E-Government Interoperability Framework based on a Real Time Architecture

Aris Puji Widodo[1], Jazi Eko Istiyanto[2], Retantyo Wardoyo[3], and Purwo Santoso[4]

**[1, 2, 3] Department of Computer Science and Electronics, Gadjah Mada University
Yogyakarta, Indonesia, 55287**

**[4] Department of Government Faculty Social and Political Sciences, Gadjah Mada University
Yogyakarta, Indonesia, 55287**

Abstract

One of E-Government (E-Gov) implementations is to produce a form of relationship namely Government to Government (G2G). The main factor of G2G concerning interoperability between central and local governments is the heterogeneity in developing E-Gov applications. The interoperability that has occurred are done by creating a brand new architecture that has a function to allow existing E-Gov applications to be able to communicate with each other. E-Govs in Indonesia are developed separately by the respective governments, i.e. central (national) and local (province, district/city). Therefore, interoperability and data integration have such a restriction.

This paper focused on creating an Enterprise Architecture Framework (EAF) for interoperability and data integration on E-Gov applications in Indonesia. This EAF combines the approach of Service Oriented Architecture (SOA) and Event Driven Architecture (EDA). The purpose of this combination is to produce EAF that is real-time, the relationship between the one-to-many services, and the message transmission models that are asynchronous. The services provided and the events that are used to trigger are defined by using Web Service Definition Language (WSDL), while the orchestration mechanism that occurs between services is defined by using the Business Process Execution Language (BPEL). BPEL is conducted by the Government Service Bus (GSB). To evaluate this EAF, the model of services which has been defined using a variety of data is measured based on execusion time.

Keywords: *Enterprise Architecture, E-Gov, SOA, EDA, Real Time Architecture.*

1. Introduction

E-Gov is used to improve the relationship between the central and the regional governments, between regions and other government parties, especially the citizens [1]. Implementation of e-Gov can produce a form of relationship such as: Government to Citizen (G2C), Government to Business (G2B) and Government to Government (G2G) [2].

As observed in Indonesia, the implementation of E-Gov in the level of preparation (web site design) is considered very good, meaning that almost every local government, central government and government agencies have a web site and the information is updated regularly [3]. The web site that is designed by the local government, central government or agency became a way to form a G2C relationship in terms of sharing information more openly to the citizens. In transaction and transformation level, the focus is still in separated development using a wide variety of development platforms [3][4]. The interoperability process that occurs between local government, central government and government institutions is still done by exchanging physical data storage media, such as floppy disks, compact disks and other storage media [5]. This condition means that the shape of the relationship of G2G on e-Gov in Indonesia has not worked well, and do not meet the spirit of the policy in INPRES No. 3 Year 2003 and the blue print of E-Gov applications systems [6] [7].

One of the main factors regarding the form of G2G relationships is the interoperability between local and central government, so the exchange, collecting and integration of data between local and central government can be achieved [8]. One of issues and challenges in conducting interoperability is the heterogeneity of the e-Gov applications that currently exist. The heterogeneity of existing e-Gov applications includes technology, architecture and platform which are used by each local government based on capability and knowledge. In order to create the interoperability of heterogeneous systems, the approach is to build an interoperability framework architecture that can support the existing previous architectures [9]. The approach of interoperability framework architecture aims to interconnect the applications of e-Gov that are separated and isolated so they are able to communicate with each other [10].

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.

Interoperability framework architecture is developed with protocol-based approach and Extensible Markup Language (XML) scheme, which are distributed by a public network [11][12]. By using this protocol based framework architecture and XML schema, the interoperability process that occurs can only be done by exchanging XML scheme that is an open standard and does not provide services. To add services to the architecture of XML that are based on interoperability framework, an approach was used based on Service Oriented Architecture (SOA) using the web services technology [10][13][14][15][16][17][18]. The services provided in a SOA architecture approach are non-real-time to the business processes since the services provided are only based on request and response information, and are synchronous messaging [19].

In order to provide services that function in real time to the business process, this paper is focused on creating architectural framework for interoperability and data integration using a combination of SOA and EDA-based approach. This SOA approach that was used is based on the concept of making the communication architecture. This architecture can perform a wide range of heterogeneous platforms of existing e-Gov applications. The communication is represented among open standard services. The EDA approach that was used is based on the concept for provision of dynamic services. The combination of SOA and EDA approach aims to provide services that can be real time for business processes, because the services are available in a combination with the event as a trigger for asynchronous message sending.

Concerning the condition of e-Gov in Indonesian government which has heterogeneity ranging from technology, architecture and platform development, the architecture can be used as standard development of e-Gov in Indonesia in order to achieve interoperability between central and local governments, so later on it can lead to data integration as a whole.

