.

In this stage also, the final specifications will be made to form services. Services are built according to the components in more detail based on the specification stage that has been done before.

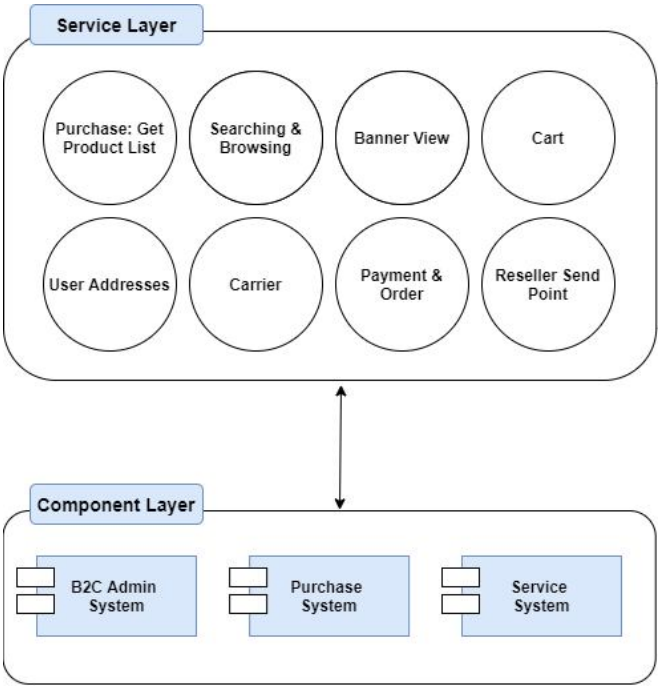

Figure 7: Component Layer for Web Application Ditoko.com

In Figure. 7, Component layer which consists of B2C Admin System, Purchase System requires access to the functions contained in the service layer, so that in Figure. 7 depicted connected to a service layer consisting of 8 services, namely Purchase: Get Product List, Searching & Browsing, Banner View, Carts, User Addresses, Carriers, Payment & Orders and Reseller Send Points.

4.5 Realization

In this Refine Detail Component stage, the components used are defined, and then the components are connected to the service layer and also the operational layer, thus forming an integrated system.

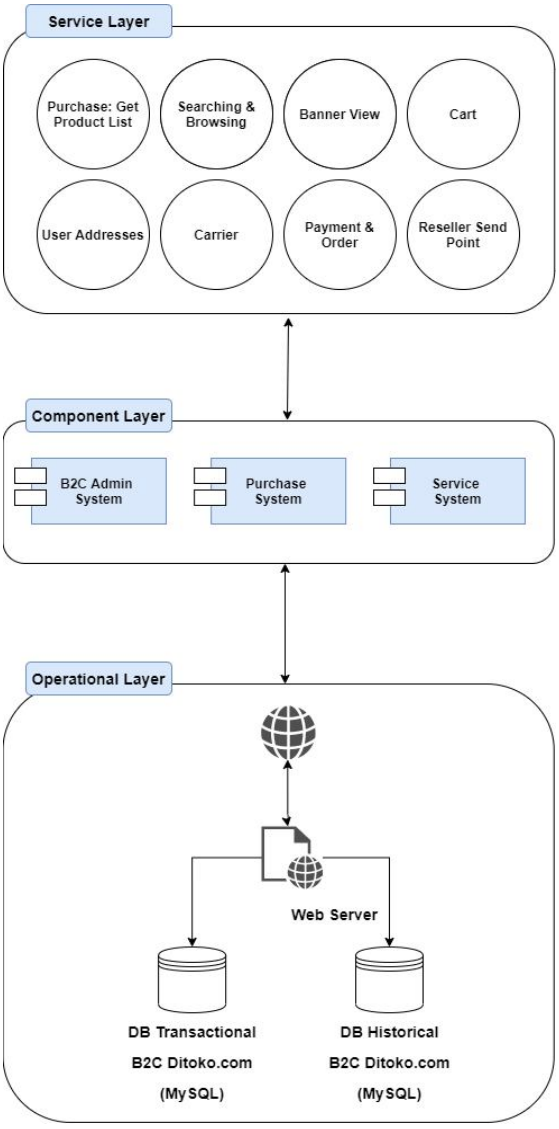

Figure 8: Refine Detail Component for Web Application Ditoko.com

In Figure. 8 explains the relationship between each layer, starting from the bottom, namely the operational layer that uses two MySQL databases where there is a DB that functions to store transactional data & other DBs that function to store historical data from the Ditoko.com Web. Then the component layer consists of B2C Admin, Purchase and Service System. Followed by a service layer that contains services that are used to integrate each application.

