Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Fundamental Architectures of Datacenter ECR System Prototype by Applying Principle of Messaging Services

To cite this article: K Y E Aryanto *et al* 2019 *J. Phys.: Conf. Ser.* **1165** 012008

IOP Conf. Series: Journal of Physics: Conf. Series 1165 (2019) 012008 doi:10.1088/1742-6596/1165/1/012008

Fundamental Architectures of Datacenter ECR System Prototype by Applying Principle of Messaging Services

K Y E Aryanto[1], K T Dermawan[2] and I M Putrama[3]
[1,2,3] Ganesha University of Education

E-mail : yota.ernanda@undiksha.ac.id[1], teguh.dermawan@undiksha.ac.id[2], made.putrama@undiksha.ac.id[3]

Abstract. Interoperability is a component that contains information systems integration. It provides an interface from multiple data sources to support data and information integrity between various information systems. This study aims to: (1) Propose Architecture of Messaging services to support data integration of Datacenter ECR. (2) Present a fundamental architecture prototype Datacenter ECR with messaging services principle. This research is expected to provide the solution to the data integration architecture between systems that is running on HTTP protocol over internet. The integration model uses messaging services schemes and improve it with customization end-to-end encryption messages. This type of research is research and development (Research and Development) with Agile SCRUM Framework research model. It will provide any robust workflows to each step of development and management resources during design architecture and develop Datacenter ECR prototype. For the testing process, three (3) test process stages are performed: (1) White-box and (2) Black-box testing (3) Performance consistency testing

1. Introduction

Universitas Pendidikan Ganesha is a university responsible to the Ministry of Research Technology and Higher Education (Kemenristek dikti), located at Jalan Udanaya No, 11, Singaraja, Bali Province. Today, Universitas Pendidikan Ganesha (UNDIKSHA) has ICT Department called UPT TIK UNDIKSHA. UPT TIK with hands directly provides any support in IT Infrastructure and any solution by implementing various information system to support all operational activity in UNDIKSHA. UPT TIK UNDIKSHA has wrapped all information systems into one integration system. It is called E-Ganesha. E-Ganesha is platform that contains any of information system in website and mobile platform, which covers systems running at UNDIKSHA such as Lectures Database, Employees Database, KINERJA, SKP, HTL, Information Academic Systems, SSO (Single Sign ON) and HELPDESK.

Many information systems have been developed by UPT TIK Universitas Pendidikan Ganesha to support operational effectiveness in the working environment of Universitas Pendidikan Ganesha. The problem is quite crucial related to the establishment of various information systems support, most of the information system does not have data synchronization so that there are many data discrepancies that complicate the process of extracting the data digest for the purposes of accreditation forms and statistical analysis [9]. Based on these cases, it is necessary to develop a schema integration system data warehouse. In 2017, there are research proposing a solution to the data integration system. That solution contains messaging services architecture with asynchronous schema data integration, which is called Datacenter ECR (ETL, Crypotpost, and Rest Services). On that system, Technical Services provide schema Extraction, Trans-form and Loading and RESTFul API with end to end encryption to ensure data security over the air. End to end encryption guarantees of encryption process uses SSL on alternative port 443, it handle problems that may arise when port 443 is in the maintenance phase, such as SSL renewal, or expired SSL which will allow the entire authentication process and the contents of the data to be unsecured. Based on main requirements of data publishing, UPT TIK has to publish sensitive data, privacy and data security, in case that security must be maintained, so that requires data security mechanisms that do not only take advantage of SSL technology on port 443. This process can be addressed by developing an information publishing mechanism that is encrypted with open-ssl technology.

Encryption and Decryption are important parts in ensuring data security in the data transmission process. There are several algorithms that are often used in the process of decryption and encryption namely AES, DES and RSA. AES and DES belong to the Symmetric Algorithm, whereas RSA belongs to Asymmetric Algorithm. Based on initial studies research, the technology that we apply especially in integration of source systems into the data center system is to utilize ETL technology (Ex-traction, Tranform, Loading) and asynchronous data transmission by utilizing framework messaging products. An architecture that is used in Datacenter ECR prototype is based on messaging services system, which uses RabbitMQ framework to support it.

