ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 4, Issue 12, December 2019

A Proposal of IoT Message-oriented Service Framework for Serverless Software Architecture

[1] Sung Geun Yoo, [2] Won-Young Lee, [3] Sangil Park
[1][2][3] Graduate School of Nano IT Design Fusion, Seoul National University of Science and Technology, Seoul, Korea

***Abstract:* - In IoT service incurring massive data traffic, asynchronous service technologies such as Microservices or Representative safe transfer API (REST API) are rapidly introduced in order to increase the efficiency of the cloud computing service. Moreover, the emergence of function-based computing techniques called "serverless" service also boost the cloud-based service efficiency. In response to the technical shift, this paper proposes message format which can improve event stream process much faster in an asynchronous based microservice environment. This message format is compatible with the JSON Web Token (JWT) consequently, being capable of simultaneously transmitting events and commands and streaming processes as in image processing and real-time data analysis in an asynchronous based service.**

Key words: Microservice, JWT, Cloud Computing, Serverless, Event Stream.

1. INTRODUCTION

As many applications based on cloud computing have been proposed, various techniques for improving efficiency have been proposed [1]. Recently, Microservice Architecture or Serverless is proposed, which improves the efficiency of development and deployment based on Service Oriented Architecture (SOA). It adopts asynchronous distributed computing and functional programming techniques to solve response speed with clients [2-4].

However, the asynchronous based architecture focuses on immediate response to events without sharing memory for processing efficiency, so it cannot be a good solution for events where atomicity is required tasks such as real-time transactions. There are many difficulties in processing applications that IoT sensor data or real-time large-scale realistic broadcasting, which must be processed or real-time event streams [5]. Asynchronous processing has the disadvantage that the order of processing is not guaranteed even if the data comes in sequentially because it processes the event and receives the processed result in the form of a callback [5]. Already, such ideas as dynamic transaction processing for real-time distributed systems have been proposed by Yoon-Ik et al. [6]. Recently, however, the introduction of message queues such as Not Only Structured Query Language (NO-SQL), Remote Dictionary System (REDIS), and MessageMQ has resulted in the development of fast and reliable message passing techniques. Therefore, there is a need for new methods that are suitable for architectures such as microservices and serverless by improving existing methods.

In this paper, we propose a message framework that can deliver additional information such as sequence value of event and grouping which can be grouped each time an event occurs in order to control the event stream processed asynchronously in a serverless architecture. The proposal proposes a detailed message structure and features in Chapter 2, Proposed Method, in a format compatible with Recommendation RFC 7519 JSON Web Token (JWT) in the International Standardization Organization's current standardization track. Finally, Chapter 3 concludes with an overview of what asynchronous-based event stream processing is available for.

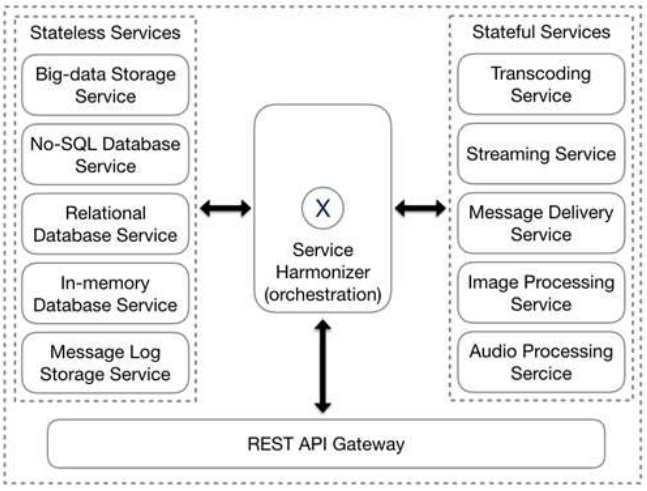

Figure 1: The Proposed IoT Message-oriented Service Framework

II. PROPOSED SERVICE FRAMEWORKS

In the server system based on the existing asynchronous processing, processing caused by sequential events had to be processed by using threads. However, flow control is difficult

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 4, Issue 12, December 2019

in an asynchronous server employing a distributed system. To solve this problem, record the sequence of events in the message format, store the values in an in-memory key-value store such as Redis, and then order them in the proper window size. By sorting the values, an asynchronous server can process the event stream without stopping the processing flow [5]. In this manner, the newly proposed IoT Message-oriented Service Framework is illustrated Figure 1. The proposed message format for processing such non-blocking event streams is shown in Figure 2. The message representing each event is divided into header, body, and verify signature like network packets. The header adds and stores the order of the event stream and information on the group to which the event belongs.