SOA Reference Architecture serves as a guide to the development of Adaptive SOA, where in the SOA Reference Architecture, system development is divided into 6 layers, namely the Operational System Layer, Service Component Layer, Service Layer, Business Process Layer, Consumer Layer and Integration Layer using the Gateway API. With the integration of the Gateway API into the system that is already running, the system that will be formed almost fulfills all layers in the SOA Reference Architecture.

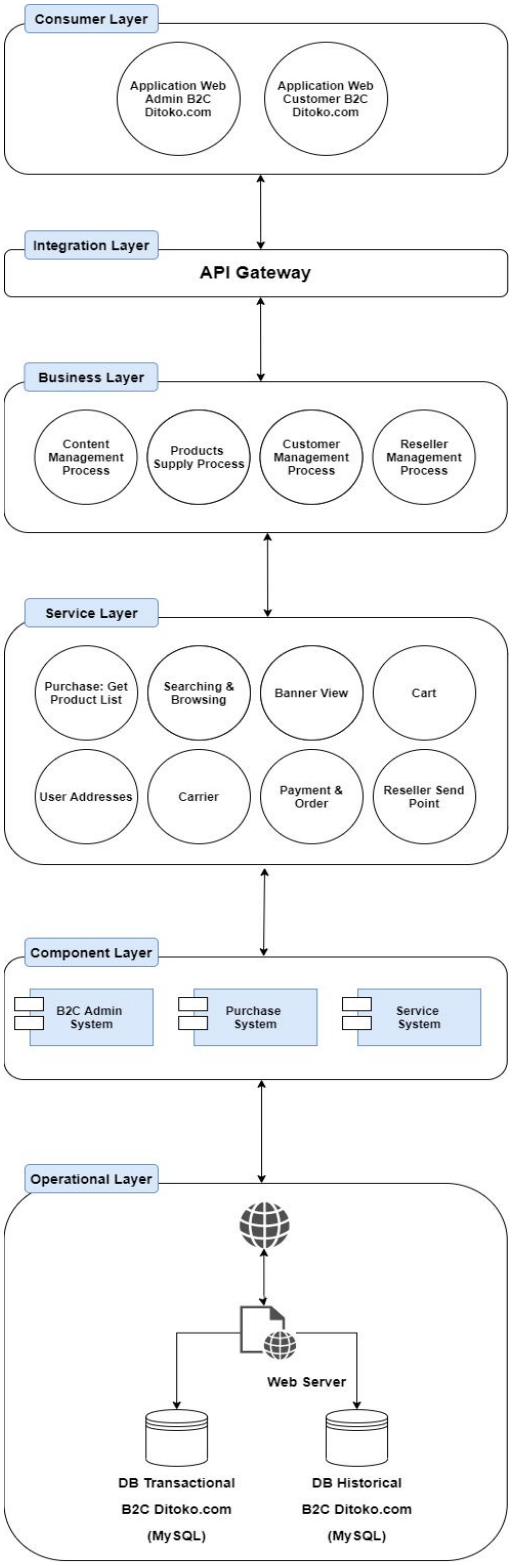

Figure 9: SOA Reference Architecture for Web Application Ditoko.com

4.6 Implementation

The implementation of this system includes the application of the application using the Gateway API then testing each service is divided into two parts, namely for the B2C Admin web management system Ditoko.com and the Ditoko.com Online Shopping System for customers.

The B2C web management process of Ditoko.com uses the Gateway API as a service medium and is integrated with the B2C Admin web application, following an overview of the integration of the Ditoko.com B2C web application and the Ditoko.com B2C web application through the Gateway API in Figure. 10.