The problems of implementation RabbitMQ Framework in environments system at UPT TIK was technical architecture must be written and applied in that environment. UPT TIK has various information system that is still running on, with website and mobile apps platform. The technical architecture must be covering integration schema that will be used to communicate of each system. In that goal, all integration system can be provided with direct asynchronous data integration to ensure data stay tune on update without manually triggered.

Based on the problems, the development of fundamental architectural of a data center system (data center) using messaging services is needed to support the integration of data on information systems developed by UPT-TIK Universitas Pendidikan Ganesha. Improvisation that researchers do is applying *synchronous* and *asynchronous* data transmission technology with core messaging services and utilizing **ETL, Cryptopost** and **REST Service.**

This paper is structured as follows, Section 1 is the introduction, Section 2 provides a brief description about methodology research, Section 3 describes an overview and result the communication prototype DC System, and Section 4 draws conclusion.

2. Method

On the development datacentre system ECR, it needs technical design steps to ensure technology can be robust in any cases of data integration and security. To maintain that process on development, we use project management SCRUM. SCURM that we used in the development process of Data Center with E-CR has several sequence steps; those are like a guideline to make development process on the track of functional and non-functional specification. Several steps that we must follow to ensure the design fundamental architecture were design messaging services model, Systematic schema E2EE, Boundary system messaging services, data transmission message masking with E2EE.

2.1. Messaging Services Model Architectures

Datacenter ECR prototype is developed by applying fundamental architectures of messaging services. It is used to guarantee data integration between systems. The Data Center System Prototype is supported with a data communication architecture model in the form of asynchronous and synchronous. The asynchronous data communication model applies the concept of data communication in the form of topic and queue messages. In addition to the asynchronous communication model, the data communication model implemented is a synchronous model by utilizing data communication through the HTTP protocol in the form of a web service (in the form of a RESTful API). Scenario model is used for developing data prototypes with asynchronous and synchronous.

In this research, fundamental integration schema used is using basic of messaging concepts, which involved minimum two entities, called sender and receiver. Sender is entity to produce some messages, which will be sent into virtual pipe (in this case, we called it channel. Then, another entity that wants to retrieve those messages is called receiver.

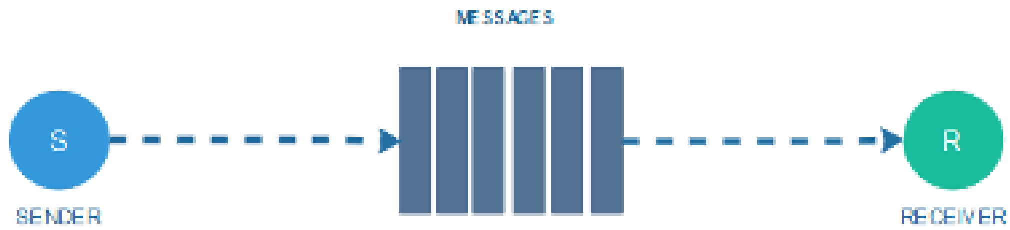

Figure 1 Messaging system architecture fundamental that will be used on this research. That contains three components namely; sender, receiver and messages. Sender is entity that produces messages and send it, and then receiver is entity that receives messages. Messages on this schema is means data formatted that is ready to send and receive

In this research, we ensure of messages must be formatted. Formatting messages is important in order to ensure the messages that have header and body by copyright sources. It means all of messages must be signed as original messages by valid sources, then all information in the messages must be covered to prevent data or information leak over the networks.

2.2. E2EE Schema during data coverage

E2EE is a data security concept in the form of a message with the End to End Encryption model. Encryption is a process of randomizing messages with forms that cannot be read. Reading a random message can only be done using only the secret key. End to end Encryption is a technique that is widely used to guarantee the confidentiality of data and information when transmitted through local networks or the internet.

