applied sciences

MDPI

Article

# A Function as a Service Based Fog Robotic System for Cognitive Robots

**Hyunsik Ahn**

Department of Robot System Engineering, Tongmyong University, 428 Sineon-ro, Nam-gu, Busan 48520, Korea; hsahn@tu.ac.kr; Tel.: +82-10-2518-4777

Received: 1 October 2019; Accepted: 24 October 2019; Published: 27 October 2019




**Featured Application: Robot Cloud and Cognitive Robots.**

**Abstract:** Cloud robotics is becoming an alternative to support advanced services of robots with low computing power as network technology advances. Recently, fog robotics has gained attention since the approach has merit relieving latency and security issues over the conventional cloud robotics. In this paper, a function as a service based fog robotic (FaaS-FR) for cognitive robots is proposed. The model distributes the cognitive functions according to the computational power, latency, and security with a public robot cloud and fog robot server. During the experiment with a Raspberry Pi as an edge, the proposed FaaS-FR model shows efficient and practical performance in the proper distribution of the computational work of the cognitive system.

**Keywords:** robot cloud; cognition as a service; cognitive robots; sentential cognitive system; cloud service; human–robot interaction

## 1. Introduction

Cloud computing is a widely advancing information and communications technology service and is a key technology of the advanced industry. Robot clouds applying cloud computing to robots allows robots to connect to a cloud environment, uses a huge computational infrastructure, and obtains the results of high level programs from the cloud [1]. The cloud robots share information including environments, actions, and objects and offload heavy computation to a cloud server [2].

However, such cloud robot services could give rise to security issues of privacy breaches and latency issues of control signal delays for robot motions. Recently, to solve these issues, fog robotics, distributing computing work properly with fog servers and edges, is receiving attention for its advantages in reducing latency and security matters (Figure 1) [3–5].

These merits of fog robotics can accord with cognitive robots to reduce the cost of the robot and its human–robot interaction (HRI) services. If the cognitive robot adopts the fog robotics model for offloading burdened computing tasks to clouds or fog servers, it also needs to consider privacy, security, and latency as well as abundant computing power for advanced intelligent functions. Cognitive robots represent experienced cognitive information, store it in a proper form, and retrieve it using a reasoning procedure [6,7]. This means that the fog robotics model of cognitive robots needs to consider the characteristics of the cognitive structure [8].

In this paper, a function as a service based fog robotics (FaaS-FR) model for the sentential cognitive system (SCS) of cognitive robots is proposed. The FaaS-FR model includes the edge as the local robot system, the fog robot servers for private, security, and computing power, and the robot clouds for high performance computation. The previous SCS consists of multiple modules to recognize new events that the robot has experienced and describes them in a sentential form to be stored in a sentential memory and retrieved with a reasoning process in the future [9]. In this approach, the SCS adopts

FaaS-FR, and each module of the SCS is classified with the functionality of privacy, security, and latency as well as required computing power. According to the functionality, the computation of modules is executed on the edge or offloaded to fog robot servers or robot clouds.

The merit of FaaS-FR is that advanced services utilizing high performance computation is possible even in the edge system of low cost and low computing capability. A module in the SCS acquires and transfers raw data to the fog sever or a cloud. Then, the server processes the data and sends back the results to the SCS. In the implementation and test, we can observe that the FaaS-FR model makes cognitive robots more efficient via proper distribution of the computing power and information sharing.

The contributions of this study are as follows: (1) a fog robotics model, FaaS-FR model, is suggested to be applied to a cognitive robot for efficient and advance services at a low cost; (2) with this model, a functionality-based modular networking in the SCS of a robot is proposed and tested.

This paper is organized as follows. In Section 2, related work on robot cloud and fog robotics are described. Section 3 details the theoretical background of the proposed FaaS-FR model. Section 4 describes an application of the model to a SCS for a cognitive robot platform. Section 5 provides the implementation of the proposed approach to a service robot through experimental results. Finally, conclusions and future work are presented in Section 6.

## 2. Related Work

Cloud robot services applying the technology of cloud computing to robots utilize computation, storage, and communication in the internet infrastructure. For instance, RoboEarth, a pioneering robot cloud service, supports the service of storing software components, mapping information, behavior information, and object information in a database (DB) and utilizing a cloud engine for services; robots can use it by virtualizing the information [1].

