

Implementation of microservices architecture on certification information system (case study: LSP P1 State Polytechnic of Malang)

I F Rozi*, R Ariyanto, A N Pramudita, D R Yunianto and I F Putra

Department of Information Technology, State Polytechnic of Malang, Malang, Indonesia

*imam.rozi@polinema.ac.id

Abstract. Currently, Technology develops rapidly. This influence in every aspect of life in every field. Ease of use, tools availability, and a higher data transfer rate are the reasons why technology is used massively, it affects certification field as well. Certificates given through certification process is a formal recognition towards professionals in higher education level. The process of certification data is accessed through the website. Usually built with monolith approach which means an application wrapped up in a huge package. When there are any changes in the codes, it will also affect the others. This makes the system more complex, hard to fix the bugs, and hard to do modularization. Microservices architecture offers a slightly different approach, it is about a software architecture's concept that allows us to make large-scale software be more organized and can quickly adapt to the existing business development. To speed up deployment and testing process by taking advantage of continuous integration, so the development process is more consistent and able to checks the flow of application periodically to reduce possible errors.

1. Introduction

These days, Information Technology develops rapidly, this affects every aspect of life, whether it is economy, politics, arts, up to education. This makes an impact on every aspect of life in every field. Practical use, tools availability, and higher data processing rate are the reasons why technology is used massively, it affects certification field as well.

Certification Program is an effort to improve someone's quality in the education field and increase their welfare by periodically encourage them to improve their professionalism. The certificate given through certification process is a formal recognition towards professionals in higher education level. At this moment, data processing for certification can be accessed through the website.

Generally, an Information system is built by monolith approach where an application packaged in a huge container when there are any changes in the codes, it will make changes the others. This leads to a mandatory to make good processing information's from raw data so that it can transfer information in each available system. The problem is about the adaptation needed in every change, mainly on code complexity and its maintenance. Where this could lead to a bottleneck in the distribution process because of the high dependencies [1]. That's why we have to consider other architecture which can adapt well on every change.



Development of Technology and Research has been speeding up for years. Recently, there is a term in the Software Engineering field that is called “Microservices”. Microservices architecture offers slightly different approach. A software architecture that possibly be a solution to make big-scale software more organized and more adaptive to changes of business needs [2], Conceptually, Microservices is a software architecture to develop an application that consist of small services, runs by their own process and communicate each other by using API as the medium and by HTTP/HTTPS protocol [3].

Information System of Competency Certification is a web-based portal system that is used by Badan Nasional Sertifikasi Profesi. As a notable institution, it’s information system on Competency Certification is still adopt a monolith architecture. This may cause some difficulties in advancing the development, testing, repairmen, and deployment when monolith application goes bigger and more complex.

2. Literature review

2.1. *Microservice*

Making a conclusion from the main point that makes microservices is microservices. That microservices gives “services” as a component. From this point, then comes up with the newest development paradigms. One of the paradigms is Decentralized Management [2].

Microservice architecture is a software architecture that requires split application on business [4], whereas in research Auth0 is explained that microservice architecture is a new architecture which application development can be done with small web services that interact each other [5]. There are some benefits from microservice architecture [6]:

- Codes written in Microservice architecture is much lesser and we can do application testing independently.
- Easier maintenance of the software
- Ease of Scalability
- Developers can freely choose various programming language and framework

2.2. *Monolith*

Monolithic Architecture is an architecture of application that runs all the logic in a single application server [3]. As a result, Application built with monolithic architecture only needs maintenance on a single server. In research Fowler M, it is explained that monolithic architecture is a software application that its modules could not run independently [6]. By using monolithic architecture, each application should run at the same time, and each module requires a lot of storage. There are disadvantages of using monolithic architecture [6]:

- Requires huge size, and more difficult to do maintenance and application development
- May occur dependency conflict and the app needs restart when there are additional modules included
- May occur code conflicts when doing deployment at the same time
- Monolithic architecture has limited scalability
- Only uses one technology to create an application with monolithic architecture

2.3. *Continuous integration*

It is an approach that encourages engineers to integrate their jobs that includes implementing new feature. Bug fixes, update documentation periodically and do Continuous Delivery after Continuous Integration finished based on the criteria [6].