The description of the application of End to end encryption on data security applies the encryption process by using the public key and the reading process using the private key. The encryption standard used is one of the AES (Advanced Encryption Standard) which consists of 3 chipper blocks, namely AES-128, AES-192, and AES-256. The algorithm developed to implement message encryption technology is RSA (Rivest Shamir Adleman). This algorithm involves encryption with exponential functions. A plaintext data is encrypted in blocks. Each block has a binary value that is less than a certain value (n). The following architectural design will be applied in the End to end Encryption process in this study:

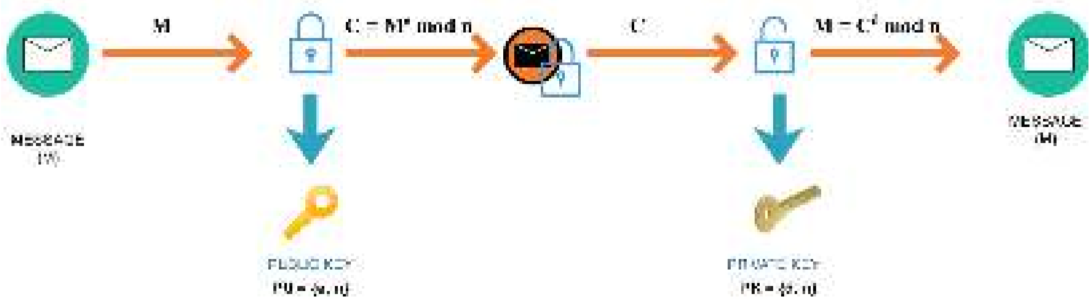

Figure 2 End to End Encryption Architectures that maintains of transport key between end point to ensure masking message and open messages that will be used in Datacenter ECR Prototyping.

Encryption process involves public key in the form of PU = {e, n}. So that message (M) encrypted with PU = {e, n} will follow the encryption formula.

$$C = M^e \bmod n$$

This encryption formula will result in a randomized binary sequence based on the use of a public key. A message (plaintext) that is scrambled next is called cipher text. The Decryption process involves a private key (private key) in the form of PK = {d, n}. Decrypt a cipher text (C) C with PK = {d, n} is using a series of decryption formulas

$$M = C^d \bmod n$$

The decryption stage will return a plaintext (M) message to be read again. Based on the series of formulas above, the system will produce public and private keys to support the encryption process. In this process of decryption, key must be required to open up wrapped messages by sender. Receiver must guarantee that the key still exists during retrieve data. Without key, those messages cannot be changed to plaint text. The plaint text of messages can be read without any permission of protocol. It means to open it, the receiver is not required to use specify protocol (such as HTTP, STOMP, etc), the receiver just ensures that key and devices still on safe and guarantee operating system on that devices can store the private key safely.

2.3. Boundary and Schema Messaging Services Systems

The Data Center System Prototype was developed with a data communication architecture model in the form of asynchronous and synchronous. The asynchronous data communication model applies the concept of data communication in the form of topic and queue messages. In addition to the asynchronous communication model, the data communication model implemented is a synchronous model by utilizing data communication through the HTTP protocol in the form of a web service (in the form of a RESTful API). Scenario model is used for developing data prototypes with asynchronous and synchronous.

During data communication in Datacenter E-CR, we are required to design principle core of messages services architecture. In this research we use entity *publisher* and *consumer.* Messages can be published by *publisher* and consumed by *consumer*. Messages (data) will be put into queue. Queue that ensures data messages can be retrieved by consumer, and *publisher* can put on it queue.

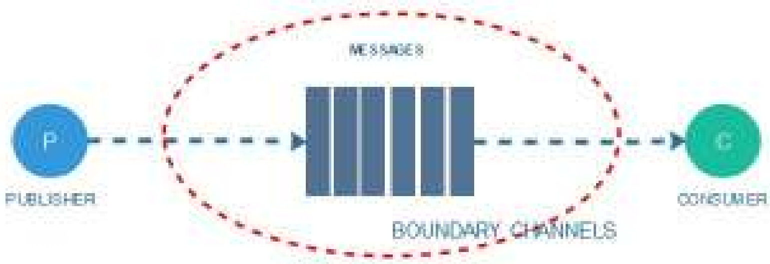

Figure 3 Core message schema by consumer and publisher. Publisher is entity who will produce messages (data and information). Then, consumer is entity who will consume messages (data and information). Messages are represented with open stack queue that is ready to put and retrieved to each other publisher or consumer

