3. Verification of Business Agility Improvements by Applying a Serverless Architecture

　In this chapter, we built a website for purchasing goods using a serverless architecture, and compare it with the construction work in an on-premises environment, and describe the result of verifying whether the development period will be reduced.

　The scope of verification is the part of system development that is developed by the IT department using a serverless architecture based on the business strategy determined by the management.

　The reduction of the development period is verified from the following two points of view.

　・Amount of documentation

　・Construction time

3.1 Hypotheses and Premises

　Hypotheses and Premises in the verification process are described below.

3.1.1 Hypotheses

　We hypothesize that the development period will be reduced by decreasing the amount of documents in each stage of the system development process and reducing the construction time by applying a serverless architecture,.

3.1.2 Premises

　Premises for the verification process are described below.

(1)Cloud service providers

In this verification, we built a website for purchasing goods using a serverless service provided by Amazon Web Services (AWS), a leading cloud service provider. [3]

(2)Reasons for choosing the model system

Japanese government is promoting cashless payments. So we believe that the number of e-commerce site launches will increase.

"The Current Status and Significance of Cashless Payment by the Ministry of Economy, Trade and Industry" [4] describes that the advantages of promoting cashless payment are "increased efficiency and sales at stores" and "data utilization". When a retailer launches an e-commerce site, it can expect to increase sales by diversifying its sales channels. Also, it is possible to link to marketing and product development by analyzing and utilizing the data of purchase information on the site.

We believe that the number of e-commerce sites will increase because cashless payment is the main means of payment and it provides the advantages mentioned above.

(3)Building a goods purchasing website

The requirements for the goods purchasing website are summarized in the requirements specification (Appendix A-1), and we constructed a system that satisfies the requirements. The serverless architecture to be applied is based on the architecture design published in AWS samples, which is quality assured by AWS. We reused Serverless Airline Booking (an example of an airline ticket purchasing system) [5], which has similar to our requirements, as the reference architecture. And we designed a system that meets our requirements. (Figure 2)

Figure 2 System Configuration diagram of the goods purchasing website

3.2 Verification Method

The measurement of construction time and Amount of documentation was performed by the following procedures.

3.2.1 Construction time

The system build work was carried out in the following procedures from A-1 to A-4. The construction time was measured as the time required for steps A-3 and A-4, which are the construction work itself. The building process was assigned to one member who had no experience with AWS development.

Step A-1: Based on our requirements specification, we referred to the AWS Developer Guide to learn about the design and configuration required for the selected AWS service in advance.

Step A-2: Summarize the data used in the system as a data definition document (Appendix B-1). Summarize the values set in the AWS Management Console as a parameter sheet (Appendix B-2) for each function to be implemented in the goods purchasing website.

Step A-3: Develop the application source code for a goods purchasing website using the runtime supported by Lambda. Code development is divided among the members.

Step A-4: Build a goods purchasing website using the AWS Management Console and AWS CLI according to the data definitions and parameter sheets, and measure the working time. ~~The system building process is handled by one member who is inexperienced in AWS development, with the exception of the implementation of the authentication functions required to launch the Step Functions state machine from an AppSync query.~~

3.2.2 Amount of documentation

We measured the amount of documents according to the following procedure B-1 to B-4.

Step B-1: Make a list of deliverables that are required for waterfall model development in on-premise environment. The list of deliverables has been standardized list based on the findings of the system construction that each company to which the member belongs.

Step B-2: 3.2.1 after the construction, set up additional deliverables required for the development by applying a serverless architecture to the list of deliverables made in step B-1.

Step B-3: ~~Create a document for on-premise development to compare and evaluation with the amount of work required to create the same document in development by applying a serverless architecture.~~

Compares the amount of document between on-premise development and development with a serverless architecture. The evaluation is in the following three stages.

　・Same workload as on-premise

　・Reduced workload as on-premise

　・No documentation required

Step B-4: For documents that are rated as "Reduced workload" or "No documentation required", provide a rationale for the merits of applying a serverless architecture.

Step B-5: The results of the above steps B-1 to B-4 are summarized as an evaluation sheet (Appendix C-1). The issues that emerged during the system build work of 3.2.1 are also described in the evaluation sheet.

3.3 Verification results

The measurement results of the construction time and the amount of documentation are described below.

3.3.1 Construction time

The total construction time of the goods purchasing website (total of steps A-3 and A-4 in 3.2.1) was 10 hours. Step A-3: Development of application source code took 7 hours in total, and Step A-4: Set up time using the AWS management console and AWS CLI took 3 hours.

3.3.2 Amount of documentation

As shown in Figure 3, 68 documents were required for the on-premises development, and 56 documents were required for a serverless architecture. So we reduced 12 documents in total. (18% of the documents created during on-premise development).

Here is the detail

　・Same workload as on-premise development:38

　・Reduced workload as on-premise development:18(26% of the documentation created during on-premise development)