2. SOA

SOA is a form of an architectural model that refers to the principle of technology-oriented service [20]. Figure 1 describes the scope of the service in SOA which consists of functions, procedures or a process that gives response if requested by a client or it can also be viewed as a logical encapsulation from an individual or a group of specific activities as the implementation of business processes. The concept of service-oriented allows to be approached by dividing the problem into a set of small service, and the solutions to these problems must be solved by allowing all to participate in a service orchestration. SOA is not associated with a particular technology, but more toward a modular approach.

To implement application with service-oriented approach to SOA, the implementation is in a layer that lies between business process layer and application layer. This layer is a part of the enterprise logic called the service interface layer, as shown in Figure 2. Service interface layer has a function to perform logical encapsulation in application logic, as well as the business processes that exist in business logic. Thus, the application can be more modular and be able to use a variety of technology platforms. Interface service layer is divided into several layers of abstraction: application service layer, business service layer, and the orchestration service layer [20].

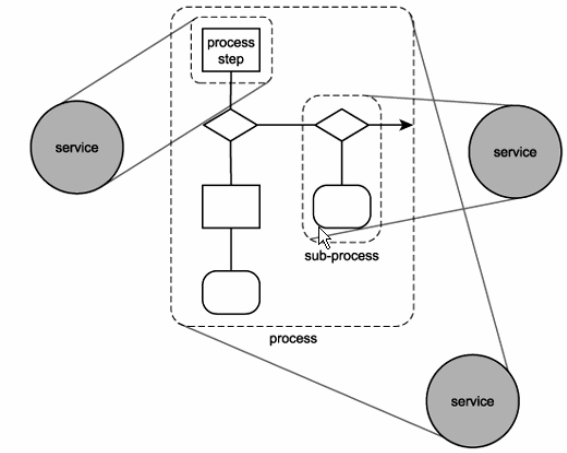

Fig. 1 Encapsulation Business Process with Services [20]

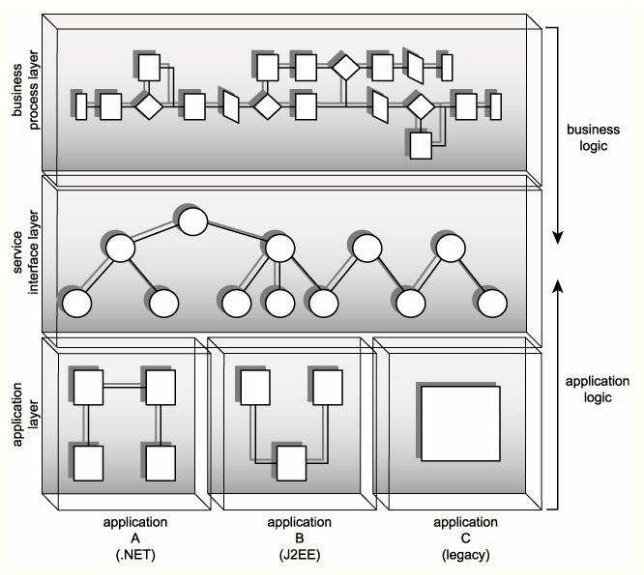

Fig. 2 The layers in Implementation on Enterprise [20]

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.

Conceptually, SOA approach is based on the architecture depicting interaction among the main parts. These parts include Service Provider, Consumer Service, Service Registry, and Service Broker [21]. SOA uses find-bind-execute paradigm, where Service Providers register their service in the public registry. Then the registry is used by users to find services that match the desired criteria. If in the registry there is service that meets the needs, then the user will be given a contract and the addresses of the service [21].

3. EDA

An event is an occurance of something inside or outside the business process. It can be used to identify something that is happening or anything that will occur. Each occurrence of an event, this event has a header and a body [22]. Event header contains elements that describe the occurrence of such event, such as: identity, type, name, time-stamp, number, and event makers. The body contains the event that has occurred. In the concept of EDA approach, for each event that occurs inside or outside a business process, that event is immediately distributed to all parties having interest on it; then the parties will conduct the evaluation, and optionally take action in response to the event. EDA is extreme loosely coupled, and highly distributed. It means the event developer will know about the occurrence of the event, but does not have the knowledge about which parties will process the event and the people who has the interest [22]. In order to ease the search of an event, EDA will use an asynchronous flow model, where the flow of events begins at the time it is created and it ends at downstream execution of the action [22].