Boundary of messages environments in this schema is Channels. Channel is a virtual pipe that connects a sender to a receiver. In channel, it contains messages. A Message is an atomic packet of data that can be transmitted on a channel. Thus to transmit data, an application must break the data into one or more packets, wrap each packet as a message, and then send the message on a channel. Likewise, a receiver application receives a message and must extract the data from the message to process it. The message system will try repeatedly to deliver the message (e.g., transmit it from the sender to the receiver) until it succeeds. In the simplest case, the message system delivers a message

directly from the sender's computer to the receiver's computer. However, actions often need to be performed on the message after it is sent by its original sender but before it is received by its final receiver. For example, the message may have to be validated or transformed because the receiver expects a different message format than the sender. A Pipes and Filters architecture describes how multiple processing steps can be chained together using channels. To represent sender and receiver on these concepts, we use terminology of Endpoints. Endpoints is entity of the boundary system who will publish and consume those messages on the channels. In the architecture we call sender as publisher, and receiver as consumer.
Publisher will send data updates through on channel. Those messages will be put in queue messages on that channel. To retrieve any of queue messages, the consumer must join and connected into that channel. In this channel, we can implement messages posting by topic or queueing. Topics are messages that are posted and broadcasted to any endpoint that subscribe into those topics. Broadcasting through on channels must be set on mode durable messaging consumer. Durable consumer guarantees any of messages that will be received. In this case, we can handle of fail over messages retrieval by consumer.

2.4. Data transmission with messaging masking (by applying E2EE)
By combining method resources of points 2.1, 2.2, and to 2.3, this research creates fundamental architectures of Datacenter Systems Prototype by applying principle of messaging services. In the development phases, we handle data integration scenario to create integration system design with messaging services. In the scenario, it contains any activity related on data transmission between three various systems.

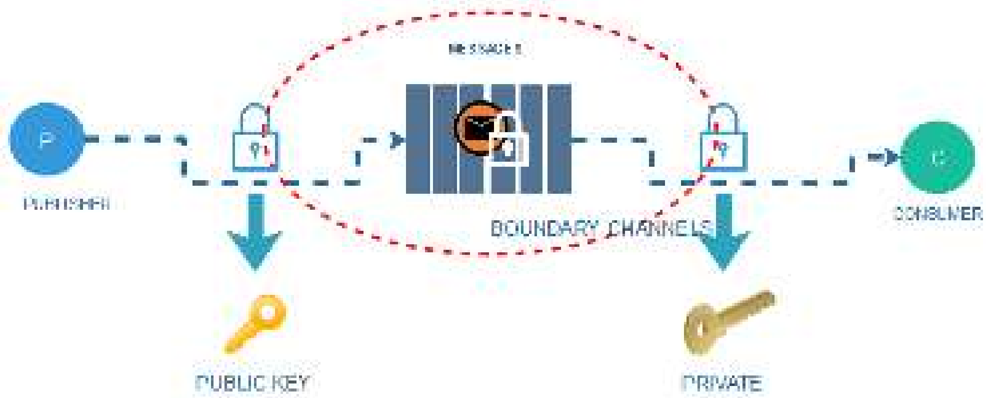

Figure 4 Architecture Design Data Transmission with messaging masking (Applying E2EE), by this architecture, in workflows of message, all message must be encrypted by publisher, then send it into channel. To encrypt, it is a must to have public key. Encrypted messages will be put into queue. Consumer will retrieve messages through those channels. To open messages, consumer must decrypt messages with private key. Ensure consume has right private to open those messages

In the workflow process of this architecture, all the end point (publisher or consumer) must require secret key to encrypt and decrypt messages. Messages will be encrypted using public key by publisher, and then messages can be decrypted by using private key. Private Key derivate by public key. On this schema we can sign off all messages through their header messages, to ensure content messages by publisher sources.

