Proposal of a New software architecture for interoperability to improve the communication in the Edge layer of a smart IoT ecosystem

1st Juan Moreno-Motta
School of Informatics and System Engineering
Universidad Nacional Mayor de San Marcos
Lima, Perú
juan.moreno2@unmsm.edu.pe

2nd Felipe Moreno-Vera
Deparment of Computer Science
Universidad Católica San Pablo
Arequipa, Perú
felipe.moreno@ucsp.edu.pe

3rd Frank Moreno-Vera
School of Electronic Engineering
Universidad Nacional de Ingeniera
Lima, Perú
fmorenov@uni.pe

***Abstract*—In the current years, IoT has evolved to such an extent to extend to all corners of each place through devices, which connected to a network, either local or internet itself, generate information to be processed with a specific purpose, to this level there is a problem called interoperability of devices where not only is the compatibility of adding or removing devices to an ecosystem and there is compatibility, it is also expected that the information generated is standardized and optimized to transmit. This paper presents a new software architecture pattern for interoperability between devices that generate heterogeneous information in the edge layer of an IoT ecosystem.**

***Index Terms*—Interoperability, IoT, data encode, data decode, protocol buffer, Edge Computing, REST API, Software Architecture, IoT Ecosystem.**

I. Introduction

A. Background

The Internet of Things (from now until the end IoT) is a technology that has been emerging and converging of many technologies generating new paradigms to implement these architectures.

In 2014, Oliver Kleine [1] estimated that IoT devices will be increased at least 1.7 billions. In the same year, Utkarshani Jaimini [2] said the storage for those devices surpass 150 Exabytes in 2017.

According to Guinard et al [3], IoT is a system of physical objects that can be discovered, controlled or interacted with electronic devices that communicate through various network interfaces and can finally be connected to the Internet.

B. Present context

In 2017, Talavera et al. [4] study all IoT solution in topics like Agribusiness and Environment implementations, They found around 72 projects those represents all developed before, with this study they propose a general architecture for this solutions but they consider as challenge the way to implement a standard, compatibility and security guaranty between devices in the edges and cloud services.

Also, in 2017 Woznowski et al. [5] develop SPHERE (A Sensor Platform for Healthcare in a Residential Environment), in this project they have determined that there are 9 requirements that the IoT needs to cover to implement a smart city, but the most difficult requeriment to garanty and implement was the Interoperability.

Elsts et al. [6] based on SPHERE, consider that systems must comply with existing low-power IoT standards and protocols to (1) be susceptible to future extensions with third-party components; (2) reduce learning time for new staff.

Alkhalil et al. [7] mentions that the challenges related to the origin of the data, despite the current techniques, remains very challenging in its implementation and optimization. Therefore, this challenge of data origin, of the 7 main ones, is directly linked to interoperability, making it more difficult to deploy heterogeneous systems.

In two research about this, Pace et al. [8] and Madaan et al. [9] said that is very complicate to integrate all information generated by all systems with different devices, brands, hardware design, protocols, body message encode-decode, different programming languages, different data structs, etc.

C. Problematic and proposes to solve it

After of analysis all previous works above, we note the problematic was centering in how to implement a secure system that manage interoperability without complications.

Yacchirema et al. [10] present a platform oriented to Ambient Assisted Living (AAL), that platform called AAL-IoTSys is a prototype based on Wireless Sensor Network (WSN) with heterogeneous devices using Binary encoder to transfer information between low power devices.

Androec et al. [11] propose web semantic to allow IoT interoperability using JSON-LD. with this method they can create a description about data information and link objects and properties in a JSON file.

Sun et al. [12] proposes a REST API for Web of Things which work with JSON and compare Micro-services architecture against Monolithic architecture. They use REST API to communicate all devices including the IoT ecosystem, the control of the environment through the central service is dynamic.

Lim et al. [13] and Malik et al. [14] propose and compare architectures based on SOAP (XML) and REST API (JSON) services applied to try to improve scalability of interoperability, but they do not have success.

Kum et al. [15] based on previous works (some of them was mentioned above) they propose an architecture for Fog Computing applications, in this case, Fog Computing allows to manage better the information between edge layer and cloud services trough middle servers called "Fog servers" (See Fig.1).

