Jurnal Teknik Informatika (JUTIF)
Vol. 3, No. 1, Februari 2022, hlm. 45-48

DOI : https://doi.org/10.20884/1.jutif.2022.3.1.124
p-ISSN: 2723-3863
e-ISSN: 2723-3871

DESIGNING MICROSERVICES ARCHITECTURE FOR SOFTWARE PRODUCT IN STARTUP

Muhammad Rikza Nashrulloh*[1], Ridwan Setiawan[2], Deni Heryanto[3], Ade Sutedi[4], Rickard Elsen[5]

Department of Computer Science, Institut Teknologi Garut, Indonesia
Email: [1]rikza@itg.ac.id

(Naskah masuk: 25 Januari 2022, Revisi : 2 Februari 2022, diterbitkan: 25 Februari 2022)

Abstract

Digital technologies in the world continue to develop in various sectors. This happens not only in developed countries, but also in developing countries such as Indonesia. A trend that continues to develop in the world of information technology becomes a consideration when starting a start-up company. In order to anticipate changes and developments in this trend, it is not uncommon for problems to arise that can directly affect the quality of service for startups. Therefore, a startup needs to create a dynamic culture and infrastructure for the introduction of new technologies. The problems start when startups mature, with more teams, more complex systems, and more traffic to websites or apps. At this stage, startups usually start thinking about scalability issues. to avoid problems, a startup must develop the right architecture for a software product, and a microservices architecture can be a solution to problems. A microservices architecture is an architectural style that structures an application as a collection of small self-contained services modeled around a business domain. This chapter describes the design of microservices architecture for software product in Startup Using Web Service Implementation Methodology.

Keyword. *Designing Microservices Architecture, Startup, Web Service Implementation Lifecycle*

1. Introduction

Many researches have been conducted on enterprise software development[1],[2]. However, it is still rare to find software development activities at startup[3],[4], On the other hand, the existence of startups is now having a great impact on human life, and the habits of information technology and the rapid development of human behavior are beginning to be influenced by lifestyles, learning methods, product purchases, etc.[5],[6]. Today, many people are launching startups with the goal of creating innovative, high-tech creative products and actively growing to reach a wide range of global markets.[7],[8]. The startup has very limited time and resources to build the product [9],[10] Therefore, startups need effective practices to overcome these challenges[11],[12]. Some people believe that startups can find highly beneficial market segments to innovate without the need for large investments, but competitive markets fail most startups. increase [13],[14]. Market-oriented products are important to startups, but they are profitable, sustainable, and the next step in startup growth globally.[15],[16].

Kotakery is an Indonesian startup with the idea to create a mobile and web based crowdfunding and food sales platform, in order for kotakery to be rapidly deployable, scalable and robust, the right software architecture needs to be developed.

The microservice architecture can overcome great complexity, because every need in kotakery will be broken into small parts, but cross-platform applications can be developed, and programming languages can be used according to the needs of the required service. This study is focused on designing microservice architecture for kotakery as a startup using Web Service Implementation Methodology (WSIM).

2. Methodology

In this study, the methodology used to design the microservices architecture is based on the Web Services Implementation Method (WSIM). The development phases of WSIM are requirements, analysis, design, coding, testing and deployment

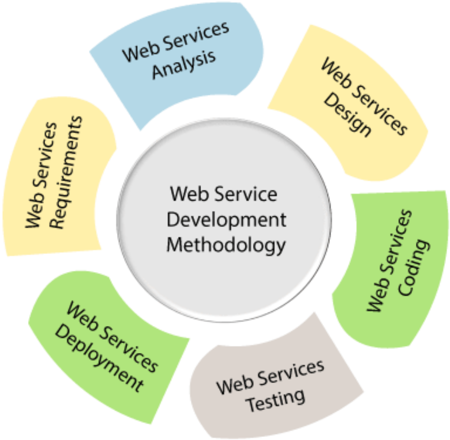

Figure 1. Web Service Implementation Lifecycle

The life cycle of a web service implementation as shown in Figure 1 are :

