E-Gov. Application Integration Design In Indonesia

Candra Taufik
Study Program Informatics Technique, Politeknik Piksi Ganesha Bandung
E-mail: **ctaufik@gmail.com**

Abstract

Decentralization gives authority to agencies / institutions / organizations in each province to plan the creation and development of applications tailored to their needs. Data related to the development is stored in each application of the organization which is periodically reported to the respective leader of government. The development data is very necessary for decision making in the next period development planning. Transparency and accountability are the absolute requirements for reporting development data, for that it is necessary to increase the speed and accuracy of data collection, so that progress in regional development can be monitored continuously and decision making can be carried out more quickly. Therefore, it is necessary to integrate applications from all applications found in agencies / institutions / organizations in each province.

Application integration may use various technology such as : SOAP, REST, SSO, etc. This research is aimed at studying alternative E-Gov application integration architecture that could be implemented in several government agencies / institutions / organizations. Qualitative methods are used in this study, namely by studying some related literatures to obtain alternative architectures and then create new architecture and design. With the existence of an alternative integration architectures, it is expected to facilitate the design of application integration in more detail.

Keywords— *application; architecture; design; E-Gov; integration; SOAP; REST*

1. INTRODUCTION

In Indonesia, using a decentralized system in government allows the agency / organization or more precisely SKPD to develop the applications needed to manage development according to their respective duties and functions.

The Regional Work Unit (usually abbreviated as SKPD) is a device of the Regional Government (Province and Regency / City) in Indonesia. SKPD is the executive function executive who must coordinate so that the administration of the government runs well (wikipedia.org). The legal basis applicable since 2004 for the establishment of SKPD is Article 120 of Law no. 32 of 2004 concerning Regional Government. Into the SKPD including the Regional Secretariat, Expert Staff, DPRD Secretariat, Service Offices, Agencies, Regional Inspectorates, other regional institutions that are directly responsible to the Regional Heads, Sub-Districts (or other equivalent units), and Kelurahan / Desa (or other units of the same level).

The laws and regulations that form the legal basis for the implementation of application integration in Indonesia are as follows: Minister of Home Affairs Regulation Number 13 of 2006 concerning Guidelines for Regional Financial Management as amended the second time by Minister of Home Affairs Regulation Number 21 of 2011; Government Regulation Number 39 of 2007 concerning State / Regional Financial Management; Minister of Home Affairs Regulation Number 54 of 2010 concerning Implementation of Government Regulation Number 8 of 2008 concerning Stages, Procedures for Preparation, Control and Evaluation of Implementation of Regional Development Plans; Law Number 23 Year 2014 concerning Regional Government as amended second time by Law Number 9 of 2015; and Regulation of the Minister of Communication and Information Technology Number 14 of 2016 concerning Guidelines for Nomenclature of Regional Devices in the Field of Communication and Information Technology.

The definition of Integration can be found in the excerpts from H. Panetto & J. Cecil (2013). Integration could be physical integration, application integration or business integration. According to laws and regulation in Indonesia mentioned above, kind of integration that

appropriate for government organization is application integration. It is the need of government because of good governance purpose.

Taking the sample of one of the provinces in Indonesia, namely West Java, there are several E-Gov applications, namely: SIPKD stands for Regional Financial Management Information Systems; RKPD Online stands for Regional Online Priority Work Plan; ATISISBADA stands for Information Technology Application for Regional Goods Cycle; E-Monev stands for Electronic Monitoring and Evaluation); E-SAKIP stands for Electronic Government Accountability Performance System).

Based on the legislation mentioned above the relationship between applications can be concluded in the form of a diagram. In the diagram, it is explained that there are 4 main stages in regional development: planning, budgeting, implementing and controlling.

Figure 1 is about Regional Development Business Process. The planning process is carried out by the RKPD with output in the form of the General Budget Policy (KUA), Priority Budget Ceiling Priority (PPAS), SKPD Renja. After the planning is complete, the next activity is budgeting whose role is held by SIPKD. The budgeting results are then carried out by organizations in the regional government and then followed by performance measurement activities (SAKIP) and asset management by the Regional Goods Cycle. Some activity data, indicators, budget realization data and activities are then obtained by E-Monev to be evaluated and produce an activity / program evaluation report.

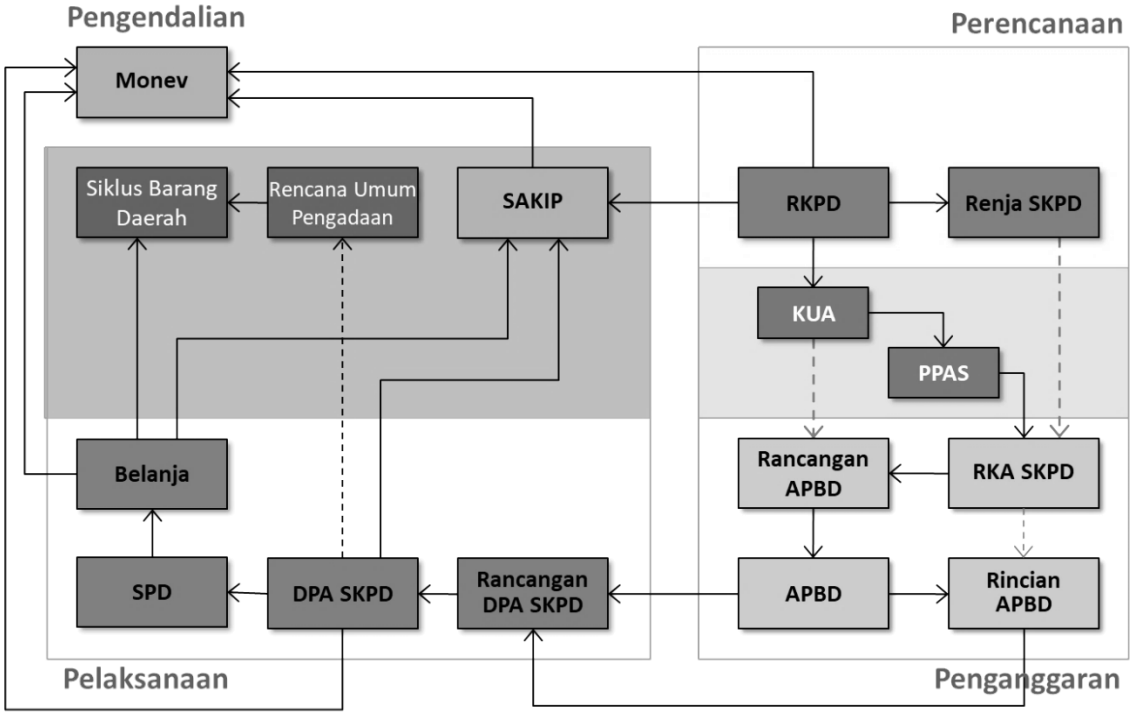

Figure 1. Regional Development Business Process (Torche Indonesia, 2017)