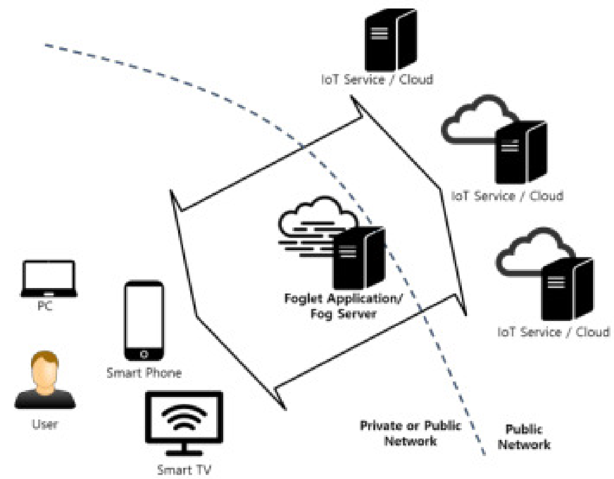

Fig. 1. kum et al. Architecture [15].

Luan et al. [16] propose to create a routing gateway to manage all information and communication between devices (edges), fog nodes and cloud servers. They work with a network topology which improve and reduce time to transfer data. The aim was achieve to devices to communicate all servers and nodes was mobiles, the architecture transmits data trough mobiles (5G infrastructure) as end users or edge layer (See Fig.2).

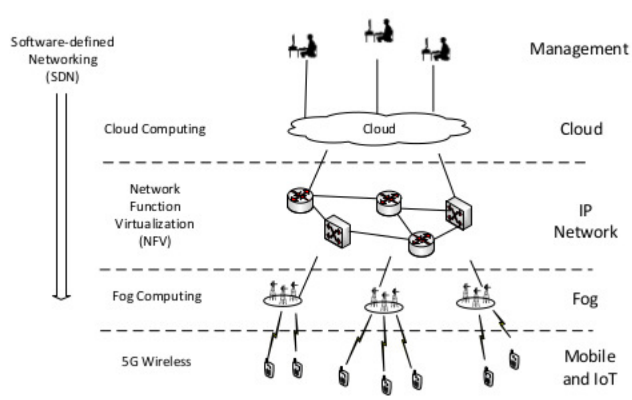

Fig. 2. luan et al. Architecture [16].

Lysogor et al. [17] conduct a study on the transfer and exchange of data in heterogeneous networks where there is an absence of network infrastructure and their research focuses on the use of satellite networks, however, there is a great limitation on the size of the data transmitted. They show that the binary format generated by Protocol Buffer allows more bytes to be transferred than the JSON format (See Fig.3).

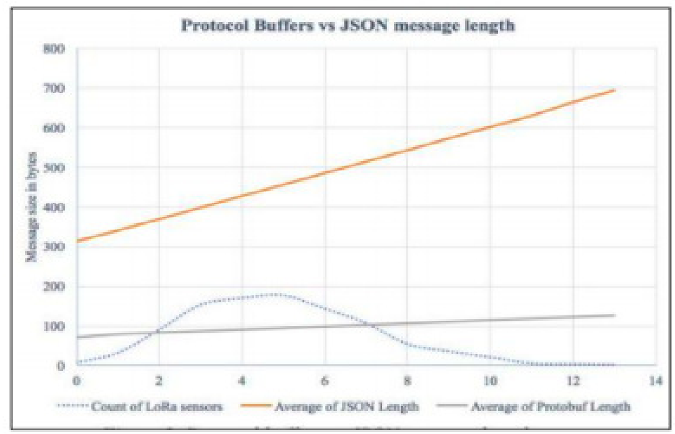

Fig. 3. lysogor et al. performance chart [17].

Nitin Naik [18] write a complete document about protocols, applications, operating systems, and other metric for different situations in IoT environments. In this document they present HTTP, AMQP, MQTT and CoAP protocols (commonly used in IoT solutions) (See Fig.4). According authors, the user can decide their relevant usage in IoT systems based on their requirements and suitability.