- Requirements phase: The purpose of the requirements phase is to understand business requirements and translate them into web service requirements. Requirements analysts need to pull out requirements (this is a way to investigate and discover system requirements from users, customers, and other stakeholders). Analysts need to interpret, integrate, and communicate these requirements to the development team. Requirements should be grouped into a centralized repository that can be viewed, prioritized, and mined for interactivity.
- Analysis phase: The purpose of the analysis phase is to improve the web service and transform it into a conceptual model that the technology development team can understand. It also defines a high-level structure to identify web service interface contracts.
- Design phase: At this stage, the detailed design of the Web service is completed. The designer defines the Web service interface contract that has been determined during the analysis phase. The defined Web service interface contract identifies the elements and corresponding data types and interaction patterns between the Web service and the client.
- Coding phase: The coding and debugging phase is very similar to the coding and debugging phase of other software components. The main difference is the creation of additional web service interface wrappers, WSDL generation and client stubs.
- Test Phase: In this phase, the tester performs interoperability testing between the platform and client programs. The test you perform is to make sure that your web service can withstand the maximum load and stress. Other tasks, such as profiling web service applications and inspecting SOAP messages, should also be performed during the test phase.

- Deployment phase : The goal of the deployment phase is to make sure that the web service is properly deployed on a distributed system. It is executed after the testing phase. The main task of the deployer is to ensure that the web service is properly configured and managed. Other optional tasks, such as specifying and registering the web service in the UDDI registry, are also performed during this step.

3. Result and Discussion

At this stage, WSIM has been implemented to develop microservices for kotakery. In the beginning, the developers took a comprehensive and holistic approach to the problem as a whole, rather than being content with a narrow definition of the problem.

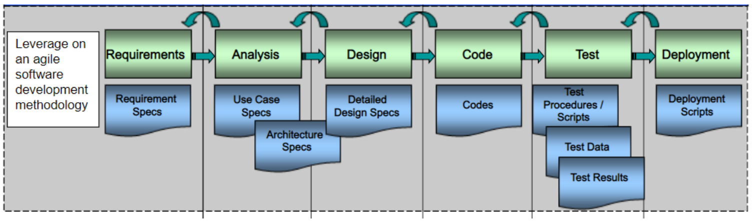

Figure 2. WSIM Activities

3.1. Requirements

Based on the results of previous research, we can conclude that building an application requires several services, such as: (1) Product Data, (2) Send Email, (3) Send Notification, (4) Payment, (5) User data management, (6) Campaign management, (7) User Authentication.

3.2. Analysis

The candidate for service is the thing that is required to be served. Service candidates are selected based on the functional needs of the architecture microservice. Based on the results in the requirements stages, the service candidate obtained as shown in Table 1.

Table 1. Service Candidate

Service Name	**Explanation**
(1)	**(2)**
Email	Used for sending email
Notification	Used for sending push notification
Product	Used for showing product data
Payment	Used for payment method
User	Used for managing user
Campaign	Used for managing campaign
User Auth	Used for user authentication

The service interface functions as a service interface that will interact with the client application. Service technology using REST API. Table 2 explains the service identity interface in a microservice.

Table 2. Service Interface Identification

Service Name	Method
(1)	(2)
Email	POST
Notification	POST
Product	GET, POST, PUT, DELETE
Payment	GET, POST, PUT, DELETE
User	GET, POST, PUT, DELETE
Campaign	GET, POST, PUT, DELETE
User Auth	POST

3.3. *Design*

Design stages used to defines the design of uri (unified resource identifier), database design, Entity Relational Diagram (ERD), and REST API modeling using Restful API modeling Language (RAML).

URI design is intended for each service has a unique uri. Named uri based on service names and parameters to facilitate client applications in access services. Service names and name parameters created based on the tables in the database are taken from the fields in the database. Database on the microservice of kotakery application using MySQL and MongoDB databases. The design of the MySQL database design was made by using PHPMyAdmin, whereas MongoDB database created using Visual Studio Code. The REST API design was created using RAML which will produce several information, namely: (1) Basic information, (2) Security, (3) Data Types, (4) Resources, and (5) Methods.

