Exploiting Interoperable Microservices in Web Objects Enabled Internet of Things

Muhammad Aslam Jarwar, Sajjad Ali, Muhammad Golam Kibria, Sunil Kumar, and Ilyoung Chong
Department of Information and Communications Engineering
Hankuk University of Foreign Studies, Yongin-si, Korea
{aslam.jarwar, sajjad, kibria, sunil75umar, iychong}@hufs.ac.kr

Abstract— **The vision of internet of things is to connect every object beyond the boundaries of anywhere and whenever for the provision of services from the smart home user to fourth industrial revolution. The objects in the internet of things will be more heterogeneous because of their specialty and accuracy in their domain. Now a days in the Internet of things the data and its machine understanding gets more importance and remain the focal point rather than the objects that generate that data changes frequently due to innovations and their wears and tears. To make the machine understanding, exchanging and sharing information and knowledge; these objects needs a light weight and novel platform for the provision of IoT services in the future. The Web of Objects is a platform for IoT services because of its service modularity with the support of interoperable microservices and virtualization and granularity of heterogeneous objects through composite virtual objects and virtual objects. To realize the cross domain IoT applications we proposed WoO enabled interoperable microservices architecture and present a reference use case for the implementation.**

Keywords— Web of Objects; Interoperability; Microservices; Internet of Things; Interoperable Architecture; Composite virtual object; virtual object

I. INTRODUCTION

The vision of Internet of Things (IoT), is to connect every object on the earth to the internet in a way, so that they can communicate, share, exchange & understand the data, information, and knowledge in order to provide cross domain IoT services for achieving a certain types of applications and solutions. There are a large number of domains of these applications and solutions includes smart health and care, applications in fourth industrial revolution, disaster recovery and rehabilitation, smart e-governance, smart home, connected car, cyber physical systems for self-driving of automobiles, smart agriculture, smart farming, and user-centric ubiquitous services in user smart spaces.

When these large number of applications connected to the internet, will definitely connects a large number of objects to the internet, and create, exchange data for communication and accomplishing their functionality with the objects beyond their natural domains. The communication and creation of data out of their natural domains will change the physical world into a home of information and knowledge. In a study [1], it is mention that IoT is not only to connects devices from the physical world to the web for communication but also these devices will transform the web from communication and linking channel to data exchange channel, and this transformation will enhance the capabilities of the objects to become smarter. The predictions of these large number of objects increase to two hundred billion in 2020 [2]. This increase in a large number of applications and their large number of objects from various domains will create many challenges and opportunities for the scientist, engineers, academics and workforce concerned to the internet of things. According to broader concept and vision of IoT, in the future IoT, every connected object will belong to multi-domains and needs different technologies, fast implementations and fast improvement and replacement.

Among these challenges the following three are the most important and are the problem statements of this article.

- How to consider the communication of these billion of objects from different domains and handling the complexities of semantic cooperation among them.
- How every connected object understands the data and information of each other to provide IoT services for accomplishing the common goal?
- How to support that the functionalities of every connected object, rapidly implemented in best fit and lightweight technologies and deployed independently, and less centralized management in order to rapid scalability, recovery, and resiliency?

Due to importance role of semantic web technologies in semantic interoperability for exchange and sharing information created from sensors, actuators or objects data across the entire value chain. Due to the importance and hype of microservices from Amazon, Netflix towards internet of things to support dynamism of development, deployment, and scalability and in order to cope with these three challenges, opportunities as per the requirement of broader concept of future IoT applications and services, in this paper we present a novel semantic interoperable architecture which uses state of the art microservices concept in Web of Objects (WoO) platform [3].

To the best of our knowledge, that this semantically interoperable architecture based on microservices concepts for the internet of things middleware is first one and overall will reduce the dearth in semantic, and lightweight but powerful middleware IoT platforms. Therefore in our architecture:

- The complexity of the first challenge will be supported by the virtualization of objects in a decentralized manner and by using semantic ontology. Each object from different domains will operate in its own container and communicate and cooperate with semantic ontology.