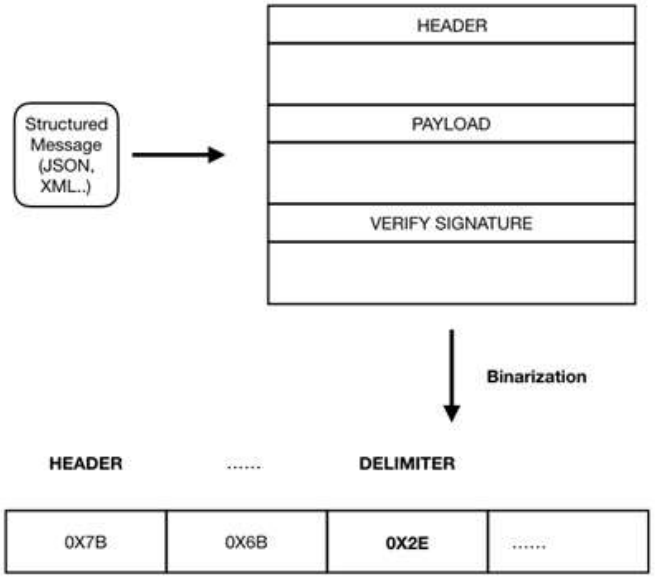

Figure 2: The Diagram of Proposed Message Format

By default, the sequence of events is managed by the client sending the event. The body is basically a structured document such as JavaScript Object Notation (JSON). JSON messages are vulnerable to errors and are not URL-safe. In other words, the JSON message contains a string that cannot be passed as a URL parameter. In addition, even small errors in a message are likely to propagate throughout the message structure. Therefore, the header and the body are encoded in Base64 format as shown in Figure 3. and separated by a dot character.

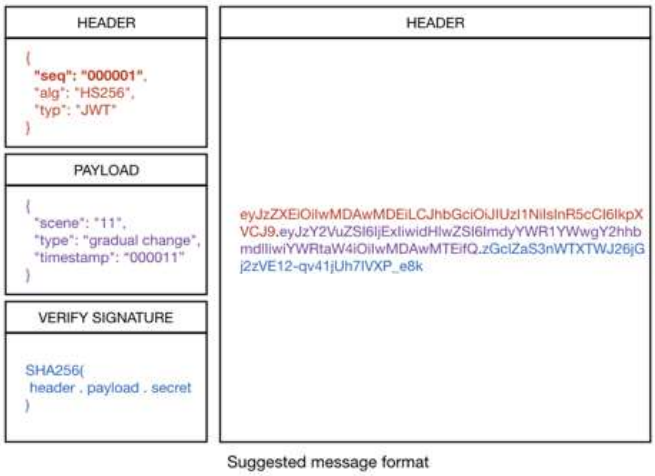

Figure 3: The Structure of Proposed Message Format

Using this method, the message format can be separated into the header, body, and verification signature through simple pattern matching. Therefore, even if the entire message is not decoded as shown in Figure 4, the headers can be separated to determine the order of events. Finally, the signature is a two-way hash with the Secret Message as the key value. If the server and client are sharing the same incognito message, you can use the verification signature to see if the contents of the header and body are correct. If someone who doesn't have an incognito message tampered with the header or body, the server can immediately see the tampering. The encoding method is illustrated in Figure 5.