2.4. REST API

REST (Representational State Transfer) is a standard web-based architecture that implemented on web-based application development. Usually using HTTP (Hypertext Transfer Protocol) as protocol to transfer data. API is a specification from a data structure, object, function, and parameters needed to access source of application. All specification above formed an interface that belongs to the application to communicate with other application, an API can be used with various programming language, or by using URL (Uniform Resource Locator) that already provided by the website [3].

2.5. API Gateway

API Gateway is a gate among services. The gateway becomes entrance of all request, then forward the request to authorized service [7].

2.6. Java

Java is Object Oriented Programming language and can be applied to various operating system and open source [4].

2.7. Spring

It is a framework that is core support for dependency injection. It means that the developers can integrate with other frameworks. Most used Spring Framework is Spring Boot, Spring Data dan Spring Cloud [8].

3. Result and discussion

3.1. System architecture

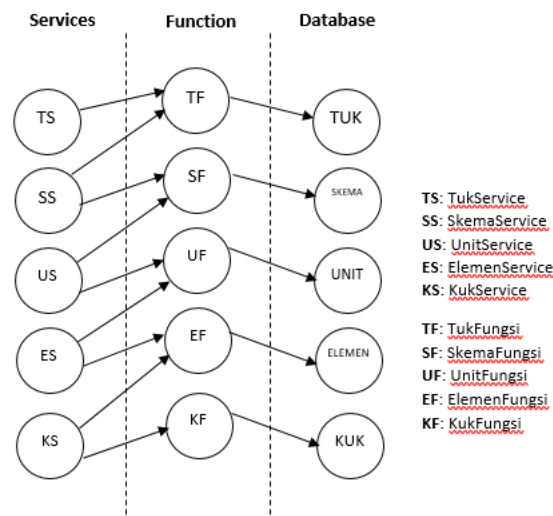
This research focused on system development based on microservice where REST is used as an intermediary from each service and data transfer by JSON so that the data can be processed on frontend system. In the Figure 1, it needs a portal to access each service that runs independently. API Gateway as a solution to translate protocol between backend and frontend. In frontend, data processing from backend is done by the endpoint. Discovery Service, services that are used to record the list of all service.

3.2. Architecture service system analysis

The service architecture analysis process is done by describing the connection between the services that are running. Services available on Information System Certification, are:

- Service Tuk, serve as service to manage data of Competency Test Site
- Service Unit Kompetensi, serve as service to manage data of Competency Unit
- Service Skema Kompetensi, serve as service to manage data of Competency Schemes
- Service Kode Unit Kompetensi, serve as service to manage Competency Unit Code
- Service Elemen Kompetensi, serve as service to manage Elements of competence
- Service Pemohon, serve as service to manage Applicant from the User who is registering
- Service Pekerjaan, serve as service to manage Job owned by Applicant
- Service Asesmen, serve as service to manage owned by Applicant
- Service Admin, serve as service to manage admin data
- Service Pemohon, serve as service to manage Applicantdata
- Service Asesor, serve as service to manage Assessor data

To find out the relationship between services, a dependency graph is made between services that are running. Following are the steps to form a candidate service component for decomposition of business functions according to operations on each entity by making a dependency graph.

**Figure 1.** Services dependency.

In figure 1 shows the dependency between services, business functions contained in each service, just as SS has a dependency with TF business functions to access data from Tuk entities.

3.3. Functional system analysis

Microservices that run on backend includes REST must have a token authentication that is protected by JWT(JSONWebToken) [5]. Communication between frontend and backend is asynchronous. As a result, server doesn't have to remember the session just like usual communication pattern client-server on web-based application.

Table 1. Functional testing.