- The second challenge will be supported by WoO platform that supports to harmonize two or more objects for understanding, creating, and extracting the knowledge, which is used for intelligence context aware IoT services and applications.
- The third challenge will be supported by the concepts and patterns of microservices, which best support the concepts of best fit technology, lightweight and plug & play.

The semantic ontology based microservices approach in the WoO architecture aims to hide the complexities of communicating, processing and understanding the data shared and exchanged by the variety of objects from a variety of domains. The interoperable microservices based WoO architecture encourage rapid semantic interoperability, scalability, and replaceability by encapsulating & virtualizing diverse objects and communication protocols with proxy microservice components.

II. BACKGROUND AND STATE OF THE ART

The phrase of "Internet of Things" was first discovered by Kevin Ashton while working for the company of supply chain management "Procter & Gamble" in 1999 [4]. However in the past, IoT focus only on the data generated by the sensors and almost these sensors belonged to single domain applications and usually, that data was used in only within that domain. But with the passing of time and new innovations in sensor technologies and web technologies transforms the IoT from a single domain to multi-domain and after that to social IoT. For multi-domain and social IoT objects, interoperability is the key challenge in order to share and exchange data, information, and knowledge. Further, goes into detail, let's overview about interoperability.

A. Interoperatability

It is "the ability of two or more systems, components or objects to communicate in a way so that share data and use the information" [1]. There are three types of interoperability:

- **Technical Interoperability** concerned with hardware, software components; it makes possibilities for machine-to-machine communication.
- **The Syntactical Interoperability** deals with data formats; when data transferred from one object to another object. The data should be in well-defined, and unambiguous schemes and encoding. e.g. JSON, XML and HTML
- **The Semantic Interoperability** deals with the uniform understanding of concepts by the humans rather than machine understanding of the content [1], but in other study [5] the semantic interoperability is a way through which objects communicate data that interpreted in the same manner by both sender and receiver objects.

The above discussed three types of interoperability could be handled with semantic web technologies. E.g. linked data (RDF, RDFa, XML, and GRDDL), vocabularies (OWL, SKOS, and RIF), queries (SPARQL) and inferences (Jena Framework, RACER).

In one of the recent study [5], authors used the concept of semantic information broker (SIB). The SIB is the type of store, which contain data as RDF graphs, and an agent is used to query the store by using the SPARQL query language. The SIB is a single monolithic component, which needs to change from application of one domain to another domain. In studies [7, 8, and 9], authors handled the semantic interoperability by using Web Ontology Language (OWL), and RDF. OWL and RDF are used to represent and common data format for sharing, among objects and extracting the information and knowledge. In these studies they used the OWL, and RDF for virtualizing the real world objects by virtual objects (VOs), and then matching and linking the received data through (VOs) in composite virtual objects (CVOs), in order to exploit context-aware, emotion aware and knowledge based IoT services to handle emergency situation in the smart home and smart shopping mall.

The two main problems in studies [6, 7, and 8] discussed above and in [9, 10, and 11] that these approaches focus only on single domain IoT applications and using monolithic approach. The approach used in these studies is not suitable for cross-domain future IoT applications. Because the future IoT applications should be interoperable, encapsulated, automated, decentralized, scalable, secure, not complex and should support rapid development and deployment with latest technologies without affecting other objects and applications in the environment.

B. Microservices

The term microservice was first used by Martin Fowler in 2014 [11], and it is the extension of service oriented architecture (SOA). For the IoT applications, SOA and microservices have a similar objective of constructing multiple services and applications from the bundle of small services. The concept and significance of microservices has been less explored for the IoT environment, only a few studies [12, 13, 14, and15] discussed the advantages of microservices in the IoT environment. Therefore let to explore some significance of microservices for the current and future IoT applications. The figure 1 shows some of the vantages of microservices for IoT environment overall and particularly for our study.