4. SOA and Web Services

IT architecture that was developed with heterogeneous platform has a challenge during application and data integration. The interoperability between the applications must ensure that the integration of business processes is more effective without affecting the availability of applications that exist previously. The distributed technology that already exists, such as CORBA, COM/DCOM, EJB, JAVA RMI and others can be used to implement interoperability between applications, but should be in the same development platform for each application [13]. In order to perform interoperability on heterogeneous platforms, SOA approach which is based on web services technology can be used. Web services technology has open standards, so it allows applications that are implemented by Web services platforms/different vendors to be able to communicate with each other [23]. Basically a service in the SOA is an application. This application represents the business logic (automation logic) of a large system that encapsulates the process, which must be able to stand alone and communicate with each other [23]. Figure 3 describes the communication between one of the service with other services that are defined using the Web Service Description Language (WSDL). WSDL describes the format of a message to be sent by web application services, so it can be understood by other web application services that receive the technology using Simple Object Access Protocol (SOAP) running on the Hyper Text Transfer Protocol (HTTP) [20].

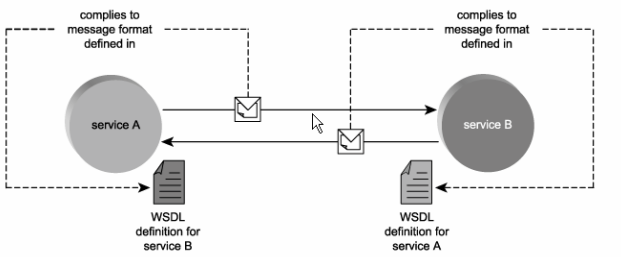

Fig. 3 Services Communication [20]

Web services are technology that is used to exchange information based on XML. It allows interoperability between multiple applications and platforms that are developed with different programming languages. SOA approach enables applications to communicate with other applications by requesting relevant service of respected application regardless of the platform and the programming language. Applications can adapt to the functions and services to suit the different processes in a flexible business processes [13].

5. Business Process Execution Language (BPEL)

BPEL is an XML-based language used to define business processes with web services. The main goal of BPEL is to standardize the business process flow that is defined to work with using the web services [24]. BPEL extends the interaction with web services model and enables to support the transaction. BPEL is based on web services, so it is assumed that each of the business processes involved is implemented by using web services, so that BPEL can regulate the interaction between web services using XML documents [25]. BPEL is used to describe a business process in two different ways, i.e. executable and abstract processes. The executable processes specification is to define the order of execution in a number of activities; the partners that are involved, exchanging messages between partners, and exception handling mechanisms. The abstract

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.

process specification is the behavior of exchanging messages with different parties without providing information about the internal behavior from the parties. The elements contained in the general structure of BPEL are composed of tag: Process, PartnerLinks, Variables, and Sequence [26]. Process is a main key element of the BPEL. The process name is defined on the tag having attribute name. In addition, the tag is also used to include information related to the process definition. PartnerLinks is an element to define the type of port from another service that participates in the execution of business processes. Variables are elements used to store the status that is used for workflow logic. Sequence is an element to organize a set of activities that can be executed consecutively. The elements that BPEL supports for sequence include receive, assign, invoke and reply.

6. Interoperability Architecture Conceptual

In this section is given a conceptual model of framework interoperability of E-Gov in Indonesia which can be used for interoperability of local governments to the central government or in the other way around. The case study used is the application of Information Systems and Population Administration (ISPA) developed by the Directorate General of Population Administration Department of Home Affairs [5].

6.1 Existing E-Gov Interoperability

ISPA applications that run on the sub-district, district/city and province consist of two versions, namely online ISPA and offline ISPA [5]. Among offline ISPA versions that are installed at each district/city (an application that has function for recording data of population registration and data of civil registration and for printing the population data), and province (an application that has a function to recap on the report of demographic data), each of them has its own applications and database. In offline ISPA, the data interoperability process is done by exchanging physical data storage media, such as floppy disks, CDs, and other storage media. Conversely, online ISPA is developed by a centralized architecture application, so the data and the application are located in one place, namely the Directorate General of Population Administration Department of Home Affairs. ISPA is planned to be accessed online at the district level (the application that has a function for recording data of population registration and data of civil registration and for printing the population data) that will be directly connected to the Directorate General of Population Administration Department of Home Affairs. The communications infrastructure of online ISPA will use VPN Dial with synchronous connections to the data center in the Directorate General of Population Administration Department of Home Affairs. The interoperability process of online ISPA is done directly because it was developed with centralized architecture [5].

6.2 Developed e-Gov Interoperability Architecture

Offline and online ISPA have limitations in terms of interoperability in both the application and the data. Online ISPA who are with Dial VPN connection models should be able to provide assurance that the connection to the data center in the Directorate General of Population Administration Department of Home Affairs will not break up, because when the connection is lost, ISPA does not operate. The number of districts is 5263. If the districts access at the same time, it will require a considerable amount of bandwidth. The IT infrastructure in the district level has not been perfect throughout Indonesia. Offline and online ISPA are developed by using client-server architecture.