3. Results

Data transmission scenarios for data transmission simulations on the prototype datacenter system with E-CR involve three information system entities, namely information systems with the names Bob, Juvia and Alice. Bob, Juvia and Alice are information systems that have their own specialization to handle certain operations. Bob is an information system that supports the operation of the student system. Alice is an information system that handles the operational letters of SK and Letter of Assignment, while JUVIA is an operating system that will handle academic operations. Juvia, Bob and Alice will communicate with each other. In this case Bob will enter the student entry, then JUVIA and Alice must get the appropriate data that has been inputted by Bob. This transmission process involves DC entities to bridge data communication between Bob, Juvia and Alice. DC is the alias used by the E-CR Datacenter prototype.

In this research, communication model applies the principles of messaging services using RabbitMQ Server. The message form used to communicate data is QUEUE and TOPIC. BOB will put a message in QUEUE, then DC_SYS will retrieve the message from QUEUE, the message taken will then be published to TOPIC. All systems that subscribe with TOPIC automatically get every message in it. The communication path is made by configuring it in RabbitMQ Server, by default the connection with RabbitMQ Server runs on port 5672. In addition, it is implemented the asynchronous data communication scheme, the E-CR Datacenter system prototype applies a data communication model with a synchronous model by implementing HTTP protocols in the form of webservices. The architecture of this research was clearly representing boundary on each platform that can join to these architecture data transmissions with combine messaging services and End-to-end point encryption (E2EE).

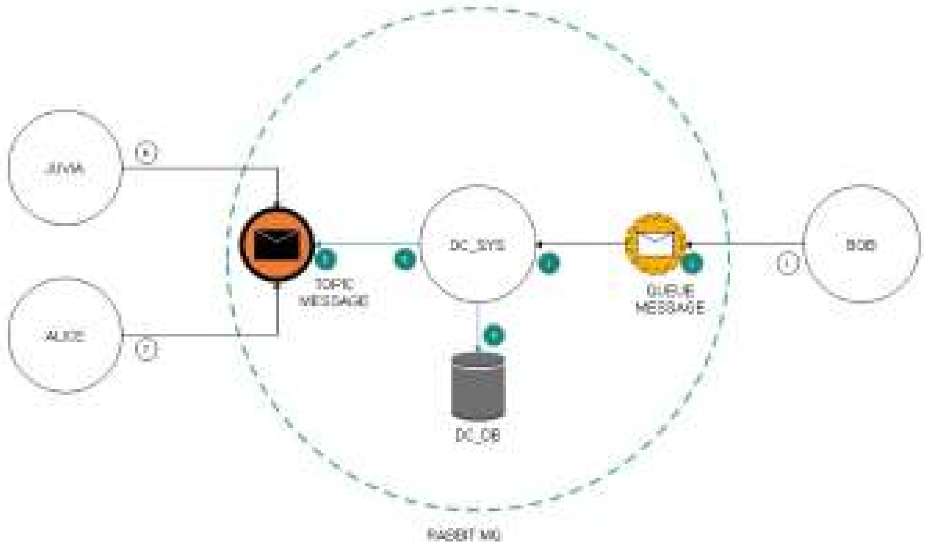

Figure 5 Fundamental architectures of Datacenter Systems Prototype by applying principle of messaging services. On this architecture we ensure of asynchronous schema involve three client systems such as Juvia, Bob and Alice System. In the middleware DC_SYS with RabbitMQ Server will facilitate communication between Bob, Alice, and Juvia. Communication between Bob and DC_SYS uses Queue Message type, and communication between Juvia and Alice to DC_SYS uses Topic Message. It means if Bob wants to insert or update data in the databases, DC_SYS will save too, and inform Juvia and Alice data changes

In the architecture designs, there are four endpoints, that represent three endpoints were alias application on UPT TIK, then one is Datacenter ECR. For example, cases to represent of workflows schema were (1) Bob has data updates, (2) Bob data updates directly sends all updates to **QUEUE MESSAGE**. (3) Datacenter ECR (as DC_SYS) consumes those messages (4).DC_SYS deploys updates data into data warehouse databases. (4) DC_SYS puts all updates information into **TOPIC**

MESSAGES. (5) Message updates are ready to subscribe. (6) Juvia as subscriber on that channel can retrieve that information, and proceed it to save bob updates too into local databases **JUVIA**. (7) Alice as subscriber on those channels can retrieve that information, process it to save bob updates into local databases **ALICE.**

Based on that workflow, we can see data transmission workflow will directly publish any updates by sources systems to all client systems at UPT TIK who joined into channels. The solution of data transmission by messaging services can do asynchronous process then execute directly, without waiting any manual trigger by each system. In this part, we ensured all channel in the architecture must be built with dedicated port, not use 445 or 80 to those channels. The purpose is to prevent any of outside interaction over network to see data transmission events during sending and receiving process on that channel.