- **Complexity Driven -** when billions of objects communicate and exchange information in future IoT applications the single business logic and codebase will make the system more complex, by breaking into small parts with microservices architecture; may drive out the complexity of the system [12].
- **Loose coupling** - microservices works in an isolated environment and communicate with each other through message coordinator (e.g. RabbitMQ, AMQP, MQTT) only when needed. The broken microservices which stream data from sensors or activating object through actuator may not influence to other microservices performance and do not become a reason of degradation of whole IoT system.
- **Lightweight -** the tiny sensors or objects distributed in a large area, and their maximum energy consumed in transmission of data to the central point. By embedding lightweight microservices in these types of objects to process, aggregate data at their own level and send only information to the central point, when there is a change in information.

- **Rapid Development** - In an IoT system every microservice is independent of other microservice, and it fulfils only a single feature or task. So the developer team develops without waiting for other teams to finish the development of their service
- **Scalable** - In the IoT environment and cyber physical systems; initially few objects and services are operated. But when the demand for services and objects increases; then to extend the system become a challenging and complex task. In microservices architecture, we have to only create more instances of already existing microservices. In microservices, scaling can be achieved with three ways; horizontal duplication, vertical decomposition, and diagonal partitioning.

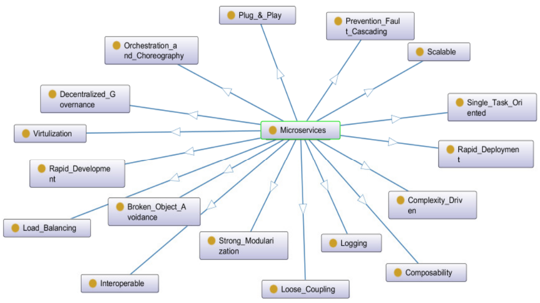

Figure 1. Microservices significance for IoT environment

- **Interoperability** - Interoperability is the core issue of user specific and industrial IoT systems. The protocol used for communication and exchange of data has great importance in IoT systems. In WoO components [3]; we use microservices with semantic web technologies for achieving the interoperable communication in both same and cross-domain applications.
- **Single Task Oriented** - Microservices architecture supports one task one microservices policy. Here the task can be just to read the value from the sensor and to update the status of the object from zero to one [12].
- **Broken Object Avoidance** - In the existing and future IoT systems the objects are distributed in different demographic locations. The circuit breaking mechanism in microservices architecture ensures that not to use broken object or VO in WoO platform.
- **Load Balancing** - In WoO platform [3] every component implemented with supported microservices. The load balancing mechanism; distribute the load justifiably on each service instance for maximizing service utilization and throughput.
- Logging - In IoT environment the objects and services are distributed on remote sites; whereas the services utilization may be on the other part of the world. The logging mechanism of every microservice individually in microservices architecture quickly helps the system administrator and developer in fault diagnosing and system auditing.
- **Virtualization** - Microservices supports object virtualization [16], which is the key concept in WoO platform. In WoO platform we use semantic web technology oriented microservices for increasing the interoperability and virtualization of physical world objects; which is the essential requirement of industrial and future IoT ecosystems.
- **Strong Modularization** – Microservices communicate through explicit interfaces; such as HTTP and REST or through MQTT. This communication mechanism covers up the dependencies in IoT objects. These dependencies creep up with IoT objects from the development technologies e.g. Node.js NPMS, java jars, .Net Assemblies. In WoO platform this concept has great suitability to support local databases module, security module, and user interfaces module for domain expert, and knowledge engineer and definitely may save time to easily plug-in new module.
- **Plug & Play** - is the core concept of WoO platform. The concept of microservices architecture to support rapid development, deployments and replaceability in an isolated manner help us to harmonize VO(s), CVO(s) and services rapidly for current and future IoT application across the boundary.
- **Decentralized Governance (DG)** - IoT services built with microservices architecture uses its own technological pile and runtime environment in a distributed locations. Therefore they are independent of each other and can be managed superficially. In WoO platform DG will support at every component of each layer separately to domain experts and knowledge engineers in rapid policy creation for extracting the real world knowledge from the data; which is created by the objects of the physical world.
- **Composability -** supports the approximation detection and re-using the existing virtual objects (VOs) and composite virtual objects (CVOs). Because Composability supports fine-grained arrangement through existing microservices. The ontology matching & alignment protocols are usually used in these microservices