However, there have been different opinions regarding the usefulness of robot clouds. An advocative side, on one hand, agrees that a robot can enhance its capability by combining it with robot cloud services. Kuffner insists that the robot cloud, for a robot system having limited computing power, can carry over-burdened tasks to cloud servers [10]. On the other hand, Laumond noted that the approach of robot clouds could result in harmful effects to the performance of robots [11]. He insisted that the rapid advancement of real-time capability of on-board processing could make the concept of cloud robotics meaningless, and the cloud robot depending on the network could weaken the autonomy and reliability of robots.

Nevertheless, a robot cloud has characteristics of cloud computing and robot technology and the unique characteristics of the robot cloud itself. Particularly at the robotics level, robots in the robot cloud can communicate with each other to share the burden of tasks rather than serve as isolated devices that allow only data exchange with remote servers. As a result, the robot cloud will optimize the outstanding achievements that exist in robotics.

There are different levels of service in the field of cloud computing comprising infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) [12]. These models can be applied to robot clouds and multiple studies regarding them have been introduced. Mourandian et al. [13] highlighted aspects of IaaS in robotic applications. They proposed an architecture that enables cost-effectiveness by delegating virtualization and dynamic tasks to robots, including robots that could belong to other clouds. Gerardi et al. [14] proposed a PaaS approach to a configurable suite of products based on cloud robotics applications. It allowed end users to be free from the low-level decisions needed to construct architectures of complex systems distributed among robots and clouds. For example, if the REALabs platform was built using a PaaS model, it was predicted that many robot applications could be developed in this area [15]. In the framework of SaaS, a robot used a remote server for the training of locations with a neural network [2]. This case was used to establish communication between the cloud and the robot through a wide environment and to identify its location in images transmitted by the robot at the SaaS level. Chen and Hu [16] described robot as a service (RaaS) with the Internet of Things. The RaaS was able to create a local pool of intelligent

devices using autonomous and intelligent robots and make local decisions without communicating with the cloud.

However, networking robots have given rise to security and latency issues. DeMarinis et al. [17] determined that a number of robots are accessible and controllable from the Internet and dangerous to both the robot and human, and the robot's sensors can be viewed as a threat to privacy. Chinchali et al. [4] issued that cloud robotics that come with a key communicating with the cloud over congested wireless networks may result in latency or loss of data, and formulated offloading as a sequential decision-making problem, and proposed a solution using deep reinforcement learning.

To overcome security and latency matters of cloud robotics and fog robotics, a new network model was introduced [18]. The model, as shown in Figure 1, distributes storage, computer, and networking resources between the cloud and the edge in a federate manner. In the work of Tian et al. [19], a dynamic visual servo system was suggested with a fog robotics approach. It separated the tasks with a cloud for high level perception, an edge for time critical tasks, and a cloud–edge hybrid for the majority of robot applications. Tanwani et al. [3] introduced a fog robotics approach for secure and distributed deep robot learning. It provides that deep learning models are trained on public synthetic images in the cloud, and the private real images are adapted at the edge within a trusted network which reduces the round-trip communication time for inference of a mobile robot. In this approach, an edge is a gateway or a local server supporting robots with better security, privacy, and high speed communication. Gudi et al. [5] suggested a fog robotics model that a fog robot server gathers the cloud information and provides it to the private robots as a broker.