No	User	Testing	Results	
			Valid	Not Valid
1	Admin	Login in the system	v	
		See Tuk	v	
		Change Tuk	v	
		Remove Tuk	v	
		See Skema	v	
		Change Skema	v	
		Create Skema	v	
		Remove Skema	v	
		CreateUnit	v	
		Remove Unit	v	
		Change Unit	v	
		See Unit	v	
		See Kuk	v	
		Create Kuk	v	
		Remove Kuk	v	
		Create Elemen	v	
		Change Elemen	v	
		Remove Elemen	v	
		See Elemen	v	
2	Applicant	Add Berkas	v	
		Change Berkas	v	
		Remove Berkas	v	
		Add Pekerjaan	v	
		Change Pekerjaan	v	
3	Assessor	Change Asesi	v	
		Add Asesi	v	
		Remove Asesi	v	

Based on the results of the functional testing table, the basic functions that must be fulfilled by the system can run as expected to support other processes. Analysis of research obtained from a new architecture where application development is carried out in the form of a fire rest that communicates with each other is different from monolithic using one application server so that it requires a thorough application maintenance process as a whole. System testing as shown in table 1, is successful in processing data from REST servers into a database that is passed through an endpoint on the fire gateway. The testing process is inputting JSON through swagger UI to return the appropriate response.

3.4. Domain-Driven Design

Domain-Driven Design (DDD) is a knowledge about how to use technology to help business process. So that the software development can be more accurate based on business process that will be implemented on an application. There are steps on using DDD method:

- Analyze a functional system to know what business process included in the application
- Categorize running functional system to know what the similarity of each function between the business process is.
- Identify the relation of functional system

3.5. Deployment

Deployment process using auto-mechanism to detect integration issue as early as possible. The system will compile and test automatically so on every backend services so it can speed up deployment [6] process. In the figure 2, integration process continuously uses some tools and platform:

- GitHub
- TravisCI
- Heroku

By using GitHub as repository storage from each service and automation testing will be done by TravisCI. When testing process met the requirements, then do automatic deployment on Heroku.

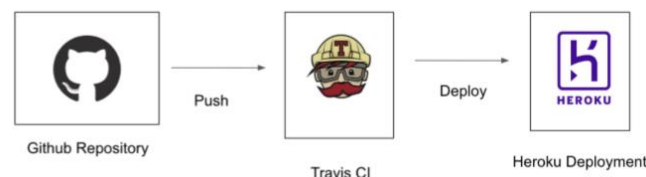


Figure 2. Continuous integration.

The Stages conducted based on figure 2, namely

- Push the application code from a repository, the storage used is GitHub
- Then the TravisCI build/test and deploy the service on the server and Implement the service on the server using Heroku
- This process is implemented into each – each service backend

4. Conclusion

From all the result of this research, we can conclude some points. System developed using microservices architecture, divide the system to a smaller component that is called service. Currently, the system is divided into 12 small components, and the service is divided based on domain. By taking advantage of continuous integration, we can do automatic and continuous deployment by using tools TravisCI and Heroku as a service provider for deployment.

References

- [1] Suryotrisongko H 2017 Arsitektur Microservice Untuk Resiliensi Sistem Informasi *J. Sisfo.* **6**(2) 235–250
- [2] Fowler M 2014 Microservices Resource Guide
- [3] Liew Z 2006 *Smashing Magazine* [Online], retrieved from: <https://www.smashingmagazine.com/2018/01/understaending-using-rest-API/>. 2018
- [4] Oracle O n.d *Java Documentation* [Online], retrieved frim: <https://docs.oracle.com/en/java/>
- [5] Auth0 *JSON Web Token* [Online], retrieved from: <https://jwt.io/introduction/>
- [6] Fowler M 2006 Continuous Integration 2006
- [7] Enterprise C 2016 *API Gateway* [Online], retrieved from: <Http://callistaenterprise.se/erbjudanden/uppdrag/apigw/>.
- [8] Pivotal Software I *Spring Framework* [Online], retrieved from: <https://spring.io/>