C. Web of Objects (WoO)

According to the ITU recommendation number ITU-T Y.4452, the concept of WoO "is a realization way of the IoT services, where virtualized objects (i.e., VO and CVO) are connected, controlled and incorporated with a resource to facilitate development, deployment, and operation of IoT services on the World Wide Web". The WoO supports semantic web modeling with semantic web technologies to support interoperability for cross-domain IoT services and applications.

The VOs are used to virtualize the real world objects and provides the relevant information about the status of the object in the real world. The every VO has unique URI [9]. The VOS contains the semantic ontology model for multiple domain interoperability and information reusability. For the use case; the execution and inference of VO ontology implementation accomplished with the microservices pattern by using semantic web technologies and java programming language.

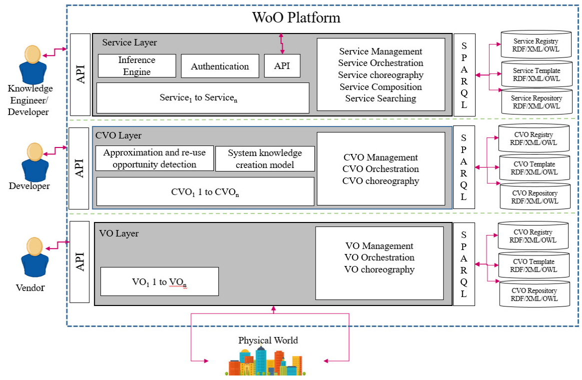

Figure 2. High-level view of WoO Platform

The composite virtual object (CVO) is the semantic conglomeration of multiple VOs to execute the features of services [9]. The CVO contain the information (etc. time, location, profile) about the VOs, it uses for abstracting service features. The execution and inference of CVO ontology also developed with multiple microservices.

In IoT environment mainly we deal with structured data, but the recent trend to integrate social media services and other information modules (i.e. weather forecasting). We also need to focus on unstructured data. Because the data belongs to social media and network mostly contain unstructured format [17]. Thanks to WoO, it supports to process both types of data formats in order to obtain the IoT services which mainly based on the knowledge creation [8], situation awareness, and situation projection.

In WoO platform the real world objects connected through VOs for receiving streams of data and for performing some actions on real world objects. For example receiving current temperature, current humidity and number of occupancy in the living room for adjusting room temperature. Here like, when, why, how, at what level the room temperature adjusted will be decided at CVO level by the CVO like “room_temprature_monitoring” CVO.

But here the main questions is who and where will be decided that which VOs/CVOs are used to execute this functionality? This decision (i.e. policies) will be taken place at the service level, and these policies are created by the knowledge engineer or domain expert at that level. Each three layers (VO, CVO, and Service) have its own databases which are implemented with semantic web technologies and the SPARQL is used as the semantic web query language for retrieving the rules (RDF, OWL) and business logic from these databases as shown in figure 2. Further, about the WoO components, the study [6, 8, and 9] explains briefly.