The result of this study are in the form of alternative architectural integration of E-Gov applications that can be used within the local government in Indonesia. The alternative architecture is derived from several architectures based on the results of a literature study.

2. RESEARCH METHODS

The research method used is a qualitative method, which starts with a literature study to obtain several examples of application integration architecture, then proceed with collecting data problems related to application integration in Indonesia, discussing examples of application integration architecture and designing new application integration architecture, as well as conclusion

2.1. Literature Study

Izza et.al. (2005) convey the problems related to semantic on Enterprise Application Integration (EAI). In computer science, semantics refers to the meaning of language, as opposed to their form (syntax). Izza et.al. (2005) proposed an Ontology-based architecture for dealing with semantic problems . Figure 2 is a global view of ontology driven service oriented architecture that related to semantic on EAI.

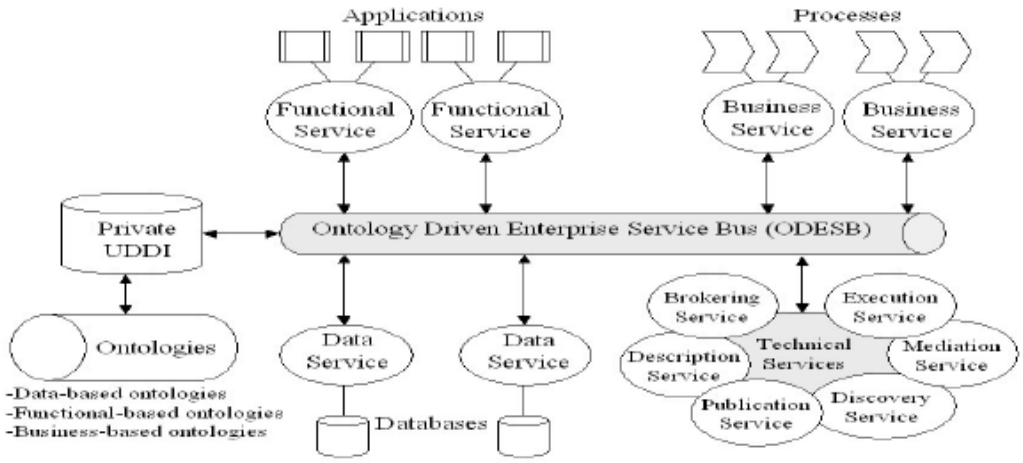

Figure 2. Global View of ODSOA Architecture (Izza et.al., 2005)

The next integration application is Single sign on (SSO) that we can see in Figure 3. SSO authenticates users to access all authorized applications to access. This eliminates the request for authenticaton again when the user changes the application during the session (Aaslund et al, 2007).

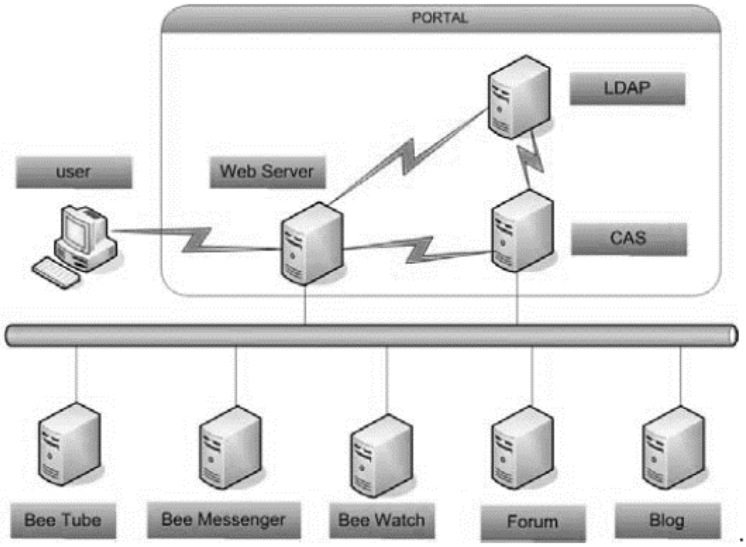

Figure 3. Single Sign On Architecture (Rudy et.al, 2009)

The use of Single Sign On also helps in organizing Users because of the use of Lightweight Data Access Protocol (LDAP) as a single user data. (Rudy et.al., 2009). Izza (2009) mentioned that in application integration there are a lot of applications which might be distributed applications.