Based on ISPA that is currently in operation, there should be an improvement pertaining to the architecture for better interoperability for applications and data. Interoperability architecture was developed using SOA approach to create services for business process, so interoperability can be performed by accessing these services. To create interoperability architecture that operates in real time, SOA approach needs to be combined with the EDA approach. The aim for this combination is to enable the service works with the event as a trigger, where the events in the services are defined using WSDL. To set the service orchestration which has occurred, a bus service called the Government Service Bus (GSB) is defined using BPEL. The development of interoperability architecture consists of the components shown in Figure 4:

1. Client (Services Consumer)
   This component is used to access the services provided by the service provider.
2. Services Provider
   This component is used to define business processes in a service using WSDL.
3. Event
   This component is used to trigger the services that are defined using WSDL.
4. BPEL
   This component is used to set the trigger based on orchestration between the services provided by the event.
5. GSB
   This component is used to perform detection, to trigger, and to distribute event (event services). Besides, this component also provides assurance that different protocols of heterogeneous platforms are

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.

able to communicate. It is also responsible for transforming the message (to replace or to add a message) throughout the participating services (mediation service).

Interoperability scenarios are developed using an approach where each district/city, province, and national levels are assumed as elements involved in the mechanism of interoperability. Each element can act as a service provider as well as a service consumer. Architecture interoperability scenarios were developed using the model given in Figure 4 as follows:

1. National and province governments as a service provider for the functions of adding, deleting, and updating that will be used by the consumer service in districts/cities. This scenario is used to create a distributed database contained in each district/city, province, and national levels.
2. District/city as a service provider for the function of read data (data on population, birth data, mortality data, recap of the population, recap of the number of births, and recap the number of deaths) that will be used by the service consumer, i.e. province and national level in accordance with the hierarchical structure.
3. Each province and nationally has GSB that is equipped with BPEL to set orchestration between the services that have occurred. GSB is also defined as an event (the read data) that is used to trigger between the services performed by the BPEL orchestration. This scenario is based on the conditions in Indonesia that has 33 provinces, and each province has some districts/cities, thus it requires the orchestration mechanism for interoperability between services owned by each of the elements involved.

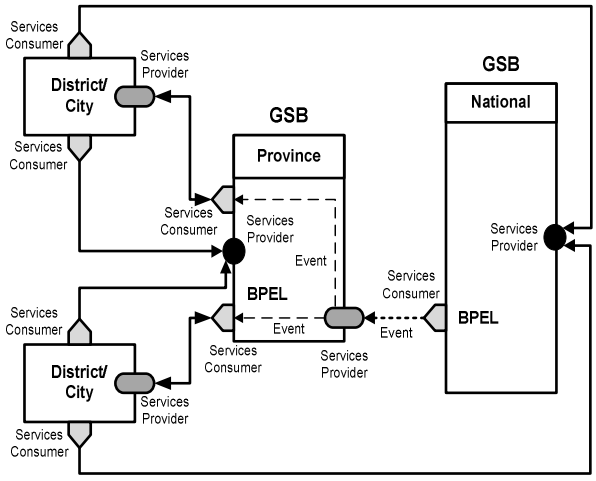

Fig. 4. Conceptual Interoperability Architecture

7. Implementation of Interoperability Architecture

To measure the validity of the conceptual interoperability architecture model, the conceptual model was then implementated as a simulation. To perform the simulation, there were used a variety of development platforms at the district/city, province, and national levels as given in Table 1.

Table 1: Development Platforms

Element	Programming Language	Database System
District/City	Php	MySQL
Province	Java	Postgre SQL
National	Java	SQL Server

The services that are used to define the base of consumer services and service providers will be placed on each of the elements, as given in Table 2.

Table 2 : Services Definition

Element	Services Consumer	Services Provider
District/City	Function add, delete, update	Read functions which includes: population data, birth data, mortality data, recap of the population, recap of the number of births, and recap the number of deaths
Province	Read functions which includes: population data, birth data, mortality data, recap of the population, recap of the number of births, and recap the number of deaths	Function add, delete, update
National	Read functions which includes: population data, birth data, mortality data, recap of the population, recap of the number of births, and recap the number of deaths	Function add, delete, update

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.