III. Proposed Interoperable Microservices in Web Objects

In the internet of things domain, most of the IoT middlewares follow the bottom-up approach for designing IoT architectures and their implementation. One of the reason of using bottom-up approach; is to interact with multiple real architectures and their implementation. One of the reason of using bottom-up approach; is to interact with multiple real world objects from multiple domains. The WoO existing architecture support bottom-up approach for sensing the data about the environment and top-down approach for exposing the RESTful web services and APIs for IoT applications and other platforms as shown in figure 3. But the microservices based architectures are more positive to a top-down approach. So here for our semantic interoperable microservices based WoO architecture, we balanced the things. As the objective of microservices to break the complex system into parts; to get drive out the complexity. Here we use green field microservices approach in WoO platform architecture; which has great significance for the internet of things.

The figure 3 shows the WoO architecture in details. At the service level there are two types of HTTP REST requests but with different types of request and response format. The one request (left side) is from the IoT applications which use simple XML and JSON data formats and the other request (right side) is from the other third party platforms and this type of request response uses the RDF and web ontology language. The reason to use the RDF and web ontology language data; is to make them semantic interoperable, so that other platforms can use and semantically understand the service features (i.e. CVO, VO) and customize them in their own way. In figure 3 the request authentication service, authenticate the user, application and request type and forward this call to “request analysis service”. The “request analysis service” analyze which composite virtual object(s) and virtual objects are necessary to execute the requested service. Then the “choreography service” do the necessary tasks such as calling to “request logging & history” microservices to log the request and response parameters. It calls “application & user registration” microservice in the case of new user and applications wants the services. The “Semantic Knowledge Synthesizer” service takes the real world data from the service database and service registry and applies the machine learning techniques (i.e. association rule mining, K-mean clustering, Linear regression) in order to find the real world situation and context of the user for providing him better IoT services. The “semantic service discovery” and “semantic service matcher” microservices are called when the user uses the natural language as input in IoT application for receiving the IoT service. So this microservice discover the appropriate requested service. “The semantic service composition” microservice composes the requested service with the corresponding list of composite virtual object(s) and virtual objects (VO) as depicted in figure 4. Then the execution of service request through CVO(s) and VOs could be achieved with multiple microservices. The number of microservices decided on the weightage of resources used in CVOs and in VOs.

The CVO layer receives the composed service in the form of RDF/XML graph containing the list of CVO and VO. The microservice “Decompose Service composition” and validate the list of CVO and VO and also check the current status of CVO with the microservice “Check current status of CVO” from the CVO registry database. The “Obtain CVO List” microservice fetch the corresponding CVO ontology from CVO database, declares the CVO instances and execute it with multiple microservices (here apache Jena is a great choice for microservices implementations). The other microservices at CVO layer such as “Semantic CVO Discovery” is for searching a CVO in the case of natural language request, “Semantic CVO Registration” microservices is for registering a newly created CVO. The “Semantic System Knowledge Synthesizer” service at CVO layer comprises with four microservices as in figure 3. This service uses the machine learning techniques (i.e. association rule mining) for realizing