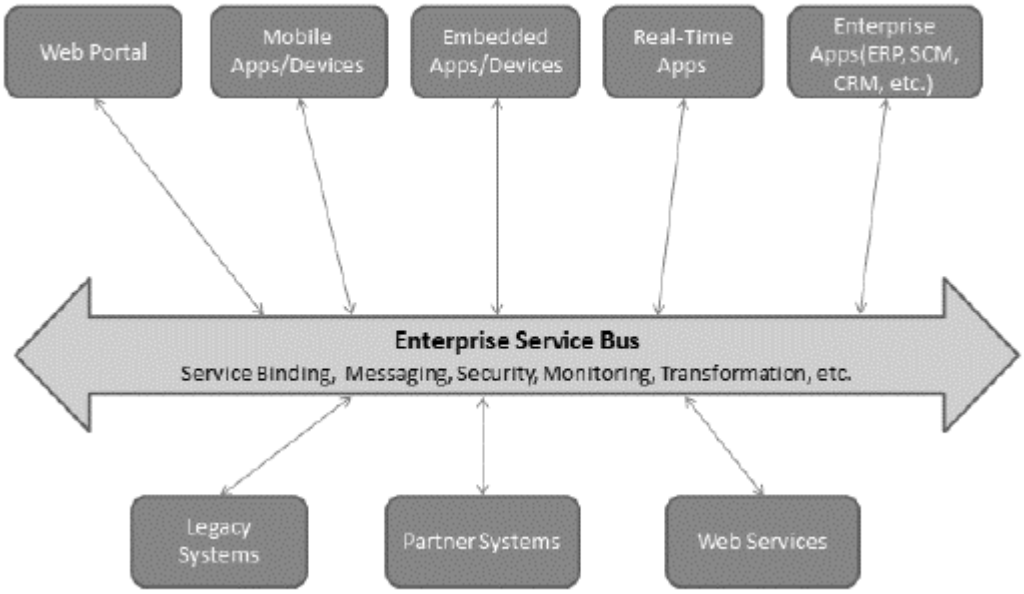

Figure 4. A SOA-oriented integration environment using ESB (Wu He & Li Da Xu, 2012).

Next is the Enterprise Service Bus (ESB), which is a collection of services provided to integrate various services from various applications, including service binding, messaging, security, monitoring, transformation etc. This can we see in Figure 4 about a SOA – oriented integration application using ESB. Wu He & Li Da Xu (2012) convey the importance of integrating distributed applications because they strongly influence the ability to compete. Xu, Bay & Wang (2010) explained that ESB can interact with various protocols such as CORBA, Java etc.

Istiyanto & Sutanta (2012) design a model of interoperability between applications. The design of the interoperability model between E-Gov applications that uses a web services architecture model using the REST method, consists of three designs, namely provider design, agent / broker design and requester design. Figure 5 shows to us about implementation of the interoperability model design between E-Gov 1 and E-Gov 2.

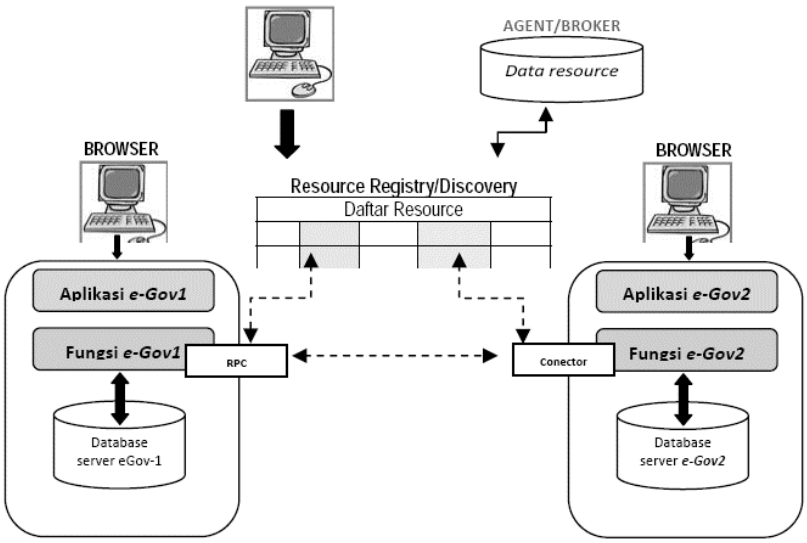

Figure 5. Implementation of the interoperability model design between E-Gov1 and E-Gov2 (Istiyanto & Sutanta, 2012)

Tihomirovs, J., & Grabis, J. (2016) describe the selection of Representational State Transfer Protocol (REST) or Simple Object Access Protocol (SOAP) as web service technology. REST will be the right choice for two simple information systems and SOAP is the right choice for the complex system and the need of additional security level.

Garcia & Abilio (2017) made an experiment to compare REST and SOAP technologies which resulted in REST being 0.1 seconds faster than SOAP and REST using bandwidth 73%

lower than SOAP. Below is the architecture used in the REST and SOAP experiments. Figure 6 is an integration architecture designed by Garcia and Abilio that support providers and consumers.

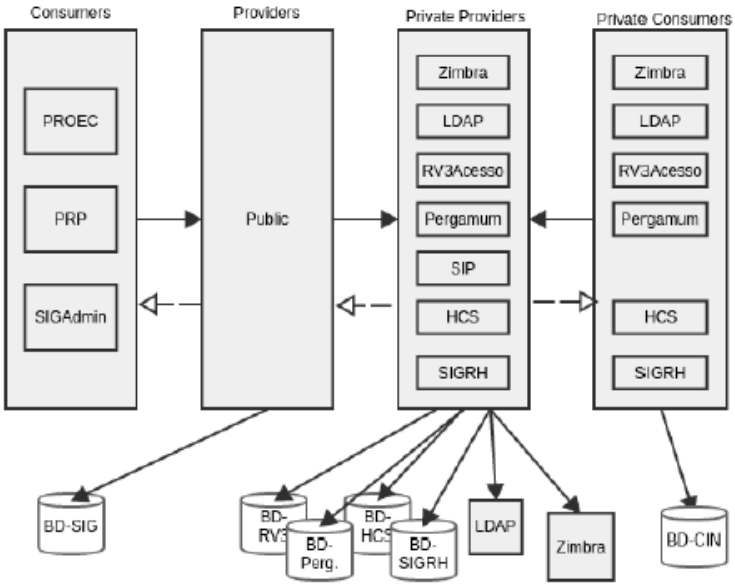

Figure 6. Integration architecture: providers and consumers (Garcia & Abilio, 2017)

Torche Indonesia, as an IT consulting company in West Java, has assisted in the development of the Interoperability System and E-Gov Application Integration in West Java. The architectural design can be seen in Figure above. Component of E-Gov Application Integration Architecture. In West Java include: Data Export, Stagging Schemes, Valid Schemes, Data Warehouse, ETL, GSB, IMM, Information Analysis and Dashboard Schemes. Figure 7 is application integration architecture designed by Torche Indonesia that implemented in West Java, Indonesia.