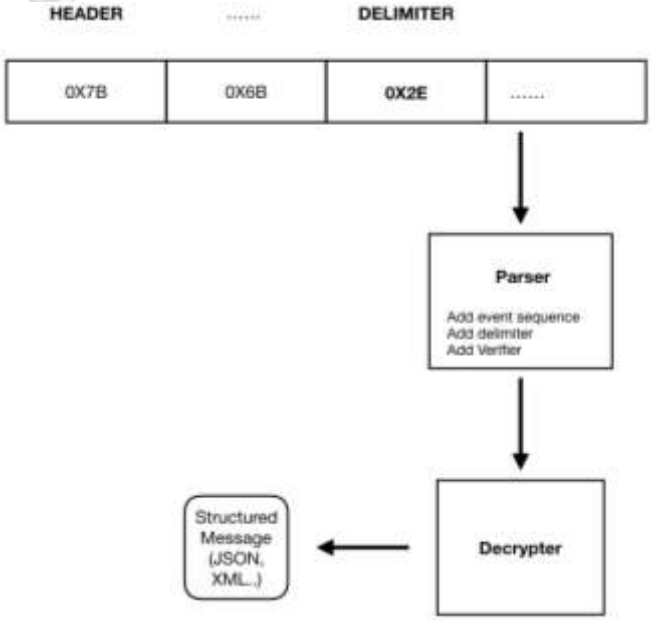

Figure 4: A Decoding Method of Proposed Message Format

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 4, Issue 12, December 2019

Basically, JWT was created to manage and issue tokens for user authentication on the web. And because it's on a standardized track, it already offers libraries in a variety of programming languages in ways that are likely to be used in the future. The proposed message format is compatible with the JWT standard and can be applied to frameworks that process asynchronous event streams through existing libraries.

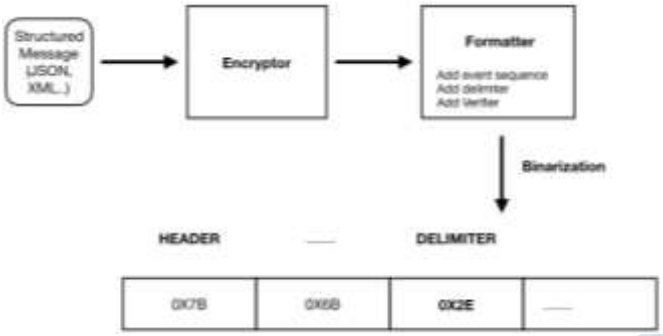

Figure 5: The Encoding Method of Proposed Message Format

III. CONCLUSION

In this paper, the message format proposed which has a very suitable structure for cloud application for event stream-based IoT sensor data processing or large-capacity media editing. In addition, it complies with the RFC 7519 JWT standard and has a merit that can be easily implemented in various languages even in an environment such as a microservice architecture or serverless. The proposed message format is currently registered patent in Republic of Korea (No. 1019281560000). In the future, many IoT/IoL applications will be enabled with our newly developed message format and message-oriented framework.

ACKNOWLEDGMENT

This research was supported by the MSIT (Ministry of Science, ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-2016-0-00311) supervised by the IITP (Institute for Information & Communication Technology Planning & Evaluation)

REFERENCES

[1] ZHANG, Shuai, et al. “Cloud computing research and development trend,” In: Future Networks, 2010, ICFN'10, Second International Conference on IEEE, 2010, pp. 93-97.

[2] M. Roberts, “Serverless Architectures,” The http://martinfowler.com/articles/serverless.html.

[3] BALALAIE, Armin; HEYDARNOORI, Abbas; JAMSHIDI, Pooyan. “Microservices architecture enables DevOps: migration to a cloud-native architecture.” IEEE Software, 2016, 33.3: 42-52.

[4] FOWLER, Martin; LEWIS, James. “Microservices. Thought Works.” http://martinfowler. com/articles/microservices.html, 2014.

[5] Sunggeun Yoo et al. “A study of message-oriented service framework for serverless software architecture. “, ISAAC 2016, AACL 08, pp. 214~215, 2016

[6] Yong-Ik Yoon. "A Study on the Asynchronous transaction processing in Distributed Real-time Database Systems." Database Research, 11.4 (1995.12): 38-50.