Criteria	MQTT	CoAP	AMQP	HTTP
1. Year	1999	2010	2003	1997
2. Architecture	Client/Broker	Client/Server or Client/Broker	Client/Broker or Client/Server	Client/Server
3. Abstraction	Publish/Subscribe	Request/Response or Publish/Subscribe	Publish/Subscribe or Request/Response	Request/Response
4. Header Size	2 Byte	4 Byte	8 Byte	Undefined
5. Message Size	Small and Undefined (up to 256 MB maximum size)	Small and Undefined (normally small to fit in single IP datagram)	Negotiable and Undefined	Large and Undefined (depends on the web server or the programming technology)
6. Semantics/ Methods	Connect, Disconnect, Publish, Subscribe, Unsubscribe, Close	Get, Post, Put, Delete	Consume, Deliver, Publish, Get, Select, Ack, Delete, Nack, Recover, Reject, Open, Close	Get, Post, Head, Put, Patch, Options, Connect, Delete
7. Cache and Proxy Support	Partial	Yes	Yes	Yes
8. Quality of Service (QoS)/ Reliability	QoS 0 - At most once (Fire-and-Forget), QoS 1 - At least once, QoS 2 - Exactly once	Confirmable Message (similar to At most once) or Non-confirmable Message (similar to At least once)	Settle Format (similar to At most once) or Unsettle Format (similar to At least once)	Limited (via Transport Protocol - TCP)
9. Standards	OASIS, Eclipse Foundations	IETF, Eclipse Foundation	OASIS, ISO/IEC	IETF and W3C
10. Transport Protocol	TCP (MQTT-SN can use UDP)	UDP, SCTP	TCP, SCTP	TCP
11. Security	TLS/SSL	DTLS, IPSec	TLS/SSL, IPSec, SASL	TLS/SSL
12. Default Port	1883/ 8883 (TLS/SSL)	5683 (UDP Port)/ 5684 (DLTS)	5671 (TLS/SSL), 5672	80/ 443 (TLS/SSL)
13. Encoding Format	Binary	Binary	Binary	Text

Fig. 4. Nitin comparison table [18].

Petersen et al. [19] make a demonstration of the performance of the different formats, arriving at the conclusion that the binary format generated with Protobuff developed by Google is the one of better performance and less memory use (See Fig.5), It is observed that Protocol Buffer for any communication protocol can serialize many more messages per second, being one of the protocols with better ZeroMQ performance.

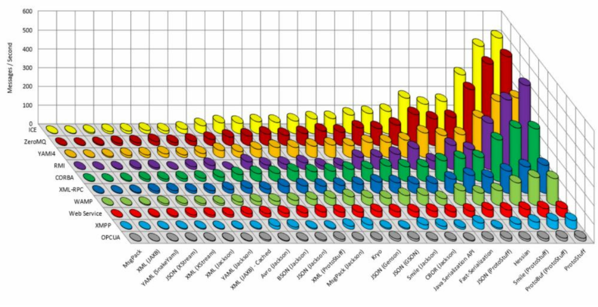

Fig. 5. petersen et al. performance chart [19].

II. PROPOSAL

After to identify all applications and architectures proposed to solve the challenge of devices interoperability, we propose our architecture to manage in a better way this problem, using Fog Computing to manage IoT solutions as components, we define an Ecosystem as the process to communicate data between devices or different IoT solutions or different Frameworks (See Fig.6) based on encode-decode data serialization.

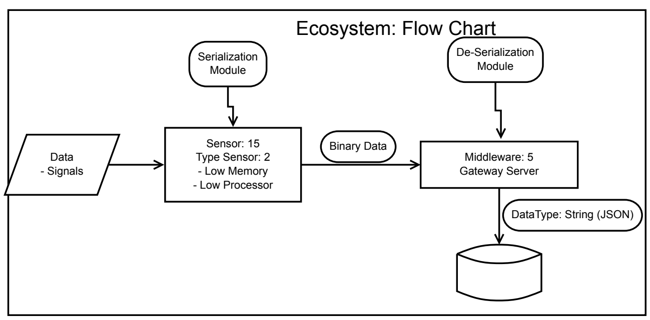

Fig. 6. Ecosystem design.

We define Ecosystems like a process to encode-decode data, based on Micro-services architecture. The data exchange between devices is very important when you need to get or analyses by sectors and that the reason to divide all interaction for a better management of the complete system.

Then, to manage all communication and interaction between devices we define our Fog Ecosystem (Ecosystem - Fog Computing) subdivide in 7 important components (See Fig.7):

- Services: This component is the service layer that Fog Computing can deliver so that other external devices can receive information (from sensors) or send an action to be executed (actuators).
- Repository: This component is of persistence where deserialized data frames will be stored in JSON format.
- Processing: This component is responsible for processing the binary data formats that are received from the component "Device", previously identifies the type (category to identify encode-decode algorithm, origin, etc) of device and the device that comes as data within the frame.
- Management interface: This component provides an administration interface. This administrator registers the types of devices and devices, as well as the Middleware and the ecosystem to which one or more Middleware belongs.
- Security: This component is used by the Middleware to validate that the devices that are connecting have the required authorization and can thus receive the data frames in binary format that is sent by the devices.
- Middle-Ware: This component is responsible for receiving the binary format, makes use of the "Processing" component that has greater processing capacity since its operation is that of an exclusive processing server. It also makes use of the File System component as an ecosystem configuration file.
- File System: This component is the Fog Ecosystem logger. Also, storage all data decoded.