Data Export is a method for storing data from each application as a whole into a staging database. Estimates of data can be obtained in various formats and how to transfer data, for example there are those that are pulled from the backup server, some are via sql files or excel files.

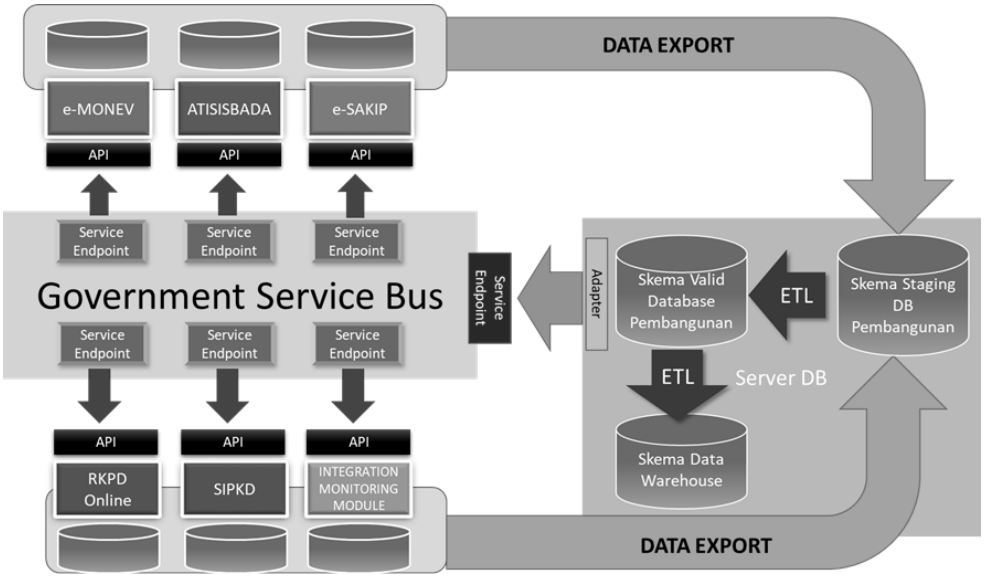

Figure 7. E-Gov Application Integration Architecture In West Java (Torche Indonesia, 2017)

Staging scheme is a database scheme that is prepared to hold data from each application, so that all data from the application is stored in the staging database.

Valid scheme is a database scheme that has been chosen so that there is no redundancy and data duplication. Valid schemes are also assisted by verification from each application manager.

The Data Warehouse Scheme is a database scheme that can later be used on the dashboard and self service analytic. The data warehouse scheme comes from a valid data scheme, because before the data is used by another application or service there needs to be validation first.

ETL, Extracting - Transfor - and Loading is the process of withdrawing data from one database to another database which is accompanied by the process of extraction, transformation and load. ETL is used to transfer data from a staging scheme to a valid scheme, and from a valid scheme to a data warehouse scheme.

Government Service Bus is a number of services provided for all applications that require data from OPD or other applications. The GSB here is only used for one direction, that is, from Valid Schema Data for each application.

The Integration Monitoring Module is a module that is expected to be able to oversee the data transfer process, regulate the use of service / government service buses, manage users from GSB.

Information Analysis is one component that processes data warehouses into information needed by decision makers. Dashboards are used to visualize data from the data warehouse according to user needs.

2.2. Problems of E-Gov. Application Integration

In the E-Gov application integration activities in Indonesia it is necessary to consider several problems that might be found in each region. For West Java Province there are at least 4 things that become problems, namely: E-Gov application can have the same data as other E-Govs; Data from one E-Gov application is required by another E-Gov application, the E-Gov application platform varies and Time synchronization / data provision needs to be agreed upon.

The similarity of data in several E-Gov applications can cause data redundancy so that there is a need for a process of selecting data to be used or used as a reference by users. Another possibility is the existence of data that contains the same meaning but is different in writing due to the use of different symbols or abbreviations.

It is possible that data from one E-Gov will be used by another E-Gov, so it is necessary to have a valid data reference and prioritize its use. Integration architecture needs to facilitate this, in the sense that the data that has been integrated can be used again by each E-Gov application.

The E-Gov application found in each SKPD is very diverse, so it is necessary to provide a kind of adapter or cross-platform integration method.

In carrying out its duties, each SKPD is very limited by time. Therefore, the provision of data or data synchronization is very limited and needs proper coordination. Besides that, it is necessary to have enough storage to hold data from the SKPD whenever the SKPD has data to be integrated.

The Regional Head as a user of the application integration system is an actor who will receive development reports compiled from integrated data. Reports are expected to be accessible anytime and anywhere. Therefore it is necessary to have storage that holds valid data or development resume data.

3. RESULTS AND DISCUSSION

Single Sign On (SSO) architecture has its disadvantages, because we can see from the architecture that the user can only enter into each application, obtain data from the application without the entire resume. Besides that if one of the applications is off, then the data on the application will not be visible.

The ESB architecture is very suitable and simpler and can even relate to various application platforms. The requirements that may be needed for an architecture like this are the need for certainty that every application connected with ESB is always online. In addition, additional applications are needed that retrieve data and create resumes and monitor data from other applications and store them in different storage / databases.

The architecture implemented by Istiyanto & Sutanta (2012) allows for interoperability between applications. This allows integration between applications, but cannot minimize redundancy and cannot generate development data resumes. Certainly, there is no storage that stores data sets from each application. Besides that, every E-Gov application will need to contact other applications as needed, so there is a possibility that each application will contact other E-Gov applications several times. The possibility of architecture can be modified for use according to the results of previous analyzes.

Ontology-based architecture designed by Izza (2005) can be used as an alternative with the terms of several additions as proposed in the review of ESB. This technology will be able to improve accuracy in data management, but needs to be equipped with Ontology-based technology.

Garcia & Abilio (2017) have an integrated architectural design which is another form of ESB, therefore it can be a good alternative with terms of several additions as proposed in the previous ESB review.