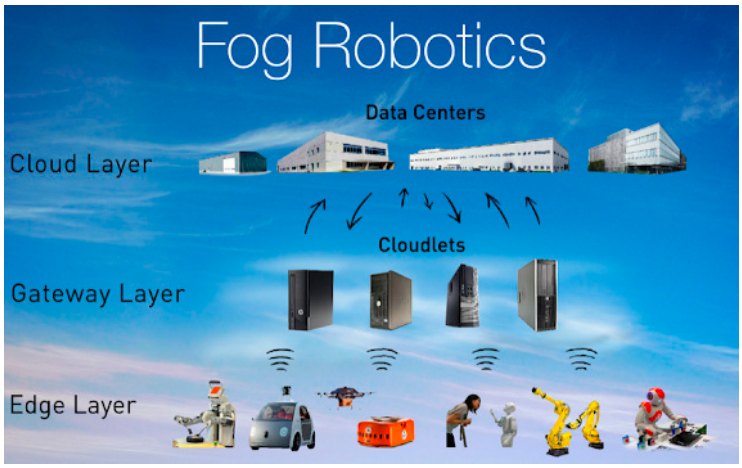



**Figure 1.** Fog robotics schematic [18].

In summary, robot cloud systems aim to improve robot performance with the aid of clouds enhancing computing power and data management capability. However, the robot cloud structure is challenged by the issues of real-time control, synchronizations, and stability risks. Alternatively, fog robotics aim to improve the performance of robots by distributing computation burden with cloud, fog, and edge computing works, and enhancing security and latency. However, each approach has its own fixed task distribution model, which can cause a lack of flexibility. In this paper, a more comprehensive model of fog robotics is proposed. The model provides that clouds, fog servers, and edges are related to the cognitive system of robots for effective and safe task distribution according to the characteristics of cognitive information. In the next section, distributing computing tasks according to cognitive functions based on the SCS is detailed.

## 3. FaaS-FR Model for Cognitive Robots

In the cloud robot paradigm, there are various deployment models, including private, community, public, and hybrid clouds. Wang et al. [20] introduced these models into the robot clouds. This study

proposes and evaluates various connection methods that occur in the implementation of cloud robots. Public cloud models exchange large amounts of data and information across networks and clouds. The cloud can be used to share data in a computing environment that everyone can share. However, the exclusive materials of an individual should be served in a proper separate manner. In a personal robot cloud, a server or cloud is privately connected to a home or company. Personal robot clouds can form an external and independent cloud and distribute the robot's computing power through the servers.

In the view of fog robotics, the function of personal cloud servers can be matched with fog robot servers which personally support edges (local robot system) [5]. Therefore, we adopt the term "fog robotic server" instead of "cloud robotic server" because the server works not for other's robots but for specific robots privately.

However, fog robotics models generally have a hierarchical model consisting of clouds, fog servers, and edges. With these models, it is difficult for edges directly to access to a cloud and get the result of services which have specific functionality with enough computing power.

In this paper, to overcome these matters, an advanced model, FaaS-FR model, is proposed as shown in Figure 2. In the model, all the functional modules of the cognitive robot are classified according to privacy, security, and computing power, and have their own networking with concept of fog robotics. The functions of the robot are suitably divided for being worked on edges, fog robot servers, and public robot clouds, respectively. In the case of information possibly being a violation of privacy, it is computed and stored in an edge or a fog robot server. If the edge needs to use the public robot cloud to utilize an advanced computing service, it can access the cloud directly to reduce latency or go through a fog robot server to the cloud. The reason that a new term is coined, FaaS, is that the classified functions of the robot can be offloaded on the cloud or fog servers according to the security, latency, privacy, and computing power.

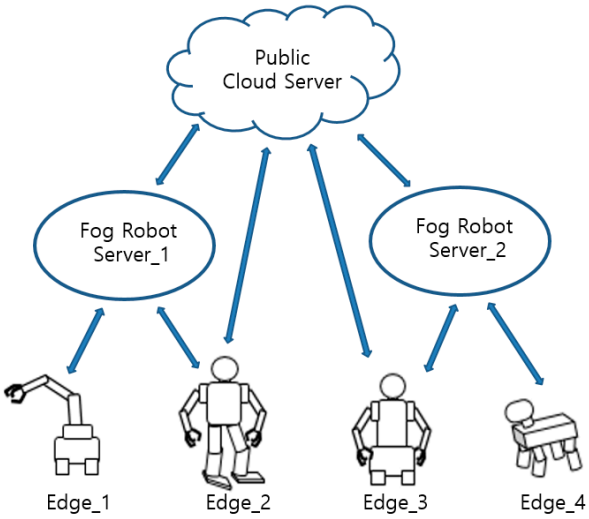



**Figure 2.** The schematic of function as a service based fog robotic (FaaS-FR) model.

Figure 3 shows a schema of modular cognitive functions of a general structure of cognitive robots. It has perception modules comprising sensing, object recognition, and speech recognition. Therefore, the robot can talk with humans regarding the visual situation that the robot recognizes. The robot also has behavior modules such as utterance and motion. In the higher part, there are interpretation and generation modules which can produce descriptive cognitive information from the perceptional. Memory modules are used to store the descripted cognitive information and can be retrieved for the future by a reasoning procedure with a virtual imager and cognitive grammar. In the view of functionality including privacy, security, safety, latency, and computing power, the functions can be