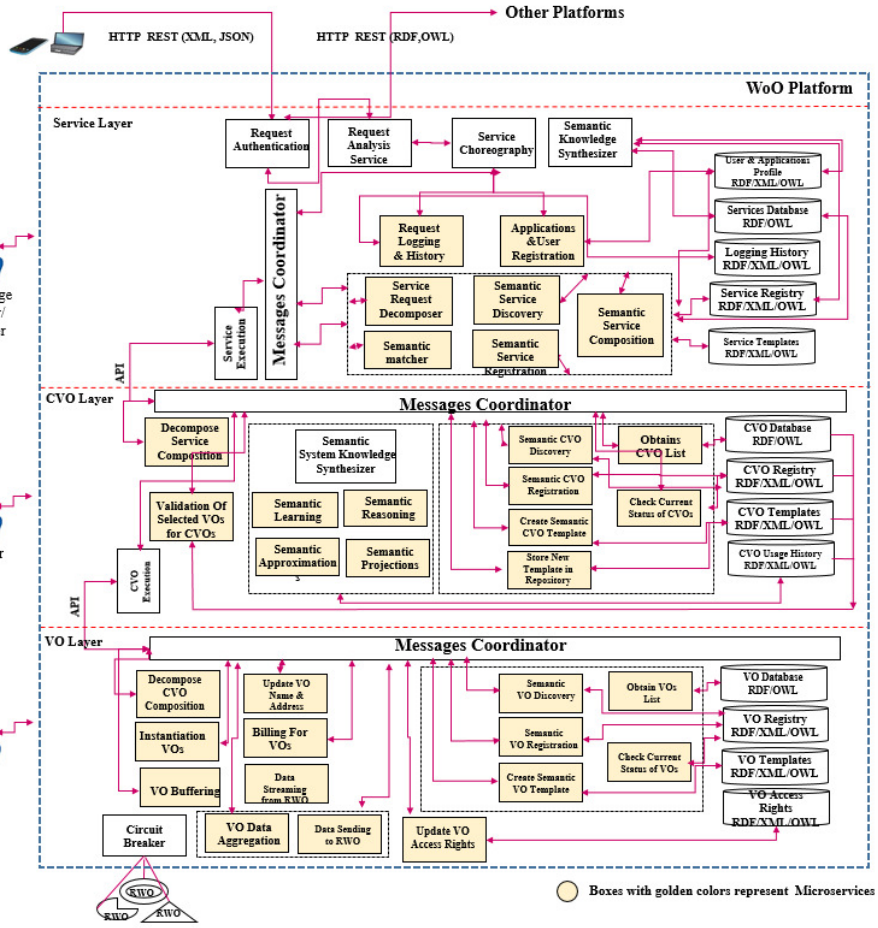

Figure 3. Interoperable Microservices in Web Objects

the CVO ecosystem and helps in reusing the CVOs in the same situation.

The role of VO level is to virtualize the real world heterogeneous objects for the retrieving and streaming the data from real world objects and map this data according to rules defined in VO ontology. At this level, there is individual single microservice for each task because the VO is the representation of the small and tiny real world object. The microservice "Decompose CVO Composition" decomposes the RDF/XML graph sent by the CVO layer (as discussed in CVO layer) and other operations perform with helper microservices. After the execution of requested VO(s), the corresponding data sent back to CVO level for the aggregation and realization of the requested service.

```
<!-- http://www.webofobjects.com/registrylist#Service -->
<owl:Class rdf:about="http://www.webofobjects.com/registrylist#Service">
<!-- http://www.webofobjects.com/registrylist#CVO_01 -->
<owl:NamedIndividual rdf:about="http://www.webofobjects.com/registrylist#CVO_01">
    <rdf:type rdf:resource="http://www.webofobjects.com/registrylist#CompositeVirtualObject"/>
    <registrylist:has_vo rdf:resource="http://www.webofobjects.com/registrylist#VO_001"/>
    <registrylist:has_vo rdf:resource="http://www.webofobjects.com/registrylist#VO_002"/>
    <registrylist:has_vo rdf:resource="http://www.webofobjects.com/registrylist#VO_003"/>
    <registrylist:cvo_id rdf:datatype="http://www.w3.org/2001/XMLSchema#string">C-01</registrylist:cvo_id>
    <registrylist:uri rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://www.webofobjects.com/cvo/c01</registrylist:uri>
</owl:NamedIndividual>
    <!-- http://www.webofobjects.com/registrylist#CVO_02 -->
<owl:NamedIndividual rdf:about="http://www.webofobjects.com/registrylist#CVO_02">
    <rdf:type rdf:resource="http://www.webofobjects.com/registrylist#CompositeVirtualObject"/>
    <registrylist:has_vo rdf:resource="http://www.webofobjects.com/registrylist#VO_003"/>
    <registrylist:has_vo rdf:resource="http://www.webofobjects.com/registrylist#VO_004"/>
    <registrylist:has_vo rdf:resource="http://www.webofobjects.com/registrylist#VO_005"/>
    <registrylist:cvo_id rdf:datatype="http://www.w3.org/2001/XMLSchema#string">C-02</registrylist:cvo_id>
    <registrylist:uri rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://www.webofobjects.com/cvo/c02</registrylist:uri>
</owl:NamedIndividual>
```