Torche Indonesia (2017) designed an integrated architecture that has Service Bus and is equipped with ETL, Staging Database and Valid Database. In the context of this integration, Torche Indonesia uses a combination of more data transfer technology. The problem that might arise is the ETL process from Staging to Valid, here we need accurate data analysis and support from several E-Gov application owners. More resources (Storage / DB) are used, because the Staging Database stores all data from each E-Gov application.

The first alternative is the Enterprise Service Bus Architecture which is modified by adding the Data Collector and Valid Data Provider application. With this architecture it is necessary to assume that the ability to transfer data through ESB has a high bandwidth. Figure 8 shows the Enterprise Service Bus Architecture using SOAP that simplified from figure 4.

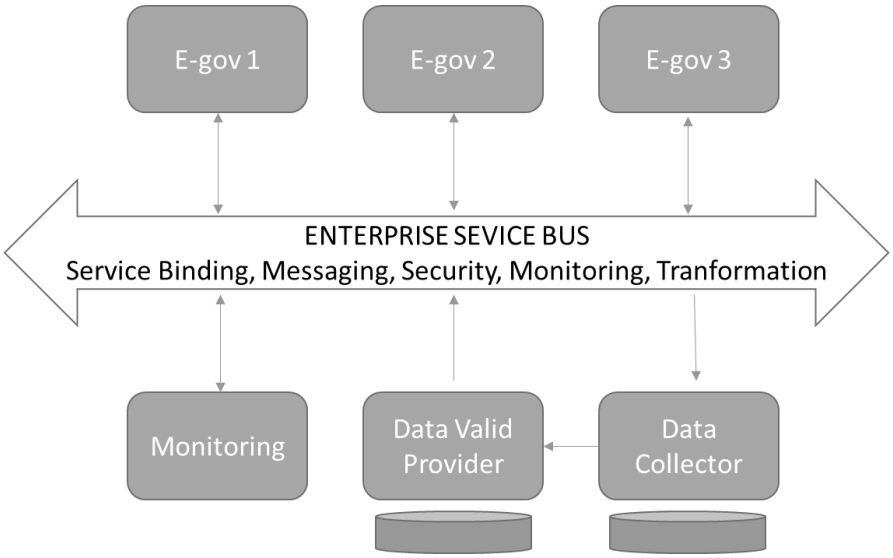

Figure 8. Enterprise Service Bus Architecture Using SOAP(Taufik, 2018)

ESB should provide a rule that every e-gov shoud transfer its data to Data Collector and take the data needed from Data Valid Provider. This rule should be communicated to all E-Gov Owners. Data Valid Provider and Data Collector must provide service to Monitoring module that collect information about the use of services and data that transferred by every e-gov.

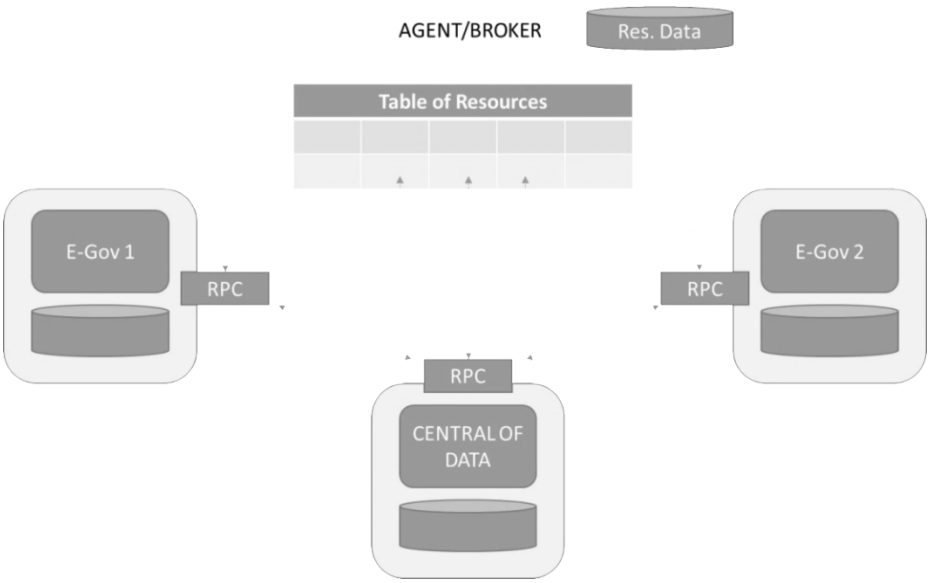

Figure 9. REST Application Integration Architecture (Taufik, 2018)

The REST architecture is the second alternative, with advantages in terms of data transfer speeds higher than SOAP. In this architecture, agents / brokers are available and each E-Gov application acts as the provider and consumer of the web service. Data interaction is only done between E-Gov and the data center, not between E-Gov and other E-Gov.

Figure 9 shows the REST Application Integration Architecture, a modification of figure 5. This figure shows us how to integrate two e-gov, but it is possible to integrate more than 2 e-gov. We can add more resource in the table of resource and add more RPC infront of new e-gov.

REST Application Integration doesn't transfer data from every e-gov to a data collector, but every e-gov may request service from Central Data. Central Data may collect data from every e-gov, validate data and create service resource to be accessed by every e-gov.

The third alternative is integration architecture that uses a combination of technologies such as those designed by Torche Indonesia (2017). Figure 10 is about Combined Application Integration Architecture that simplified from Figure 7. This architectural design has greater flexibility, but needs to be supported by adequate resources.