The message taken is still an encrypted data message placed by DC_SYS. Then each system will read the contents of the message that has been encrypted, then decrypt it with their respective private key. After the data is read, Juvia and Alice carry out additional operations related to the data obtained. All data transmitted on that channel is covered by E2EE schema, that all of systems has private and public key. That we use to guarantee data or information secure over the networks. For this research we use fully HTTP protocols to transmit all data. This process is to ensure each of systems who joined on this channel is valid and verified to consume all data and information by DC_SYS or client systems in UPT TIK UNDIKSHA.

4. Conclusions

Based on the research, prototype of datacenter implemented was by messaging services architectures. It is used to guarantee data transmission can process it with asynchronous schema over the network. DC_SYS will join into network of each client systems to ensure all data updates can be collected directly, then store it into data warehouse databases. If client system of UPT TIK joined into channel DC_SYS, clients system do not need to trigger manually to collect data updates by DC_SYS. By this schema, it will ensure and guarantee all data updates can be published by broadcasting TOPIC on those channels created by DC_SYS. All messages on the channel were encrypted by public key of DC_SYS or client system (if client system as sources to inform updates data to DC_SYS), those messages must be decrypted to see clearly the contents. All of client systems must open that content messages with private key.

5. Acknowledgments

The authors wish to thank ICT Department of Universitas Pendidikan Ganesha, Informatics of Education Department, Enginering and vocational Faculty of Universitas Pendidikan Ganesha.

References

[1] A. Dahlan, E. Utami, and E. T. Luthfi, "PERANCANGAN DATA WAREHOUSE PERPUSTAKAAN PERGURUAN TINGGI XYZ MENGGUNAKAN METODE SNOWFLAKE SCHEMA," VIII NomorJurnal Teknol. Inf., vol. 24, pp. 1907–2430, 2013.
[2] G. Singh, "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security," Int. J. Comput. Appl., vol. 67, no. 19, pp. 975–8887, 2013.
[3] Henriyadi, "Data Center Dan Implementasinya Pada Perpustakaan," J. Perpust. Pertan., vol. 17, no. 20, pp. 41–47, 2008.
[4] H. Hamad, M. Saad, and R. Abed, "Performance Evaluation of RESTful Web Services for Mobile Devices.," Int. Arab J. e-Technol., vol. 1, no. 3, pp. 72–78, 2010.
[5] S. Mumbaikar and P. Padiya, "Web Services Based On SOAP and REST Principles," Int. J. Sci. Res. Publ., vol. 3, no. 5, pp. 3–6, 2013.
[6] A. Rodriguez, "RESTful web services: The basics," Online Artic. IBM Dev. Tech. Libr., no. November, pp. 1–11, 2008.
[7] Obaida Mohammad and A. Al-Hazaimeh, "A New Approach for Complex Encrypting and Decrypting Data," Int. J. Comput. Networks Commun., vol. 5, no. 2, p. 88, 2013.
[8] C. Science and S. Engineering, "International Journal of Advanced Research in," vol. 3, no. 5, pp. 875–878, 2013.

[9] I. M. Putrama, G. R. Dantes, and D. G. H. Divayana, “Implementasi SOA menggunakan Apache ServiceMix dalam rancang bangun Data Warehouse (Studi Kasus di Fakultas Teknik dan Kejuruan Universitas Pendidikan Ganesha) (PDF Download Available),” Proc. Semin. Comput. Sci. Inf. Technol., pp. 127–138, 2016.
[10] P. Rola and D. Kuchta, “Implementing SCRUM Method in International Teams — A Case Study,” no. July, pp. 300–305, 2015.