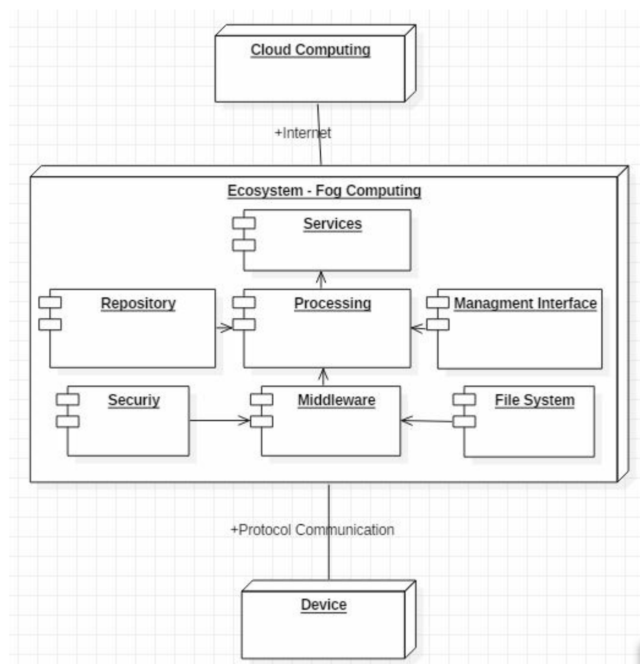

Fig. 7. Architecture proposed.

Then we have 2 more components: devices and Cloud Computing with protocols like internet (for Cloud services) and Protocol buffer to interact with devices.

- First: "devices component" represent all devices which will be connect to Ecosystem trought Middle-Ware component, in this part is implemented encode-decode data (based on protocol buffer), in general could be sensors or any devices which generate data.
- Second: "Cloud Computing" represent the final repository of data analysis of all information collected and preprocessed from Fog Ecosystem.

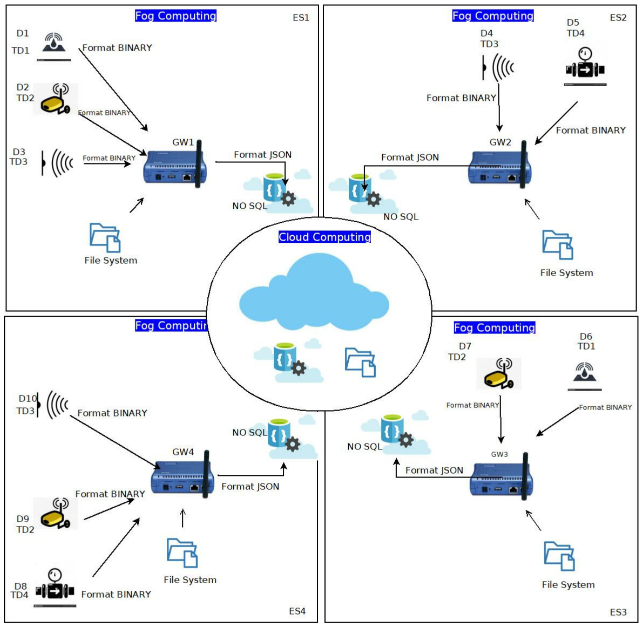

Fig. 8. Architecture proposed for multiple solutions.

III. Conclusions

In this paper, we propose a new architecture for interoperability using Fog and Edge computing to manage and improve smartly communication and transfer information between devices at edge-fog level in the IoT ecosystem that can be integrated with other ecosystems achieving scalability and identifying the origin of the data, keeping them all ordered and communicated among themselves.

To achieve this, we use Protocol buffers with our data standardized format for all devices (or at least most of them) to get a better performance and get less latency transferring data and a high flexible fog architecture to manage all dynamic changes in devices.