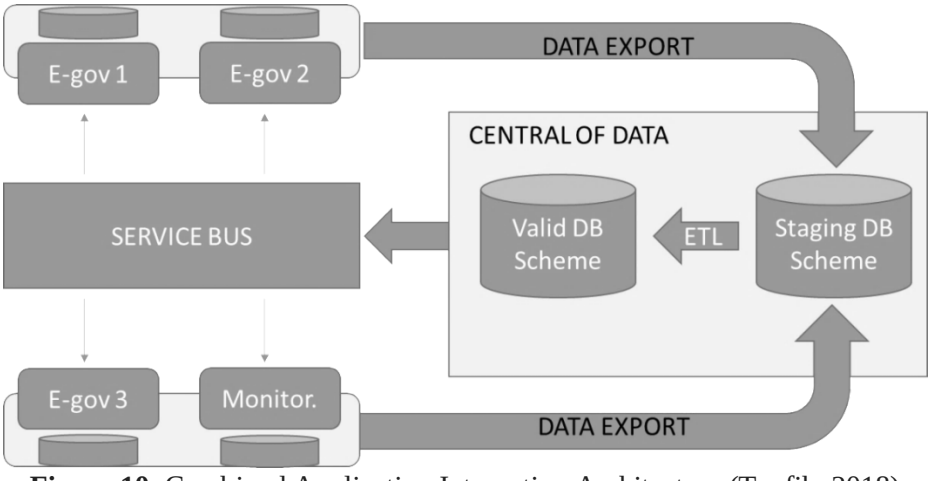

Figure 10. Combined Application Integration Architecture (Taufik, 2018)

In the combination application integration architecture, data sending triggers (data export) originate from the data center side because the Data Center directly accesses the database of each E-Gov application, while in ESB and REST, triggers the sending of data to the data center from each application E- Gov.

Detailed design is based on Combined Application Integration Architecture because of flexibility and is suitable for application integration of E-gov in Indonesia, according to Torche Indonesia (2017).

The Monitoring Module has 4 classes namely GSBMonitor to retrieve log data and display activity logs, UserMgt to manage users, Notification to retrieve status and provide status notifications, and IntegratedMonModule as a control class. Figure 11 shows the Integration Monitoring Module Class Diagram.

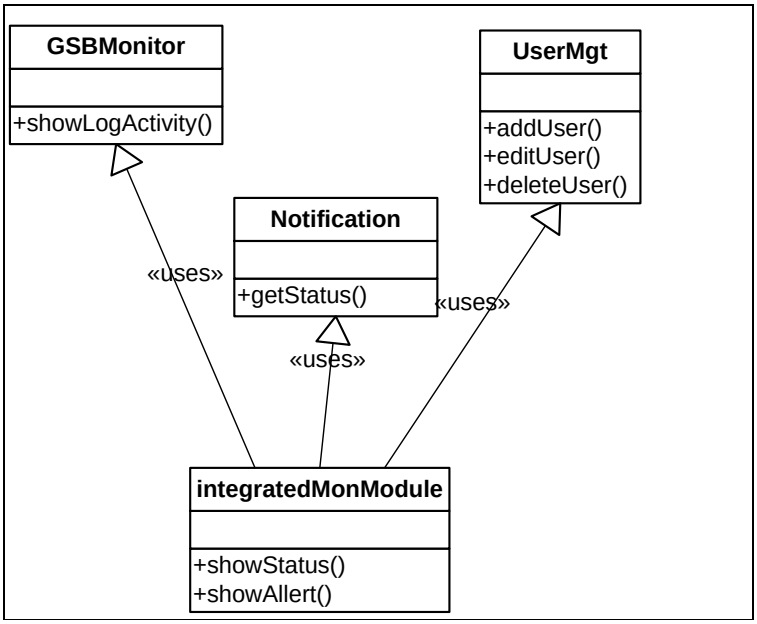

Figure 11. Integration Monitoring Module Class Diagram(Torche Indonesia, 2017)

Figure 12 is Government Service Bus Class Diagram that has 13 classes : RKPDCommonGoal, RKPDSupervision, RKPDField, RKPDOPD, RKPDProgram, RKPD Activity, RKPDExpend, SIPKDActivity, SIPKDBudget, SIPKDOutcome, SIPKDIndicator, SIPKDRealization, and ATISISBADA.