The services used to add, delete, and update are placed in the province and national use one-to-one scenario without using BPEL, but using WSDL to be directly accessed by the district/city. This scenario aims to create a distributed database for each province and national level through a mechanism that is done together. That is when the district/city adds the population to the data, then the province and national levels will also be added to the population data. Then, the read services are placed at the district/city and province using one-to-many scenario where the event acts as a trigger while BPEL sets orchestration with other services as a partner. To be able to detect events and orchestration between services, it is necessary to use GSB. GSB which is used is GlassFish 3.1.2 server as an open ESB. The mechanism of message exchange uses In-Out model, which is a two-way messaging model to satisfy the asynchronous message exchanges. This architecture has a consumer service and a service provider, as well as other services unit including different orchestration that has occurred. It needs to be in a combined package called the service assembly as shown in Figure 5.

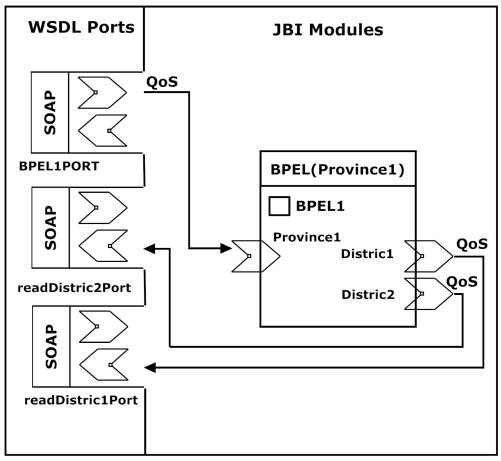

Fig. 5 Service Assemble in Province level

In Figure 5, Java Business Integration (JBI) module consists of several components that interact with the services model. The components that are used as a supply or consumption of services that are performed by the BPEL Service Engine will provide services in order to execute business processes. The executable business processes is defined by using BPEL which is an XML-based language. Execution includes the exchange of business process or business process orchestration messages between the web services with other services as partner. The contract between business processes with service is defined by the WSDL partner. Messages that are exchanged between business processes with service partners are wrapped into WSDL according to the JBI specification; then JBI will be redirected to the Normalized Message Router (NMR). NMR interacts with external web service through binding components. Binding component is responsible for wrapping the detailed protocol message exchange.

In Figure 6 the national does a request that consists of events and parameters of the data. Then the assignment is done by copying an event and the parameter data to be used in invokation as input parameter. Event and data parameters provide notification to the province, where the notification is then used as a request to the district/city based on events that are requested by the national. In BPEL of provinces as seen in Figure 7, event and parameters request are accepted as notification from the national, then re-assign and re-invoke are performed. Furthermore, based on the corresponding event the sequence province BPEL service requests are made to the district/city. District/city will give response to assign the appropriate parameters in response to invoke the BPEL province, in a way of parameter data concat from a number of districts/cities. Then the province also assigns the appropriate parameters response in the response from the national BPEL invokes, by using the parameter data concat from a number of provinces. Results that are given as a response to the national is in the form of parameter data that are obtained from the data in districts/cities in accordance with the national event given as requested.

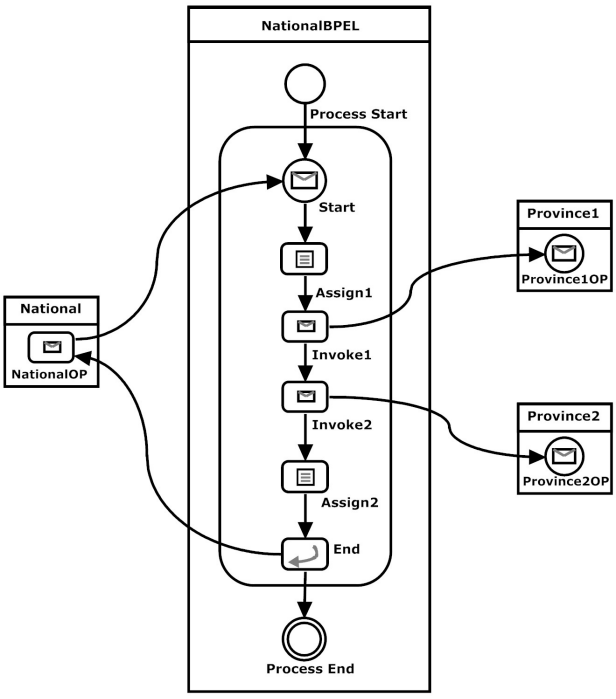

Fig. 6 National BPEL

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.

