ISSN 2347 - 3983

Volume 8. No. 9, September 2020

International Journal of Emerging Trends in Engineering Research

Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter255892020.pdf

https://doi.org/10.30534/ijeter/2020/255892020

Designing of operations integration of systems using SOA

[1]Muhammad Rizki, [2]Ahmad Nurul Fajar, [3]Adi Fitrianto, [4]Putri Natasya Fanuel

[1]Information Systems Management Department, Binus Graduate Program – Master of Information Systems Management, Bina Nusantara university,Jakarta, Indonesia 11480. muhammad.rizki004@binus.ac.id

[2]Information Systems Management Department, Binus Graduate Program – Master of Information Systems Management, Bina Nusantara university,Jakarta, Indonesia 11480. afajar@binus.edu

[3]Information Systems Management Department, Binus Graduate Program – Master of Information Systems Management, Bina Nusantara university,Jakarta, Indonesia 11480. adi.fitrianto@binus.ac.id

[4]Information Systems Management Department, Binus Graduate Program – Master of Information Systems Management, Bina Nusantara university,Jakarta, Indonesia 11480. putri.fanuel@binus.ac.id

ABSTRACT

Integration in enterprise application between two companies is needed today. It can minimize the effort in software development to adjust and adapt the changes in business processes. In this study, we used SOA and Micro Services Concept to integrate applications for front end and back end mechanism. X is a company engaged in online healthcare and Y is a subsidiary of company X which also acts as a vendor and contributes to the supply of products. In this study, we propose the integration of these two separate applications using the service-oriented architecture concept. SOA concept can support the integration in various platform application in information systems between many organizations. We also used micro services concept to identify and analyze the list of micro services needed. In the result, we proposed reusable services, Restful web services architecture, API endpoint in micro services, and list of micro services.

Key words : Integration, Service Oriented Architecture, Web Service, Micro services, API, Endpoint

1. INTRODUCTION

The implementation of information technology for companies has also become an absolute value that must be achieved to support business activities that are in line with company needs [1]. Information systems that are implemented using information technology today have become one of the effective support tools for companies to support their business smoothly [2]. Company X is a company that engaged in online healthcare. Company Y is a subsidiary of company X which also acts as a vendor. Company Y contributes to the supply of Company X products. At present, company X and company Y have portal applications and ENTERPRISE applications that are built with different programming languages and databases. Company X has the Office Automation application on a purchasing module with Java programming language. While company Y has a portal application on the sales module built with the VB.Net programming language. Both applications are required to have the capacity to do intercompany transactions that are used for the needs of each company. On the other hand, intercompany transactions carried out must be inputted into the ENTERPRISE application at each company. The current condition is that the integration is only limited to data exchange using data parsing in Json and XML formats. Another condition is that until now the data retrieval function in both applications is still using manual query which create repeated input process. It tends to be inefficient because it is very possible for the occurrence of double action in the development activities of the two separate applications in the two related companies. Based on these needs and problems, the concept of Service Oriented Architecture can be one good solution for designing the system. Conceptually, SOA is a form of architectural technology that follows the principles of service orientation (service-oriented) [3]. In this study, we propose a solution in a form of integration framework using Service-Oriented Architecture approach. This approach was chosen because it’s an architectural approach for building applications by using a set of loosely coupled reusable, standards-based, and well-defined services [3]. Difficulties in adjusting the monolith system, given the many things that need to be considered to change a function with a codebase that is already large and complex, include functional dependency in it which is very close. It needs several steps to ensure success including coordination between departments or teams involved and re-tested to ensure changes that have been made do not interfere with the functionality that has been done previously. The migration of the current software system to the micro services introduces many benefits including the ability to distinguish the availability and scalability for various parts of the system, the ability to use different technologies and avoid locking technology or vendors, also reducing time for release and completing a better code base [4].

2. RELATED WORKS

[5] tells about the evolution of micro services, advantages and disadvantages as well as improvements for the future. Researchers conducted a study of micro services in terms of economics and budget as well as the benefits and trade-offs obtained in the implementation of micro services in [6]. It discussed about simple ways to change components by reducing the tendencies that can cause or add new problems throughout the system. Service-Oriented Architecture (SOA) is an architectural approach that improves business agility by building systems with reusable and loosely coupled services [3]. Service-Oriented Architecture (SOA) approach does not have the standard principles used for the development of the SOA. Besides that, SOA has its core principles [7]. Service-Oriented Modeling Architecture (SOMA) is a software development life-cycle method invented and developed by IBM for the design of SOA-based applications [8]. SOMA defines key techniques, as well as its role in an SOA project and its work breakdown structure (WBS). Implementing microservices using domain driven design has been conducted by [9].