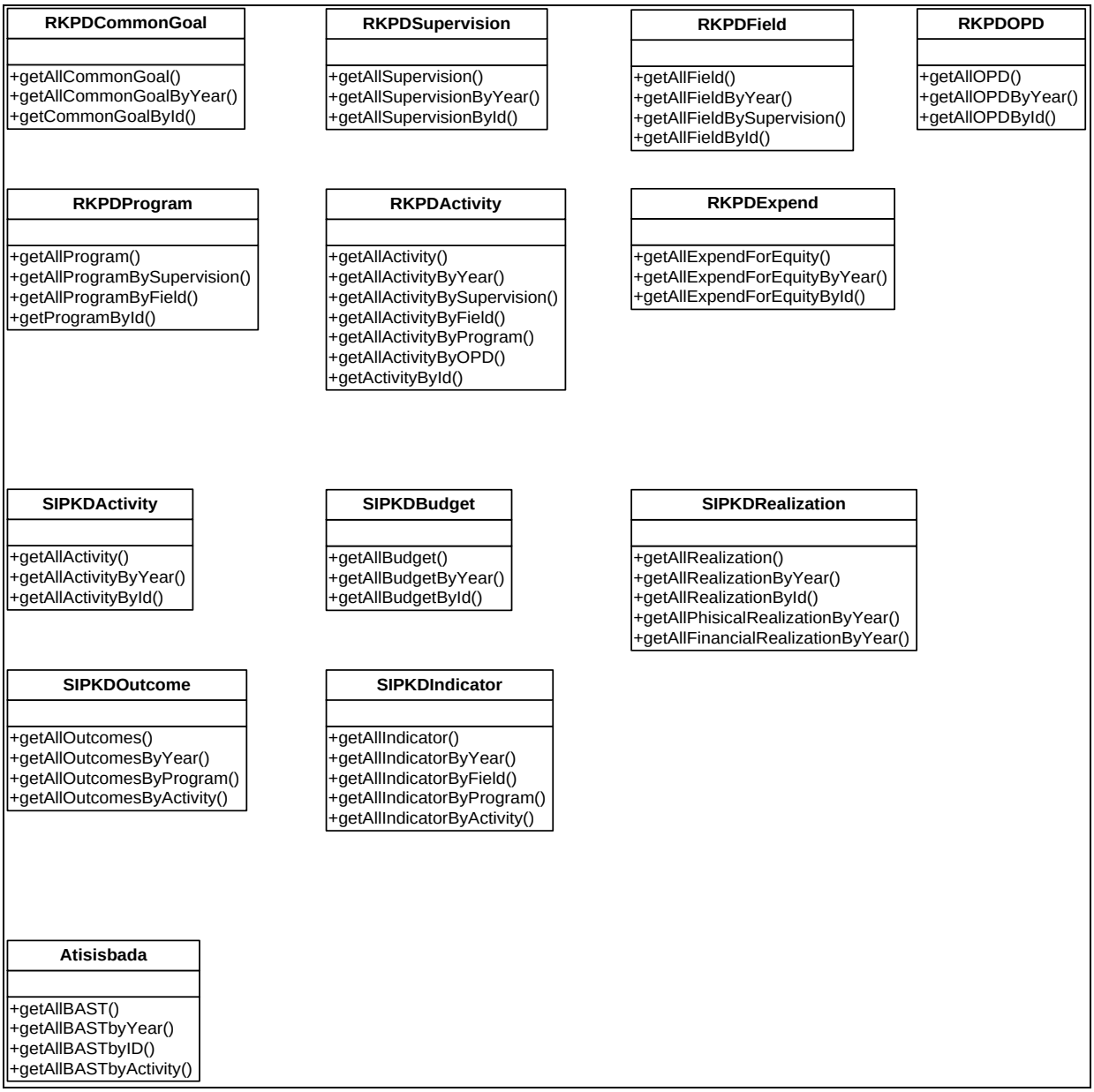

Figure 12. Government Service Bus Class Diagram (Torche Indonesia, 2017)

Figure 13 is One sample of Staging Data Design that should used by every E-gov while transferring data from its data storage to staging data storage. Every E-gov has specific database scheme, thus this staging data design should be adapted with database scheme in every E-Gov. For flexibility, the all database e-gov should be transferred to staging data storage or create a copy of e-gov database in staging data storage.

The impact of this activity is staging data storage should has huge data capacity to store all data from every e-gov. For maintaining story of data, data transferred from one period of time should not be merged with the new one. It should be kept until several years of operations to keep the history of data transfer. When something wrong happened, it sould be traced from last transferred data.

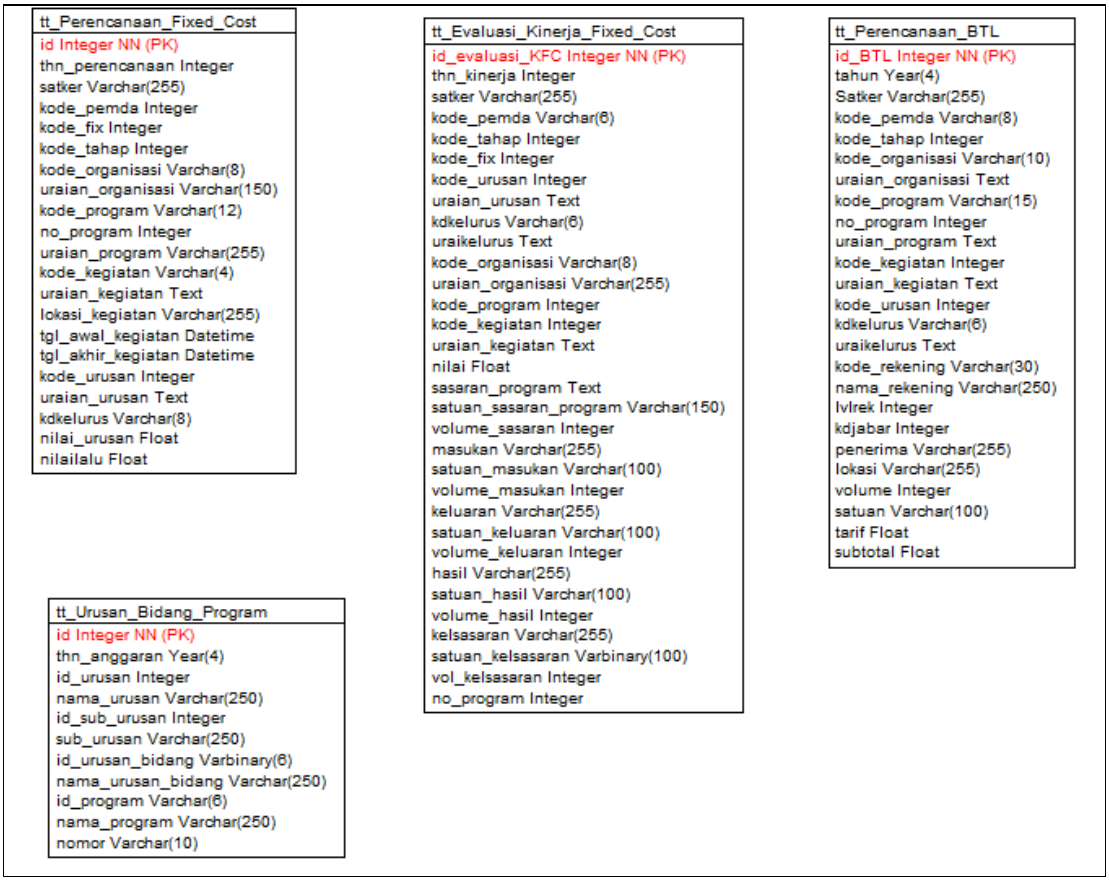

Figure 13. One of Staging Data Design (Torche Indonesia, 2017)

Valid data design in Figure 14 shows to us that collecting data from E-Gov is not simple. Some data needed has not collected during short period of project implementation, but it may be a figure of what is needed in E-Gov according to business process that implemented in Indonesia. This valid data design is not represent all valid data design in Indonesia, but could be used as reference for designing application integration.