divided with three parts: sensing and actuation part (SAP), privacy and security part (PSP), and high performance part (HPP).

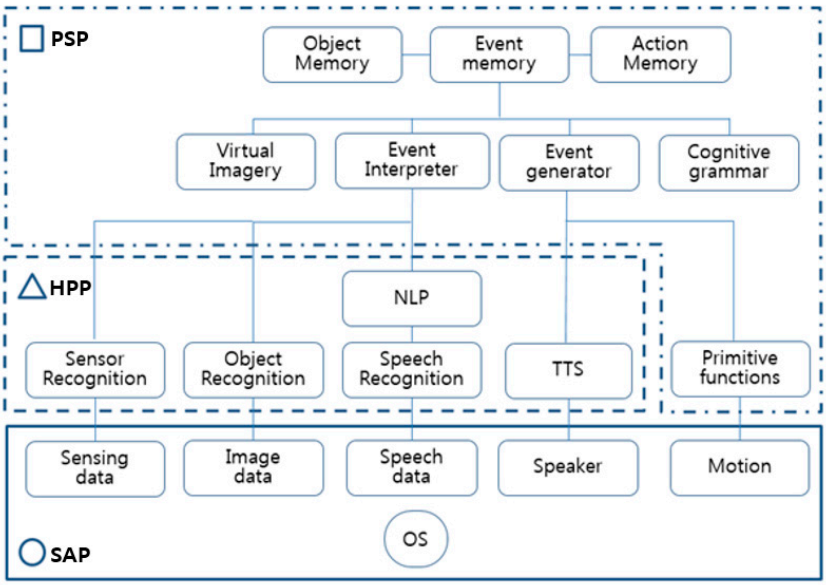



**Figure 3.** A schema of cognitive robot functionality. SAP: sensing and actuation part; PSP: privacy and security part; HPP: high performance part; NLP: natural language processing; OS: Operating System

The SAP which is covered by a solid line and marked with a circle, is the essential part including operating system (OS), sensing, and actuation which are indispensable for the robot. This part should have OS, perception functions including sensors, cameras and microphones, and behavior functions including speakers and actuators. These functions are dependent on the hardware of the robot and cannot be taken over by others.

The PSP is covered with a dashed single-dotted line and marked with a rectangle; it should be installed on edges or fog servers. For low-cost robot services with minimum computing power and network infrastructure, this part needs to be offloaded to clouds; however, these functions can be related to private and security information. Therefore, it is reasonable that this part is working on the edges or fog servers. In the case that the private information is not serious and well secured, it could be offloaded to a public cloud.

The HPP has a dashed line with a triangle; it is dependent on public robot cloud tools that can supply high-quality performance such as speech recognition, natural language processing (NLP), 2D and 3D object recognition, and text-to-speech (TTS). The Google Cloud application programming interface (API) supports multiple modules of deep learning in its public cloud [21].

Figure 4 shows various distribution of functions and offloading levels. As shown in the top, from left to right, the functions of the robot are dependent of the servers and clouds. On the contrary, the security and privacy could weaken with the direction.

The stand-alone type (Figure 4a) is a conventional type of robot computing in which the edge covers all parts (SAP, PSP, and HPP). Figure 4b,c both use private servers for the functional parts. Edge_3 of Figure 4c offloads the PSP to the private server, then the edge can work with lower computing power. In these cases, the HPP and PSP should be developed and installed on the server by developers for the functions of the robots.