3. RESEARCH METHOD

In this study the author decides to use SOMA for SOA development and Domain Driven Design for micro services design. We begin by collecting data needed to get an overview of the business processes of the related module, also to get a general picture of the current system conditions. Then, it is also necessary to identify the problem stages to see the problems faced, especially in the Office Automation application in company X and the application of the portal in company Y

4. RESULTS AND DISCUSSION

Business Modeling and Transformation is the first stage of the SOMA method. It is a declaration or definition of the business processes that are currently running in an organization or company. The result of this stage is a value chain diagram and component business model. From the presentation of business processes that have been carried out at company X and company Y at this time, it is necessary to re-engineer 5 main business processes, namely the process of making purchase orders, the process of confirming the ability of supply, the process of submitting needs, the process of sending goods, and confirmation process for receipt of goods. The concept of SOA in its architecture will improve the effectiveness of work on related units, therefore save time in using the system. Shipping activities is one of the activities that will implement the business process renewal. This is done because the related feature has several services that are declared as reusable services. According to problems that have been explained at the problem identification stage, SOA will be used as the basis architecture for data providers between the company X systems and company Y system along with the ENTERPRISE application at each company. This has an impact on the data management process which will become easier. On the other hand, the SOA method can eliminate the dependence between an application with other applications. Following are solutions for integration on each platform. The business processes of the related applications form the basis of the decomposition process. This stage is carried out to determine the service components that will be used later. The production process at company X is divided into 3 main services in the Office Automation system with a purchasing module that is making planning needs, verifying receipt of goods, and making reports on receipt of goods. Each of these main services is broken down into sub-services as needed. The production of company Y has 2 main services in the portal application with a sales module which are confirmation of the ability of the supply and planning of shipping goods. Each of these main services is also broken down into sub-services according to the needs of company Y. The main service is divided into 4, namely making purchase plans, making acceptance plans, confirming receipt of goods, and overall purchase reports. Each main service is broken down into sub-services according to the needs declared by company X. We create services layer which refers to 2 business processes, namely production at company Y and the process of purchasing products at company X. The service can be used repeatedly or can be called reusable services. From the service layer design that has been made, then it can be determined reusable service that can be used in the design of integration between two related applications. We proposed and described the reusable services in Figure 1 below

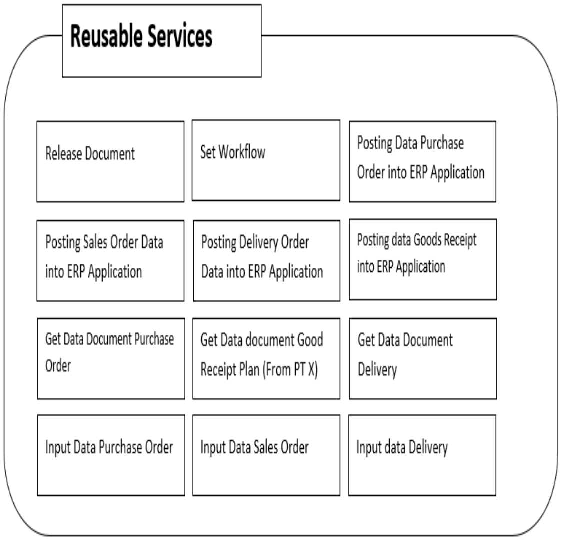
Reusable Services
Release Document
Set Workflow
Posting Data Purchase Order into ERP Application
Posting Sales Order Data into ERP Application
Posting Delivery Order Data into ERP Application
Posting data Goods Receipt into ERP Application
Get Data Document Purchase Order
Get Data document Good Receipt Plan (From PT X)
Get Data Document Delivery
Input Data Purchase Order
Input Data Sales Order
Input data Delivery

Figure 1: Proposed Reusable Services

Then, we proposed control design, that explain what parts or elements are needed for the construction of web services including service providers and consumer services. Service provider is a service provider that moves to call data or actions (functions) needed. Whereas consumer service is part of the end user side who acts as a service caller as needed that can be shown in figure 2 below :

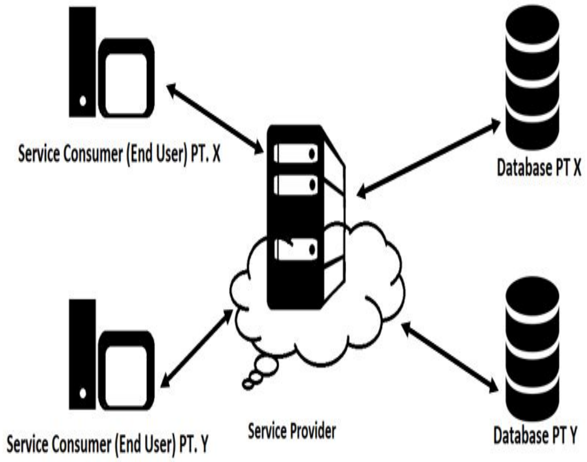
Service Consumer (End User) PT. X
Database PT X
Service Consumer (End User) PT. Y
Service Provider
Database PT Y