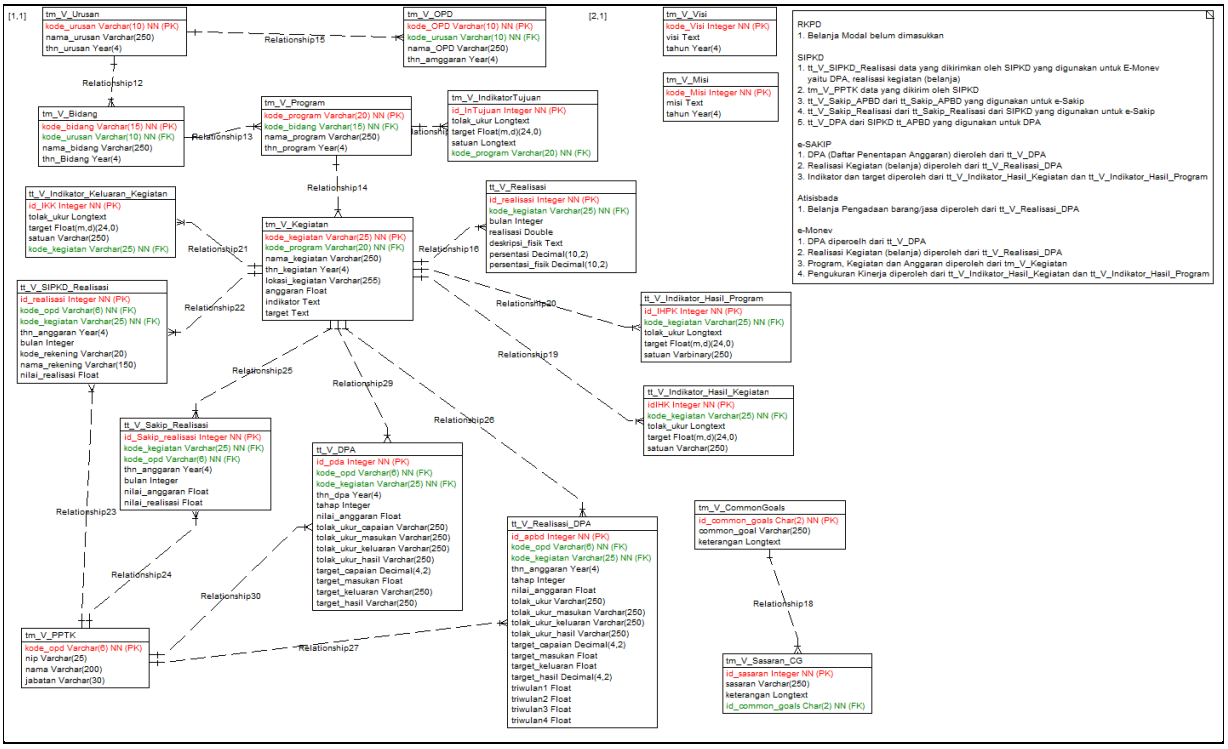

Figure 14. Valid Data Design (Torche Indonesia, 2017)

There are 19 tables in valid data design, and this may increase according to the needs of each e-gov because only 5 e-gov are integrated. Data storage needs will increase if the number of e-govs it integrates increases.

Data from a valid database could be reused by each integrated e-gov according to their individual needs. With this integration, it is expected that there is a uniformity of master data in each e-gov. With the uniformity of the data, it will be easy for the government to collect development data that will be displayed as a representation of plans, implementation and realization of development.

4. CONCLUSION

Application Integration is a requirement in industry and government, because with application integration will be able to produce faster decisions so as to increase the ability to compete or improve organizational performance.

Integration Architecture Applications that might be adopted in government are Enterprise Service Bus (ESB) designed by Wu He & Li Da Xu (2012), REST Architecture designed by Istiyanto, JE, & Sutanta, E. (2012) and Combined Integration Architecture Technology designed by Torche Indonesia (2017).

Combined Integration Architecture has flexibility in use but need support of more resources. This architecture is suitable with e-gov business process and could be a reference in activity of application integration in Indonesia.

5. SUGGESTED

For further research on the Architecture of E-Gov Application Integration in Indonesia, research is recommended on the use of Ontology Driven Service Oriented Architecture (ODSOA) as a comparison material for integration technology that has been implemented.

REFERENCES

[1] Aaslund, K., Larsen, S. (2007). OTS-Wiki: A Web Community for Fostering Evaluation and Selection of Off-The-Shelf Software Components. Department of Computer and Information Science. Norwegian University of Science and Technology (NTNU)
[2] Ahmadjayadi, C., 2006, Standarisasi Menuju Interoperabilitas e- Government, makalah keynote speech pada Workshop Standarisasi Menuju Interoperabilitas E-Government, Jakarta.
[3] Garcia, C. M., & Abilio, R. (2017). Systems integration using web services, rest and soap: a practical report. Revista de Sistemas de Informação da FSMA, 1(19), 34-41.
[4] He, W., & Da Xu, L. (2012). Integration of distributed enterprise applications: A survey. IEEE Transactions on industrial informatics, 10(1), 35-42.
[5] Istiyanto, J. E., & Sutanta, E. (2012). Model Interoperabilitas Antar Aplikasi E-Government. Jurnal Teknologi Technoscientia, 4(2), 137-148...
[6] Izza, S. (2009). Integration of industrial information systems: from syntactic to semantic integration approaches. Enterprise Information Systems, 3(1), 1-57.
[7] Izza, S., Vincent, L., & Burlat, P. (2005, February). Ontology-Based Approach for Application Integration. In First International Conference on Interoperability of Enterprise Software and Applications–Interop-ESA (Vol. 5).
[8] Panetto, H., & Cecil, J. (2013). Information systems for enterprise integration, interoperability and networking: theory and applications.
[9] R. Xu, J. Bai, and Y. Wang, “The research and implementation of power application system integration based on enterprise service bus,” 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), Beijing, China, 2010.
[10] Rudy, R., & Gunadi, O. (2009). INTEGRASI APLIKASI MENGGUNAKAN SINGLE SIGN ON BERBASISKAN LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL (LDAP) DALAM PORTAL BINUS@CCESS (BEE-PORTAL). Portal, 7.
[11] Taufik, C. (2018). Arsitektur Integrasi Aplikasi di Indonesia. JURNAL ELEKTRO-KOMPUTER-TEKNIK 1 (1), 28-39
[12] Tihomirovs, J., & Grabis, J. (2016). Comparison of soap and rest based web services using software evaluation metrics. Information Technology and Management Science, 19(1), 92-97.