E-Government Integration Based on SOA for Supporting Sleman Smart Regency (A Case Study of Sleman Regency, Special Region of Yogyakarta)

Damar Suryo Sasono[1], Djoko Budiyanto Setyohadi [2], Albertus Joko Santoso [3]
{damar10rpl@gmail.com[1], djoko@mail.uajy.ac.id[2], albjoko@staff.uajy.ac.id[3]}

Magister Teknik Informatika, Universitas Atma Jaya Yogyakarta, Yogyakarta, Indonesia 55281[123]

Abstract. Integrated government system has become considered in the world of government, especially in developing countries. SOA can be sufficient in performing the Integration with a different platform of applications, and web service is used to transfer data. The web services are rated as the most appropriate technology for implementing the SOA concept. JSON data format is easy to use and recognized in various programming languages. For the above reasons API Manager as Government Service Bus (GSB) need to create, which will be responsible for the management of the service and the interaction within the facility. With the Integration they can serve Government to Enterprises (G2B), Government to Citizens (G2C) and Government to Government (G2G) with the services. With this integration expected can be realized Sleman Smart Regency as the Incarnation of the smart city.

Keywords: software architecture, web services, smart city, e-government, SOA, SOM, GSB.

1 Introduction

E-government is a tool which is aimed to improve the performance by supporting government operations. E-government here uses the internet and technology networks that exist in the organization to provide public services to the public and existing business. E-government is not just building a website portal, but it should be more. It is being constructed by almost every local government to change the way an organization completes missions within the government.

All the national governments of the world have begun to make substantial resources to create the environment and infrastructure to conduct business electronically with their business citizens and other government entities[1]. Different application platforms covering all areas of e-government implementation must be run to provide integrated government services to Communities, businesses, and governments[2]. The use of SOA (Service Oriented Architecture) [3] can be fulfilled and provides a modern application architecture for interaction between existing distributed systems and new systems to be developed.

When many applications on different platforms are operated, and data integration among application is needed, Service Oriented Middleware (SOM) is required to bridge services between applications. In this case, the Government Service Bus (GSB), an application called API Manager, is created.

ICCSET 2018, October 25-26, Kudus, Indonesia

DOI 10.4108/eai.24-10-2018.2280597

2 E-Govermnent

Electronic Government (e-government), also known as digital government and online government. It refers to the use of internet technology as a platform to exchange information or data and provide services and transactions with citizens, businesses, and other government arms[4]. As a system analyst, the electronic governments have three requirements: (1) all e-government services based on electronic information hardware systems, and digital networking technologies to connect from one system to another, (2) integrated system becomes essential to handle the public aspiration and issues, and (3) newly developed, advanced, and revolutionary to support the government management system better than ever. Integrated e-government has two or more departmental systems that can interconnect and interact to exchange information. It means that the government system can access other systems for service requests. The e-government portal provides services in the form of information services or process flow services [3].

Figure 1 describes the details of the government information sources on domain structures in a component diagram. There are two simple main components, display level on government information display, and service level on department service provider. The display levels show the required information to the citizen or internal department which is usually needed by the chairman of some department. Furthermore, service level contains many applications, and each application produces some services sent information to simple web service management and consumed by information display to provide new information.

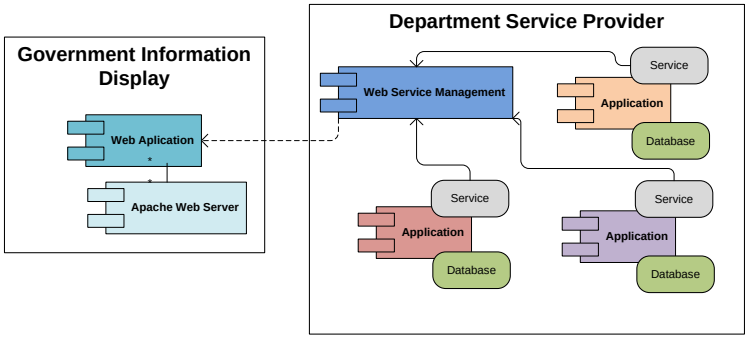

Fig. 1. Government architecture model.

3 Service Oriented Architecture

Service-oriented architecture (SOA) is a new application which has architectural style and principles. SOA consists of four core components namely: service provider, service broker, service consumer, and service description (as shown in Figure 2)[1]. The service consumer searches for required services in the service broker and the service provider offers a list of registered app services. The service description directly can find the service description in the service registry can bind and run the facility. The service broker provides and maintains the service registry, even though the current public registrar is not in vogue.

SOA also provides solutions for shared and distributed services and to achieve high interoperability, flexibility, and standardization values by utilizing descriptions, findings, and service requests[2]. This architecture style is suitable for e-government to implement integration system. SOA is based on architectural principles which facilitate the structuring and execution of various services. It is used to satisfy business goals within the system. It is a Service-oriented application uses existing infrastructure and new systems to support rapid, cheap, and simple distributed applications. A service is a work unit performed by a service provider. It is aimed to achieve the need of a service user. Moreover, the defined service is implemented in a standard way. Services can also be used and re-accessed as needed. Service is generally referred to the entity of the application. The services interact with service in other existing applications and combined into required information. In an information it can be generated from one service or some services.