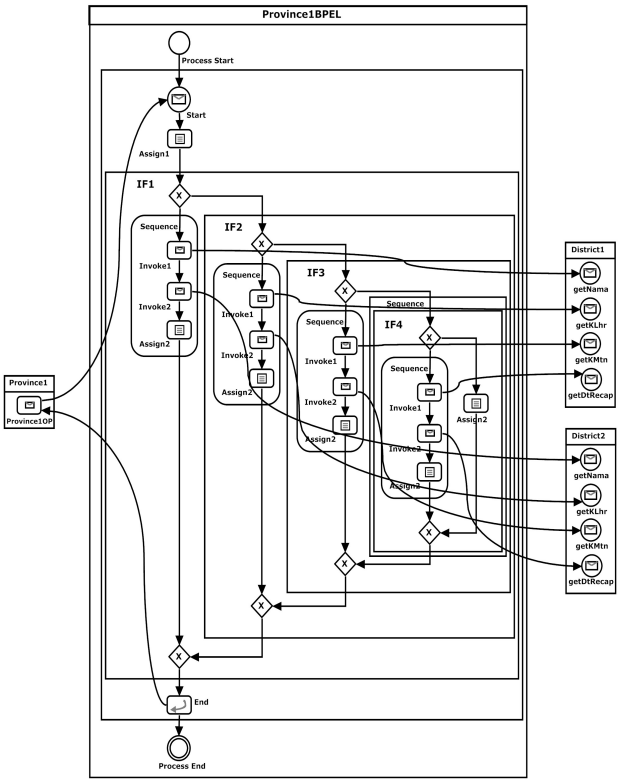

Fig. 7 Province BPEL

8. Performance Measure

In order to evaluate the interoperability architecture that has been generated, a measurement was done by calculating execution time using a number of mixed data on all services models that were defined. The environment that was used to evaluate the amount of time uses Intel Core 2 Duo 2.40 GHz processor with 2 megabyte-memory and 100 GB sized hard disk. The evaluation was done by measuring the length of the execution time for an access using BPEL (connections was not indirect to the database of the districts/cities, but using the services performed by orchestration), access without BPEL (direct connection to the districts/cities database), and LOCAL (direct national connection to the local database obtained from add, delete, and update service). This evaluation aims to determine the feasibility of the model services defined in the EAF by comparing the length of time between the execution of BPEL, without BPEL, and LOCAL. Table 3 presents the results of measurements of the length of execution time to show the recap amount of data on population by provinces, districts, sub-districts and villages. Table 4 presents the results of measurements of the length of execution time to show the whole population data.

Table 3 : Execution Time Recap of National Population

Number of Data (record)	Per Recap	Without BPEL (sec)	BPEL (sec)	LOCAL (sec)
6500	Province	7.833	0.492	0.201
	District	5.975	0.112	0.067
	Sub District	6.349	0.227	0.079
	Village	5.842	0.218	0.082
13000	Province	13.571	0.765	0.281
	District	10.662	0.187	0.124
	Sub District	11.992	0.374	0.202
	Village	11.721	0.312	0.202
19500	Province	20.443	1.153	0.369
	District	17.876	0.211	0.164
	Sub District	18.781	0.399	0.292
	Village	17.692	0.335	0.287
26000	Province	25.993	1.683	0.499
	District	23.723	0.322	0.183
	Sub District	22.918	0.413	0.399
	Village	21.769	0.467	0.401
32500	Province	33.842	2.118	0.572
	District	31.384	0.488	0.231
	Sub District	31.542	0.516	0.421
	Village	32.738	0.534	0.437
39000	Province	41.883	2.668	0.722
	District	38.667	0.589	0.301
	Sub District	38.994	0.692	0.481
	Village	38.478	0.671	0.492
45500	Province	53.881	3.276	0.871
	District	50.773	0.608	0.316
	Sub District	51.692	0.721	0.502
	Village	51.462	0.716	0.512

Execution time for using BPEL in Tables 3 and 4 were not used when the execution was first performed, because it was needed to compile the module and class firstly so it took a long time. The next acceses did not require module and class compilation, so the time was relatively shorter and stable for subsequent accesses.

Tabel 4 : Execution Time to Display National Population Data

Number of Data (record)	Without BPEL (sec)	BPEL (sec)	LOCAL (sec)
6500	35.233	17.354	1.742
13000	68.344	32.479	2.344
19500	110.276	47.199	3.988
26000	138.553	58.831	5.736
32500	167.439	70.773	7.271
39000	194.761	86.359	8.709
45500	222.992	98.794	10.851

In Figure 8 the time difference between that of BPEL and that of LOCAL execution was not significant, because the response given by the model of service was in the form of a single data output parameter. In the comparison between using BPEL and without BPEL, there was a significant

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.

difference because the connection was done in direct access to the database on each district/city. Therefore, more resource consumption was required.