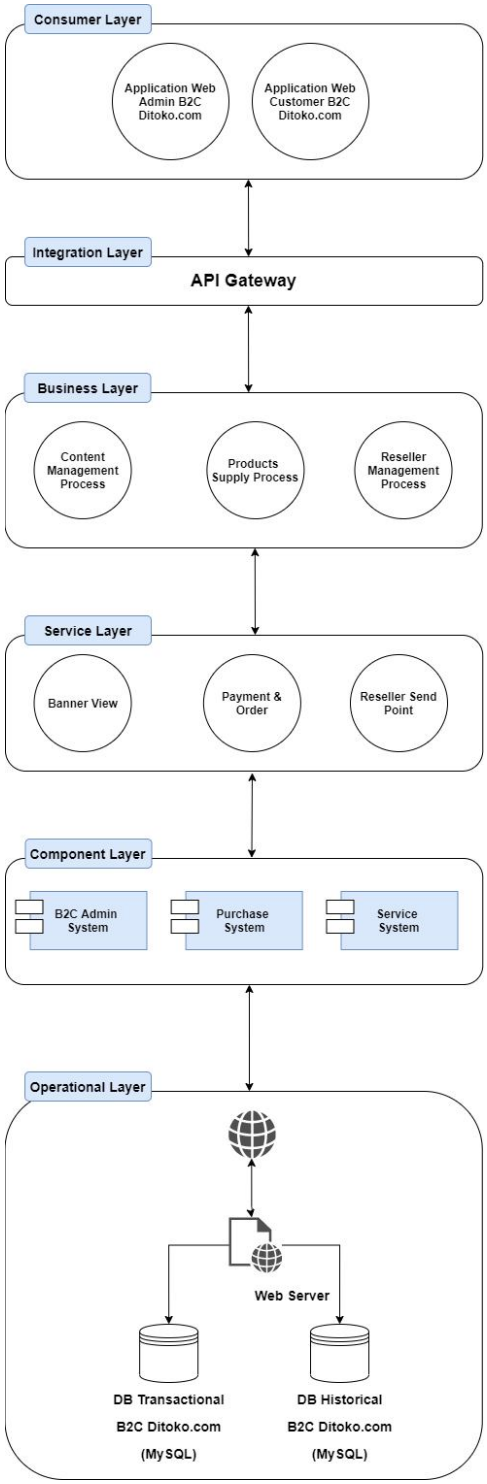

Figure 10: API Integration for Web Application Ditoko.com

4.7 Analysis Performance Results

Testing the available web services by comparing those previously using n to n (monolithic) integration and after being refactored using the Gateway API (microservice) for the following online shopping process with the addition of a send point system reseller to the B2C web application Ditoko.com, the results are shown in Table 1.

Table 1: Comparison of web service testing using n to n integration (monolithic) & API Gateway (microservice)

No.	*Services*	*Systems Side*	*Response Time (ms)*	
			Monolithic	***Microservice***
1	*Banner View*	*Admin*	3.899	1.719
2	*Payment & Order*		4.134	2.997
3	*Reseller Send Point*		N/a	3.953
4	*Purchase*	*Customer*	3.057	2.145
5	*Searching & Browsing*		2.642	1.231
6	*Cart*		2.167	1.198
7	*User Addresses*		2.449	1.224
8	*Carrier & Shipping*		3.542	2.003

In order to obtain the performance test steps for the integration solution, each test case listed in Table. 1 is run 7 times. Using a monitor with a Windows system and the tool used to collect performance measurements is the Latency Test Tools API provided by Nginx which is configured in terms of the test cases listed in Table (Table. 1).

The results achieved by the API Gateway (microservice) in this performance test outperformed the integration of n-to-n (monolithic) views of the response time. Utilization rates for each service run on n-to-n (monolithic) and API Gateway (microservice) integration have different results. The speed is measured in milliseconds (ms), and we can see from the measurement results table that by changing the architecture from monolithic to microservice using the API Gateway has succeeded in increasing the load time of each service & page to an average of **1.7x faster** from the previous.

5. CONCLUSION

SOA design using SOMA Modeling & API Gateway can improve the performance of a customer & buyer activity that uses services on the B2C web application at Ditoko.com. This increase in performance can be seen on services that use API Gateway that has much improvement from the average performance integration of n to n services where n is an integrated application. The implementation of reseller send points on the B2C Ditoko.com web application can facilitate the Ditoko.com admin in doing the automation to make the selection process for the suitable reseller and will certainly help the availability of products that can support the requests and needs of the buyer.

To improve the results that can be obtained from the SOA system using this SOMA Modeling & API Gateway, further development can also be undertaken by implement system monitoring on SOMA Modeling & API Gateway to find out the failure of every service that is accessed, considering that SOMA Modeling & API Gateway is used in applications with a semi-enterprise scale so that it can accommodate the needs of data analysis of each service used. Further assessment of the effect and performance that has been achieved is expected to be arranged in a document so that improvements can be immediately made thereafter.