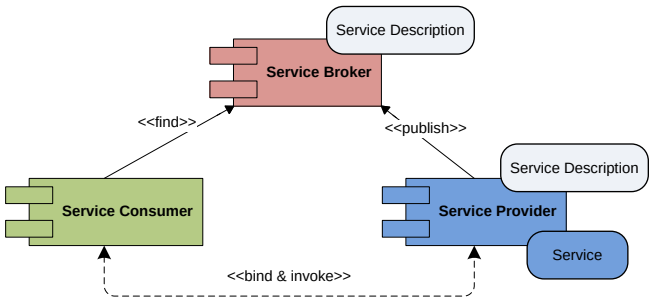

Fig. 2. SOA architectural style and principles.

4 Research Research Methodology

This research aims to develop an API manager integration system based on service-oriented architecture. This integration will involve many of the services and platforms of information technology. The following are the phases done in the study, shown in Figure 3 [5].

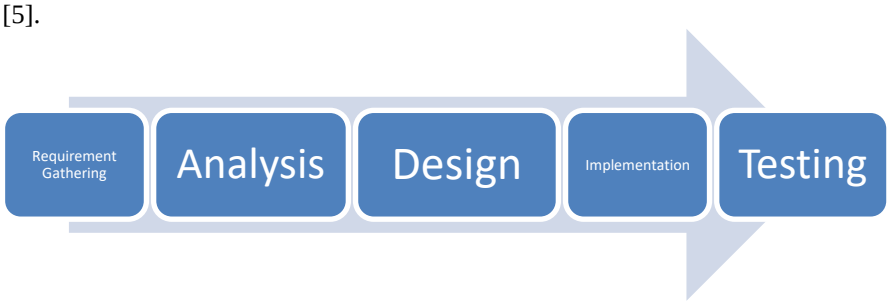

Fig. 3. Research Methodology.

4.1 Requirement Gathering

Requirement gathering is the first phase. This phase will be the collection and analysis of requirements to develop a service-oriented architecture of API manager. First, we conducted several interviews with some government officials to collect the necessary data and information. These data were analyzed and identified case studies to develop API manager. Interviews were conducted with the head of the sub-organization of information in the communication & informatics department. The following questions were addressed during the meeting :

a) What kind of data and what information is needed to create an API manager for information integration?
b) Where the necessary data and information can be retrieved from and where is data originated?
c) Who owns and manages the data and information?
d) What kind of platforms the pirates, information systems, and databases they use to manage data and information, in what form is the data exist?
e) How to design and implement API manager in Seman district?

The above questions have been asked to obtain detailed information about the API manager needs so that it can be used to analyze and identify the API manager development needs.

4.2 Existing Service Identification & Case Study Analysis

This section will discuss the analysis of existing application systems and determine the information systems and databases that will be the object of the case of this case study. First, observations were made. They were made to all departments in Sleman to find out the implemented information systems, business processes, and database platforms used for data storage and processing. The applications or databases selected in this case study are relevant to system integration issues, required data sources, and relationships between applications or databases.

From the results of covering some of the following application, agencies are used as the research object, e.g., staffing application, Evaluation of Regional Development application, asset application, and Regional Planning application. Table 1 shows the selected application as an object of the research.

Table 1. Selected Application as an Object of The Research.

Application	Owner	IP Address
Staffing	Employment agencies, education, and training	192.168.80.25
Evaluation of Regional Development	Regional planning and development agencies	192.168.80.26
Asset	Regional Finance and Assets Agency	192.168.90.49
Regional Planning	Regional planning and development agencies	192.168.80.26

Staffing application is an application that serves to manage employees throughout the district of Sleman ranging from employees additional, employee editing, job history, and training. Evaluation of Regional Development is an application that serves to see the monthly

realization of each activity that existed in the agency ranging from physical and financial realization. Asset applications are applications that work to manage assets across agencies across all Sleman districts from additional asset to asset removal. The regional planning application is an application that serves to manage budgets in Sleman district ranging from agency proposals to regional budgets.

4.3 Design & Implementation

Ticket Request, The API Client accesses the Server API by sending username and API parameters to be accessed (username, app, module, action). If the API Client username has access rights to the API Destination, then the API Server will create a ticket (log) for access and submit the ticket to the API Client and proceed to process 2, otherwise, if the API Client username does not access API will send an error message.