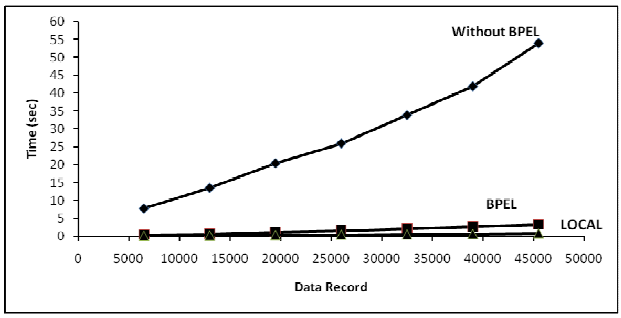

Fig. 8 Recap of National Population Execution Time

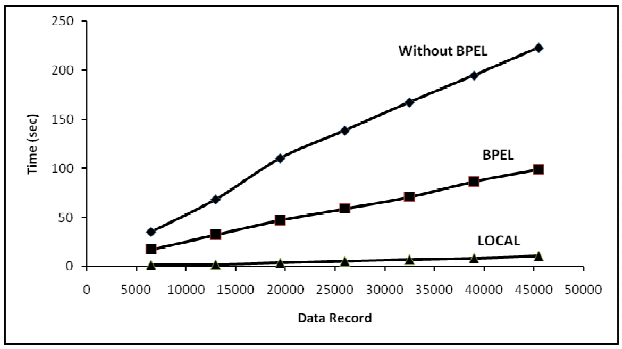

Fig. 9 Execution Time to Display National Population Data

In Figure 9 the execution time of BPEL compared with that of LOCAL result in a very significant difference, because the response given by the model of service was in the form of non-single output parameters meaning the form was a single array as a collection of the entire record of data and therefore more resources were required. The comparison between using BPEL and without BPEL was that they both require more resources. BPEL was faster because it did not connect directly to the database, but it accessed using a services-based text documents with XML format.

9. Conclusions

This paper aims to create EAF for interoperability and data integration for E-Gov applications in Indonesia based on real time architecture developed by the layer services. The aspects of real time are done by providing a trigger event to the other services in asynchronous message delivery model. The orchestration services are performed by GSB which is defined by BPEL. Based on the results of the evaluation by measuring the execution time of the services model in Table 3 and 4, the services model should be defined in EAF which is in the form of a model that has a response services with a single output parameter, such as the service recap of population, births, and deaths. The services model that has response to non-single output parameter should not be used on EAF. Rather, it uses local database (from each province and national) that records data obtained from services add, delete, and update which were requested by the district/city to produce a distributed database. The distributed database scheme aims to provide flexibility in each district/city, province, and national levels to meet different needs. The resources needed for interoperability by EAF is relatively less compared to the E-Gov application architecture where connections are made directly to the database, because EAF was built using a model services based on text documents with XML format.

Acknowledgments

The authors of this paper would like to thank Department of Computer Science and Electronics at Gadjah Mada University which provides guideline and technical assistance for the research.

References

[1] Patricia, J., Pascual, e-Government, e-Asean Task Force UNDP- APDIP, 2003.
[2] Hazlett, S., A., and Hill, F., E-government: the realities of using IT to transform the public sector, Managing Service Quality, EJEG, 2003, Vol. 13, No. 6, pp. 445-452.
[3] Sosiawan, E.,A., “Tantangan Dan Hambatan Dalam Implementasi E-Government Di Indonesia (Challenges and Obstacles in Implementing E-Government in Indonesia)”, National Proceeding e-Indonesia, ITB, Bandung, 2008, pp. 46-59.
[4] Hasibuan, Z. A., “Langkah-langkah Strategis dan Taktis Pengembangan E-Government untuk Pemda (Strategy Steps and Tactical Development of E-Government for Local Government)”, Information System Journal, MTI UI, Vol. 3 – No. 1, 2007, pp. 61-70.
[5] Setiadi, H., Hasibuan, Z. A., and Fahmi, H., “Perubahan Arsitektur Database dan Aplikasi Administrasi Kependudukan Yang Sejalan Dengan Otonomi Daerah (Changing Database Architecture and Population Administration Application toward Local Autonomy)”, Information System Journal, MTI UI, Vol. 3 – No. 1, 2007, pp. 42-51.
[6] Instruksi Presiden Nomor 3 Tahun 2003 tentang Kebijakan dan Strategi Nasional Pengembangan e-Government (Presidential Instruction No. 3 of 2003 on National Policy and Strategy Development of e-Government), Jakarta, 2003.
[7] Blue Print Sistem Aplikasi E-Government, Departemen Komunikasi dan Informatika Republik Indonesia (Blue Print System Applications E-Government, Ministry of Communications and Information of the Republic of Indonesia), Jakarta, 2004.

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.