3.4. *Code*

The stage of implementing the previous stage into a language recognized by the computer. The coding stage is done by building the services using Laravel framework and express framework. Coding starts by providing a service to create an API gateway.

3.5. *Test*

at the testing stage, the method of analysis of boundary values is used. Tests are run to check the entire service by running the http method available on each service and checking the upper and lower limits of the required parameters.

3.6. *Deployment*

During this phase, stakeholders are responsible for planning the transition of the product to the user community, ensuring that these plans are properly executed, addressing issues, and monitoring progress.

4. Conclusions

The microservice architecture contains a mixture of micro and standalone services where each service is self-contained and must be implemented as a single business capability. It is a distinct approach used to develop software systems that focuses on developing several single-function units with clearly defined processes and interfaces. For the startup, the microservice is the right choice for the development of the startup stage, and also to facilitate the development of the application for the developer in the future. Web service implementation methodology to implement the microservice architecture in a startup is can be implemented.

References

[1] N. bin Ali and M. Usman, “A critical appraisal tool for systematic literature reviews in software engineering,” *Inf. Softw. Technol.*, 2019, doi: 10.1016/j.infsof.2019.04.006.

[2] A. Dande *et al.*, *Software Startup Patterns - An Empirical Study*. 2014.

[3] M. Njima and S. Demeyer, “Evolution of Software Product Development in Startup Companies,” 2017.

[4] J. Melegati and A. Goldman, “Requirements engineering in software startups: A grounded theory approach,” 2016, doi: 10.1109/ICE/ITMC39735.2016.9026036.

[5] V. Gupta, J. M. Fernandez-Crehuet, and T. Hanne, “Freelancers in the software development process: A systematic mapping study,” *Processes*. 2020, doi: 10.3390/pr8101215.

[6] A. Maté, J. Peral, J. Trujillo, C. Blanco, D. García-Saiz, and E. Fernández-Medina, “Improving security in NoSQL document databases through model-driven modernization,” *Knowl. Inf. Syst.*, 2021, doi: 10.1007/s10115-021-01589-x.

[7] P. Bouvier, “Ten Reasons Why Fin Tech Startups Fail,” *cbinsights*, 2016. .

[8] F. Kon, D. Cukier, O. Hazzan, and H. Yuklea, “A Panorama of the Israeli Software Startup Ecosystem,” *SSRN Electron. J.*, 2014, doi: 10.2139/ssrn.2441157.

[9] M. Muñoz, M. Negrete, and J. Mejía, “Proposal to Avoid Issues in the DevOps Implementation: A Systematic Literature Review,” 2019, doi: 10.1007/978-3-030-16181-1_63.

[10] E. Klotins, “Software start-ups through an empirical lens: Are start-ups snowflakes?,” 2018.

[11] C. Giardino, X. Wang, and P. Abrahamsson, “Why Early-Stage Software Startups Fail: A Behavioral Framework BT - Software Business. Towards Continuous Value Delivery,” 2014.

[12] T. Vera, S. F. Ochoa, and D. Perovich, “Development Effort Estimation Practices in Small Software Companies: An Exploratory Study,” 2020, doi: 10.1109/SCCC51225.2020.9281161.

[13] I. Moroni, A. Arruda, and K. Araujo, “The Design and Technological Innovation: How to Understand the Growth of Startups Companies in Competitive Business Environment,” *Procedia Manuf.*, 2015, doi: 10.1016/j.promfg.2015.07.361.

[14] O. Zaks, “Success and failure in M&As: Is there a place for a paradigm change? Evidence from the Israeli hi-tech industry,” *Econ. Bus. Rev.*, 2016, doi: 10.18559/ebr.2016.1.6.

[15] S. Kaewunruen, P. Rungskunroch, and J. Welsh, “A digital-twin evaluation of Net Zero Energy Building for existing buildings,” *Sustain.*, 2019, doi: 10.3390/su11010159.

[16] H. M. Al-Mubaraki, A. H. Muhammad, and M. Busler, “Categories of incubator success: a case study of three New York incubator programmes,” *World J. Sci. Technol. Sustain. Dev.*, 2015, doi: 10.1108/wjstsd-06-2014-0006.