Figure 4. List of Composite Virtual Objects (CVOs) & Virtual Objects (Vos) for the requested service to the CVO Layer

IV. IMPLEMENTED SCENARIO & USE CASE

The figure 5: depicts the implemented scenario. In the figure, the user has multiple smart spaces to receive IoT services whenever he wants and wherever he goes. These smart spaces contain four domains. Three domains (connected car, smart home, and workplace) are implemented on proposed architecture and another one is implemented with third party IoT platform. The intra-operability in first three domains become some easy due to the same platform and for cross domain interoperability we use the "translating and formatting" server. In this scenario, the WoO server and "translating & formatting" component is the most important part.

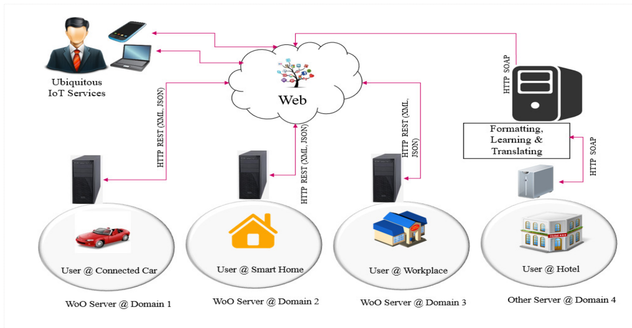

Figure 5. Scenario & Use Case Based On Interoperable Microservices for Ubiquitous IoT Services

In the given scenario the domain 1 to 3 have separate servers loaded with WoO platform (discussed in section III) and the domain 4 server is loaded with third party server. In the first three domains, when the user moves from one to another location, and if one of the service feature (CVO, VO) is not available then that domain server directly requests from other domain (with the same platform). Internally for the communication microservices are used. These microservices contains the data with metadata (CVO/VO ontologies) which is easily understandable from the same platform.

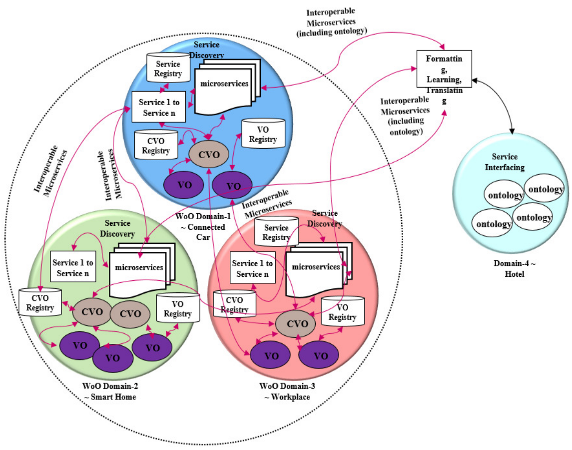

Figure 6. Interoperable Microservices within same domain and cross domain

However to make interoperability among objects with domain 4 server, we used another server loaded with a software which first checks the consistency between the requested and served ontology, then it analyzes, learn, format and translate the given ontology with respect to domain 4 server and vice versa. In this, all mechanism microservices are used because of to update, create and delete any missing and new ontology become easy. The internally implemented mechanism is elaborated in figure 6. When there is a request from one domain to other domain to use the CVO/VO of that domain; because the requested domain has some missing CVO/VO for realizing the IoT service. The copy of requested CVO/VO ontology and relevant data transferred to request domain server by using the microservices and the requested domain server uses microservices for acquisition and processing the same to satisfy the requested service by the user.