[8] Rabaiah, Abdelbaset, "Federation of E-government: A Model and Framework", Information Systems Audit and Control Association Journal, Vol. 4, 2007, pp. 1-4.
[9] Allen, B.A., Juillet, L. Paquet, P. and Roy, J., "E-Governance & government on-line in Canada:Partnerships, people & prospects", Government Information Quarterly, 2001, Vol. 18, pp. 93–104.
[10] Janssen, Marijn, and Cresswell, Anthony, "Enterprise Architecture Integration in E-government", Journal of Enterprise Information Management, Vol. 18, Issue 5, 2005, pp.531 - 547.
[11] Ioannidis, Anastasios, Spanoudakis, Manos, and Priggouris, Giannis, "An Xml-Based Platform For E-Government Services Deployment", Journal of Web Engineering, Vol. 1, No.2, 2003, pp. 147-162.
[12] Lee, Thomas, Hon , C., T., and Cheung, David, "XML Schema Design and Management for e-Government Data Interoperability", Electronic Journal of e-Government, Vol. 7 Issue 4, 2009, pp. 381 – 390.
[13] Phu, Phung Huu, and Yi, Myeongjae, "A Service Management Framework for SOA-based Interoperability Transactions", Proceedings The 9th Russian-Korean International Symposium on, Korean, 2005, pp. 680-684.
[14] Contenti, Mariangela, "A Distributed Architecture for Supporting e-Government Cooperative Processes", TCGOV, 2005, pp. 181–192.
[15] Weerakkody, V., Janssen, M., and Hjort-Madsen, Kristian, "Integration and Enterprise architecture challenges in E-Government: a European perspective", International Journal of Cases on Electronic Commerce, Vol. 3, Issue 2, 2007, pp. 13-36 , April-June.
[16] Müller, Viktor, Simon, Balázs, László, Zoltán, and Goldschmidt, Balázs, "Business Monitoring Solution for an e-Government Framework based on SOA Architecture", Proceeding of the 13th IASTED International Conference : Software Engineering and Applications (SEA), Cambridge, MA, USA, 2009, November 2-4.
[17] Peristeras, Vassilios, Tarabanis, Konstantinos, and Goudos, Sotirios K., "Model-driven eGovernment interoperability: A review of the state of the art", Computer Standards & Interfaces Journal, Science Direct, Vol 31, 2009, pp. 613–628.
[18] Stefanova, Kamelia, "Design Principles of Identity Management Architecture Development for Cross-Border eGovernment Services", Journal of e- Government, Vol. 8, Issue 2, 2010, pp. 189-202.
[19] Braun, Christian, and Winter, Robert, "Integration of IT Service Management into Enterprise Architecture", Proceedings of the 2007 ACM symposium on Applied computing, ACM New York, NY, USA, 2007, pp. 1215 – 1219.
[20] Erl, Thomas, Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall PTR, New Jersey, 2005.
[21] Open SOA, Service Component Architecture (SCA), http://osoa.org/display/Main/Service+Component+Architecture+Home, 2007. (Last access: Juli 3, 2009).
[22] Michelson, B.,M., Event-Driven Architecture Overview, Patricia Seybold Group, OMG, February 2, pp. 1-8, 2006.
[23] Sahin, K., "Service Oriented Architecture (SOA) Based Web Services For Geographic Information Systems", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Beijing, 2008, Vol. XXXVII, Part B2, pp. 625-630.
[24] Zhong D., et al., "Reliability Prediction for BPEL-Based Composite Web Service", Proceedings of The First International Conference on Research Challenges in Information Science, Ouarzazate, Morocco, 2007, pp. 265-270.
[25] Hafid, B., Balouki, Y., Laassiri, J., Benaini, R., Bouhadadi, M., and Hajji, S. E., "Using BPEL for Behavioral Concepts in ODP Computational Language", Proceedings of the World Congress on Engineering, London, U.K., 2009, Vol. I, pp. 123-129.
[26] Juric, M., B., and Pant, K., "WSDL and UDDI Extensions for Version Support in Web Services", Journal of Systems and Software, Vol. 82, 2009, pp. 1326-1343.

Aris Puji Widodo is a PhD student at Department of Computer Science and Electronics, Gadjah Mada University. His subject is on aspect software engineering architecture.

Jazi Eko Istiyanto is a professor at Department of Computer Science and Electronics, Gadjah Mada University. He holds a PhD in Electronics Systems Engineering, University of Essex, UK since 1995. He is a member of IEEE Robotics and Automation Socienty, IEEE Computer Socienty and Institution of Engineering and Technology.

Retantyo Wardoyo is an associate professor at Department of Computer Science and Electronics, Gadjah Mada University. He holds a PhD in Computation Technology, UMIST, UK since 1996.

Purwo Santoso is a professor at Department of Government Faculty Social and Political Sciences, Gadjah Mada University. He holds a PhD in Political Sciences, London School and Political Sciences, UK since 1998.

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.