Future work

We are implementing this architecture in real time applications to get our proof of concept of our proposed architecture, our test is a real time system monitors of data measure from environment, we use Temperature, Humidity, Sound and Monoxide sensors, Raspberry Pi 3 Model B (1.4GH, Quad Core) board, Arduino board, SQLite, protocol buffer to encode-decode data and Cloud services like Firebase to storage data and analyze, compare, measure latency, compatibility and processing or organizing data speed between each layer of the ecosystem (See Fig.8).

Acknowledgment

This work was supported by grant 234-2015-FONDECYT (Master Program) from CienciActiva of the National Council for Science,Technology and Technological Innovation (CONCYTEC-PERU). Also, thanks to SCOTIABANK MASTER PROGRAM GRANT for the financial support to Juan Moreno.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

References

[1] Oliver Kleine, CoAP Endpoint Identification - A Protocol Extension for Crowd Sensing in the mobile Internet, 2014 IEEE International Conference on Internet of Things (iThings 2014), Green Computing and Communications (GreenCom 2014), and Cyber-Physical-Social Computing (CPSCom 2014).

[2] Jaimini, U. (2017). PhD Forum: Multimodal IoT and EMR based Smart Health Application for Asthma Management in Children.

[3] Guinard, D. D. and Trifa, V. M. (2016). Building the Web of Things. NY 11964: MANNING.

[4] Talavera, J. M., Tobn, L. E., Gmez, J. A., Alejandro, M. A., Aranda, J. M., Parra, D. T., Garreta, L. E. (2017). Review of IoT applications in agro-industrial and environmental fields.

[5] Przemyslaw Woznowski, Alison Burrows, Tom Diethe, Xenofon Fafoutis, Jake Hall, Sion Hannuna, Massimo Camplani, Niall Twomey, Michal Kozlowski, Bo Tan, Ni Zhu, Atis Elsts, Antonis Vafeas, Adeline Paiement, Lili Tao, Majid Mirmehdi, Tilo Burghardt, Dima Damen, Peter Flach, Robert Piechocki, Ian Craddock, George Oikonomou, SPHERE: A Sensor Platform for Healthcare in a Residential Environment. https://link.springer.com/chapter/10.1007/978-3-319-44924-1_14. Last visited dec 2018.

[6] Elsts, A., Oikonomou, G., Fafoutis, X. and Piechocki, R. (2017). Internet of Things for Smart Homes: Lessons Learned from the SPHERE Case Study.

[7] Alkhalil, A. and Ramadan, R. A. (2017). IoT Data Provenance Implementation Challenges.

[8] Pace, P., Gravina, R., Aloi, G., Fortino, G., Fides-Valero, A., Ibaez-Sanchez, G., Yacchirema, D. (2017). IoT platforms interoperability for Active and Assisted Living Healthcare services support.

[9] Madaan, N., Ahad, M. A. and Sastry, S. M. (2017). Data integration in IoT ecosystem: Information linkage as a privacy threat.

[10] Yacchirema, D. C., Palau, C. E. and Esteve, M. (2017). Enable IoT Interoperability in Ambient Assisted Living: Active and Healthy Aging Scenarios.

[11] Androec, D., Toma, ,. B. and Kiasondi, T. (2017). Interoperability and Lightweight Security for Simple IoT Devices.

[12] Sun, L., Li, Y. and Memon, R. A. (2017). An Open IoT Framework Based on Microservices Architecture.

[13] Lim, N., Majumdar, S. and Nandy, B. (2010). Providing Interoperability for Resource Access Using Web Services.

[14] Malik, S. and Kim, D.-H. (2017). A Comparison of RESTful vs. SOAP Web Services in Actuator Networks.

[15] Kum, S. W., Moon, J. and Lim, T.-B. (2017). Design of Fog Computing based IoT Application Architecture.

[16] Tom H. Luan, Longxiang Gao, Zhi Li, Yang Xiang, Guiyi We and Limin Sun (2016). Fog Computing: Focusing on Mobile Users at the Edge.

[17] Lysogor, I., Voskov, L. and Efremov, S. (2018). Survey of Data Exchange Formats for Heterogeneous LPWAN-Satellite IoT Networks.

[18] Nitin Naik. Choice of effective messaging protocols for iot systems: Mqtt,coap, amqp and http.In2017 IEEE International Systems Engineering Symposium (ISSE), pages 17, 2017

[19] Petersen, B., Bindner, H., You, S. and Poulsen, B. (2017). Smart Grid Serialization Comparison.