CONCLUSION

In this paper a novel interoperable microservices web objects enabled architecture for internet of things services has been presented. This architecture focuses on interoperable microservices for same domain and cross domain IoT services. The microservices has the greatest importance for perceiving the objects as an interoperable for exchanging and understanding the data, information, and knowledge because of lightweight pattern, easily usable for tiny ontology transformation and alignment, rapid development, strong modularity, plug & play and avoidance of broken object. The paper discussed challenges in the internet of things and also explains the solutions through interoperable microservices with the support of composite virtual object (CVO) and virtual objects (VO). The sample of metadata RDF/OWL based graph presented between the service layer and VO layer. A scenario and implemented use case of smart spaces within same and cross domain WoO supported interoperable microservices has been presented.

ACKNOWLEDGMENT

This work was supported by Institute for Information & Communication Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.B0113-16-0002, Development of Self-Learning Smart Ageing Service based on Web Objects).

REFERENCES

[1] IERC, AC. "IoT semantic interoperability: research challenges, best practices, recommendations and next steps." European Commission Information Society and Media, Tech. Rep 8 (2015).

[2] https://goo.gl/yVwNfE last access 2017-04-01

[3] Web-of-Objects (WoO)-ITEA2 Project Jan 2012–Dec 2014. https://itea3.org/project/web-of-objects.html.

[4] Rayes, Ammar, and Samer Salam. "Internet of Things (IoT) Overview." Internet of Things From Hype to Reality. Springer International Publishing, 2017. 1-34.

[5] D'Elia, Alfredo, et al. "Enabling Interoperability in the Internet of Things: A OSGi Semantic Information Broker Implementation." International Journal on Semantic Web and Information Systems (IJSWIS) 13.1 (2017): 147-167.

[6] Kibria, Muhammad Golam, and Ilyoung Chong. "Knowledge-based open Internet of Things service provisioning architecture on beacon-enabled Web of Objects." International Journal of Distributed Sensor Networks 12.9 (2016): 1550147716660896.

[7] Jarwar, Muhammad Aslam, and Ilyoung Chong. "Exploiting IoT services by integrating emotion recognition in Web of Objects." Information Networking (ICOIN), 2017 International Conference on. IEEE, 2017.

[8] Kibria, Muhammad Golam, et al. "A User-Centric Knowledge Creation Model in a Web of Object-Enabled Internet of Things Environment." Sensors 15.9 (2015): 24054-24086.

[9] Shamszaman, Zia Ush, et al. "Web-of-Objects (WoO)-based context aware emergency fire management systems for the Internet of Things." Sensors 14.2 (2014): 2944-2966.

[10] Ali, Sajjad, Muhammad Golam Kibria, and Ilyoung Chong. "WoO enabled IoT service provisioning based on learning user preferences and situation." Information Networking (ICOIN), 2017 International Conference on. IEEE, 2017.

[11] Kibria, Muhammad Golam, Hwa-Suk Kim, and Ilyoung Chong. "IoT learning model based on virtual object cognition." Information Networking (ICOIN), 2016 International Conference on. IEEE, 2016.

[12] Butzin, Björn, Frank Golatowski, and Dirk Timmermann. "Microservices approach for the internet of things." Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on. IEEE, 2016.

[13] Vresk, Tomislav, and Igor Čavrak. "Architecture of an interoperable IoT platform based on microservices." Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2016 39th International Convention on. IEEE, 2016.

[14] Vögler, Michael, et al. "A scalable framework for provisioning large-scale IoT deployments." ACM Transactions on Internet Technology (TOIT) 16.2 (2016): 11.

[15] Butzin, Björn, Frank Golatowski, and Dirk Timmermann. "Microservices approach for the internet of things." Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on. IEEE, 2016.

[16] Newman, Sam. Building microservices. " O'Reilly Media, Inc.", 2015.

[17] Jarwar, Muhammad Aslam, et al. "CommuniMents: A Framework for Detecting Community Based Sentiments for Events." International Journal on Semantic Web and Information Systems (IJSWIS) 13.2 (2017): 87-108.