Figure 2: RESTFul Web service Architecture

The design also shows that the required functions will later be made in the form of services stored in the service provider section. Technically if there is an application that requires service, then the application can call functions that have been defined on the web server / service provider in the form of xml or json with the RESTful method. For its implementation, the web service or service provider consists of several components in the controller which later become several services that can be called repeatedly by client applications that require it. Service providers contain services that can be called by other applications with certain parameters as need. Figure 1 and figure 2 above shows the SOA mechanism in backend processes. After that, we proposed the mechanism in micro services for frontend functionality. In this stage, we identify the existing endpoints. It can help in mapping the problem for the conversion of monolith architecture to micro services. From the collection of endpoints, it can be further analyzed to make classification based on the functional and domains that interact with these endpoints. Various important endpoints are obtained for further analysis in order to get a big picture of the system about how some products interact with each other, here is a list of endpoints that are used in serving users to interact with the system. Endpoints below are the APIs used by applications that interact directly with the system backend with additional API versioning for future needs.

The list of endpoints above is part of the existing API that is included in the material for discussion with domain experts and developers to trace the domains that interact with these endpoints. After all the required endpoints have been collected, the discussion continues on ownership or ownership of endpoints regarding who has the duty to create, maintain and make modifications to certain endpoints. In this study, we used sample in online healthcare for frontend core processes. The frontend core processes in online healthcare divides into three parts namely Chat, Booking and Content User teams. Endpoint ownership is an important determinant when there are problems and modifications to the API. The list of teams responsible for endpoints is known, the next step is to classify endpoints based on functional domains that can represent all the services available. This can be divided into several services that complement each other in groups that are increasingly focused on specific functionalities. The services including: Common Services, Content Services, Medical Record Services, Campaign Services, User Services, Chat Services, Doctor Chat Services, Doctor Booking Services.

5. CONCLUSION

Based on the results of of research in Office Automation systems used at company X and ENTERPRISE Portal applications that are implemented and already used at company Y, those systems can be integrated using the Service Oriented Architecture (SOA) approach. By using a web service, the maintenance process will be easier because the service can be reused (reusable) so that it can save development time. Functions tend to be the same as the existing system but can be reused. Platform differences between applications are no longer a problem because this web service already uses the concept of loosely coupled, which means it is fully separate and can be accessed by other client applications that need it. In terms of data maintenance, using web services becomes more valid because the data accessed comes from centralized sources. Adopting micro services architecture requires organizational change. Organizations need to adapt according to the new architecture. Each team needs to have ownership of the services they have. Ownership means those who develop, test, install, and maintain services up to the production level

ACKNOWLEDGEMENT

We would like to thank for the research grant for support this work.

REFERENCES

1. W. Guo, *Development of Web Information System of Corporation: An Exploring Research*, 2008 International Symposium on Intelligent Information Technology Application Workshops, China, 2008.

2. Gunawan. S, Togar. A, An Evaluation of Information System Using HOTFIT Model: A Case Study of a Hospital Information System, *2017 International Conference on Information Management and Technology (ICIMTech)*, Yogyakarta, Indonesia, 2017.
3. F. Hojaji, and M. Reza. A.SA Comprehensive SOA Governance Framework Based on COBIT, *2010 3rd International Conference on Computer Science and Information Technology*, Chengdu, China, 2010.
4. Balalaie, A., Heydarnoori, A., Jamshidi, P., Tamburri, D. A., & Lynn, T. (2018). Microservices migration patterns. Software - Practice and Experience. https://doi.org/10.1002/spe.2608
5. Jamshidi, P., Pahl, C., Mendonca, N. C., Lewis, J., & Tilkov, S. (2018). Microservices:
The journey so far and challenges ahead. IEEE Software.
https://doi.org/10.1109/MS.2018.2141039
6. Christudas, B., & Christudas, B. (2019). Microservices Architecture. In Practical Microservices Architectural Patterns. https://doi.org/10.1007/978-1-4842-4501-9_4
7. T. Shancheng, Research on SOA-based Surge Current Test System, *Proceedings of 2011 InternationalConference on Computer Science and Network Technology*, Harbin, China, 2011
8. M. Kumar, G.Kumar, Image Fusion Based On Hybrid SPIHT and SOMA, *2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)*, Ghaziabad, India, 2017
9. Fajar, A.N., Novianti.E, Firmansyah.T. Desing and Implementation of Microservices System Based on Domain Driven Design. International Journal of Emerging Trends in Engineering Research, Vol.8.No.7. 2020