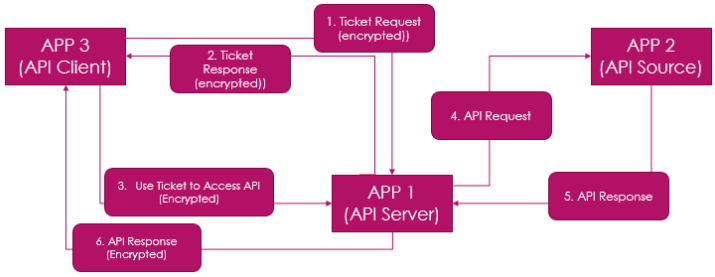

Fig. 4. API Flow.

Ticket Response, The API Server sends the result of checking the permissions in the form of ticket code if there is access or an error message if there is no access.

Use Ticket to access API, The API Client accesses the API Server by sending the ticket and parameter used for the Source API (e.g. NIP, date, etc.). If the ticket is valid (never used), then the API Server will process further according to the destination API's cache_internal setting: a) if the destination API's cache_internal setting is 0 then the API Server will proceed to process 4, and make the output of process 4 as the result of the API; b) if the destination API's cache_internal setting is greater than 0, e.,g. 60, then the server API will look for API results with the same parameters over the past 60 seconds, and make it an API result (go to process 6). If an API result with the same parameter in the time range is missing, the feed will be processed like point a.

API Request (From API Server to API Source), The API server processes requests from the API client (constructs the Source Source API, and its parameters) to access the Source API. API Response (From API Source to API Server), API Source sends API request results from the API Server.

API Response (From API Server to API Client), The API Server sends the results from the API Source to the API Client. The API Server will record the results according to the destination API log settings: a) If setting access_log! = 1 then the log data will be deleted (no logs), b) If setting access_log = 1 then log data will be recorded as follows: 1) if response_log = file, eat the result of the API will be recorded on file. 2) if response_log = db, then the result of the API will be recorded in the response column database. 3) if response_log = none, then

the result of the API will not be recorded. c) the response_src column will contain the corresponding API source (cache-file, cache-db, or API).

As shown in Figure 5, the network used in this research is the local network owned by the government of Sleman. The API manager and the staffing application are located in one block IP 192.168.90.xx.

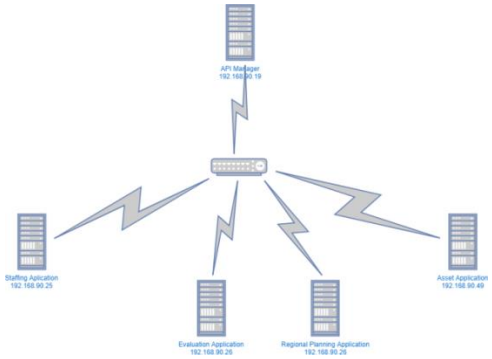

Fig. 5. Physical Network Architecture.

4.4 Functional & Performance Testing

We perform functional testing to determine the functionality of the applications that have been used in each agency. This test is done by running applications and services developed and implemented in each process. Based on functional test API Manager can display information taken in real time from the existing service.

Performance testing is performed to determine the performance of the API manager that have been developed and ready for the trial. Currently Performance testing, we use a computer with core i7 specifications with a speed of 2700 MHz CPU, 8.0 GB RAM, the operating system used is Windows, and runs through Sleman local network. The test is performed to determine the average execution time (AET).

Table 2. The Result of Performance Testing.

Application	Modules	AET (Second)
Staffing	listStaff	0,0186
Staffing	getPegawaiList	3,4977
Evaluation	getListKegiatanDistinct	1,2467
Evaluation	RekapRUP	0,0434
Regional Planning	listPPAS	1,2215
Regional Planning	RekapIndikatorProgram	0,0317
Asset	list_aset	1,7183
	Average	1,1111

5 Conclusion

The results show that the API manager can be used as data integration mediators and information systems especially system integration processes in Sleman District. In addition, performance test results show that the average execution time (AET) for service operations amounts to 1.1111 seconds. These results indicate that the execution time is considered to be sufficient so that the developed services are considered not to interfere the integration process.

References

[1] A. Arsanjani, Service-oriented modeling and architecture. IBM Dev. Work., no. January, 2004.

[2] S. M. Baraka, R. S.madoukh, "A conceptual SOA-based framework for e-Government central database." IEEE CITS 2012 - 2012 Int. Conf. Comput. Inf. Telecommun. Syst, 2012.

[3] K. A. Sedek, M. A. Omar, and S. Sulaiman, "Interoperable SOA-Based Architecture for E-Government Portal." 2012.

[4] Z. Nanping and L. Yuan, "Study and application of the SOA based E-Government system," Proc. Int. Conf. Inf. Manag. Int. Conf. Inf. Manag. Innov. Manag. Ind. Eng. ICIII 2008, vol. 2, pp. 476–479, 2008.

[5] K. Kurniawan and A. Ahmad, "Service Orchestration using Enterprise Service Bus for Real-time Government Executive Dashboard System," 2015 Int. Conf. Data Softw. Eng., pp. 207–212, 2015.