REFERENCES

1. Nuraeni **Menkominfo Perkirakan Transaksi E-Commerce Capai 25 Miliar Dollar**:, https://kominfo.go.id/content/detail/6983/menkominfo-perkirakan-transaksi-e-commerce-capai-25-miliar-dollar/0/sorotan_media, 2018.
2. Kurniawan, E. **Implementasi Rest Web Service Untuk Sales Order Dan Sales Tracking Berbasis Mobile,** Jurnal EKSIS Vol 07, No 01, May 2014.
3. Cerami, E. **Web Services Essentia,** O'Reilly, 2002.
4. Gottschalk, K. **Introduction to Web Services Architecture,** IBM System Journal, Vol 41, No 2, 2002. https://doi.org/10.1147/sj.412.0170
5. Rahman, A & Kuswardayan, I. **Perancangan dan Implementasi RESTful Web Service untuk Game Sosial Food Merchant Saga pada Perangkat Android,** Jurnal Teknik POMITS Vol 2, No 1, 2013.
6. Tanaem, P Fiodinggo & Manongga, D. **RESTFul Web Service Untuk Sistem Pencatatan Transaksi Studi**

Kasus PT. XYZ, Jurnal Teknik Informatika dan Sistem Informasi Vol 2, No 1 April 2016.

7. Block, G., Cibaro, P., Felix, P., Dierking, H., & Miller, D. **Designing Evolvable Web APIs with ASP.NET**. O'Reilly, 2014.
8. Hidayanto, A. N., Hidayat, L. S., Sandhyaduhita, P. I., & Handayani, P. W. **Examining the relationship of payment system characteristics and behavioural intention in e-payment adoption: a case of Indonesia,** International Journal of Business Information Systems, 19(1), 58-86, 2015. https://doi.org/10.1504/IJBIS.2015.069065
9. Stubbs, J., Moreira, W., & Dooley, R. **Distributed Systems of Microservices Using Docker and Serfnode,** In 2015 7th International Workshop on Science Gateways (IWSG) (pp. 34–39), 2015. https://doi.org/10.1109/IWSG.2015.16
10. Di Francesco, P., Malavolta, I., Lago, P.: **Research on architecting microservices: Trends, focus, and potential for industrial adoption,**. In: Software Architecture (ICSA), 2017 IEEE International Conference on. pp. 21–30. IEEE, 2017 https://doi.org/10.1109/ICSA.2017.24
11. Francesco, P & Lago, P. **Research on Architecting Microservices: Trends, Focus, and Potential for Industrial Adoption**, 2017
12. M. Richards. **Microservices vs. Service-Oriented Architecture,** O'Reilly Media, 2015.
13. Wolff, E.: **Microservices: Flexible Software Architecture,** Addison-Wesley Professional, 2016.
14. Ishida, Y. **Self-Repair Networks: A Mechanism Design** (Vol. 101). Springer, 2015. https://doi.org/10.1007/978-3-319-26447-9
15. Newman, S. **Building Microservices**. O'Reilly Media, Inc, 2015.
16. Thones, J. **Microservices Software**, IEEE, 32(1), 116–116, 2015.
17. Jiang, K., & Song, Q. **A Preliminary Investigation of Container-Based Virtualization in Information Technology Education,** (pp. 149–152), 2015. https://doi.org/10.1145/2808006.2808021
18. Ismail, B. I., Mostajeran Goortani, E., Ab Karim, M. B., Ming Tat, W., Setapa, S., Luke, J. Y., & Hong Hoe, O. **Evaluation of Docker as Edge computing platform,** (pp. 130–135). Presented at the Open Systems (ICOS), 2015 IEEE Conference. https://doi.org/10.1109/ICOS.2015.7377291
19. Ma, X., Yao, Q.-b., & Shan, S. **The Study and Application of System Integration Based on ESB for Enterprise Group,** International Conference on Wireless Communication and Network Engineering (WCNE 2017). Xiamen, China: Journal of DEStech Transactions on Computer Science and Engineering, 2017. https://doi.org/10.12783/dtcse/wcne2017/19912
20. Kurniawan, Kabul & Ashari, Ahmad. (2016). Penerapan SOA menggunakan Enterprise Service Bus pada Proses Pengecekan Status Perizinan Pemerintah Kabupaten Sleman. IJCCS (Indonesian Journal of Computing and Cybernetics Systems). 10. 137. 10.22146/ijccs.15525.