In the case of the conventional robot cloud models (Figure 4d,f), all the high performance computations are offloaded on the cloud. In the case of Figure 4d, the edge has computing power to cover PSP. However, the model of Figure 4f transfers the PSP to the cloud to reduce the computational load of the edge. We will be able to see this case for mass robot service providers. The public cloud should safely manage private information for the robot.

The model of Figure 4e shows the typical characteristics of fog robotics. It provides flexibility in adopting applications among the edge, the server, and the cloud. Specifically, Private Server_3 can be a fog robot server to do the job of PSP. This is applicable to achieve high quality functions with low computing power by offloading them to clouds. In this case, there are two kinds of services: The first method is that the edge directly accesses the cloud for the HPP service. The other method is that the fog robot server mediates for receiving the service from the cloud and additive processing, and then transfers it to the edge. In this paper, the model in Figure 4e is adopted as the FaaS-FR model.

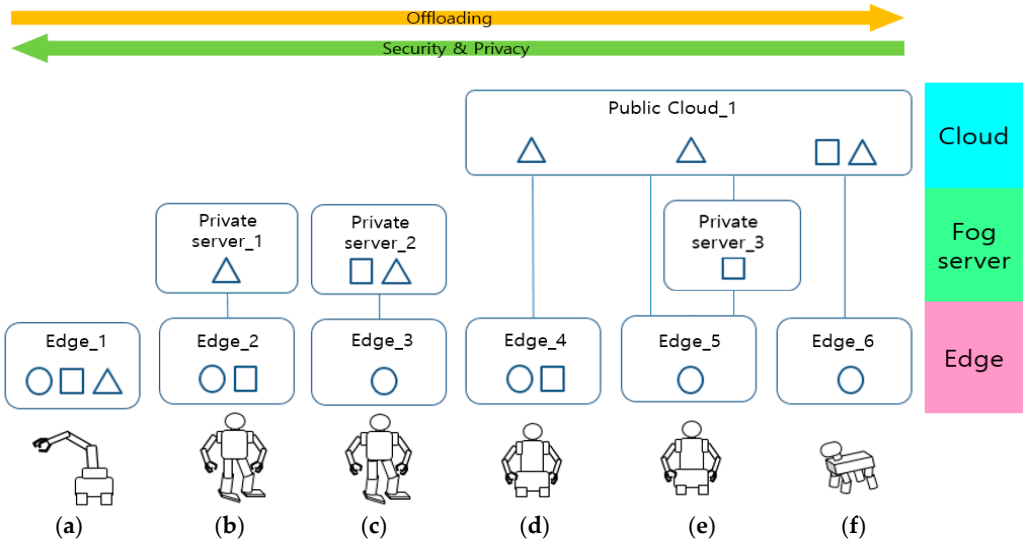



**Figure 4.** Various distribution of robot functions and offloading levels: (**a**) stand-alone type; (**b**) and (**c**) use private servers for offloading computation burden; (**d**–**f**) adopt clouds. The proposed FaaS-FR adopts model (**e**).

The FaaS-FR model can be applied according to the specified functionality and the service level. Table 1 shows the level of FaaS in the fog robotics model. In the cloud, the function is offloading high performance computation. In the case of the fog robot server, it is used for privacy and security as well as computing power distribution. Edges, shallow computing systems, cover OS and elementary data acquisition and actuation. For the clouds and fog servers, both PaaS and SaaS can be adopted according to the functionality and the computing power of the edge. In the PaaS case, the user should develop an application using APIs supported by the PaaS [22]. On the contrary, SaaS supports applications without any application development.

**Table 1.** The characteristics computing of FaaS-FR.

| Computing Type | Functional Parts | Service Levels |
|---|---|---|
| Public robot cloud | HPP | PaaS, SaaS |
| Fog robot server | PSP | PaaS, SaaS |
| Edge | OS, SAP | - |

PaaS: platform as a service; SaaS: software as a service.

## 4. A SCS Model Based on FaaS-FR

Figure 5 shows the proposed FaaS-FR model which is applied to the SCS of a service robot. The functions are categorized according to cognitive functions for allocation according to functionality. The modular functions with a dotted line are offloaded to a public robot cloud or a fog robot server. The applications for the functions use APIs of PaaS or SaaS of the cloud and fog server.

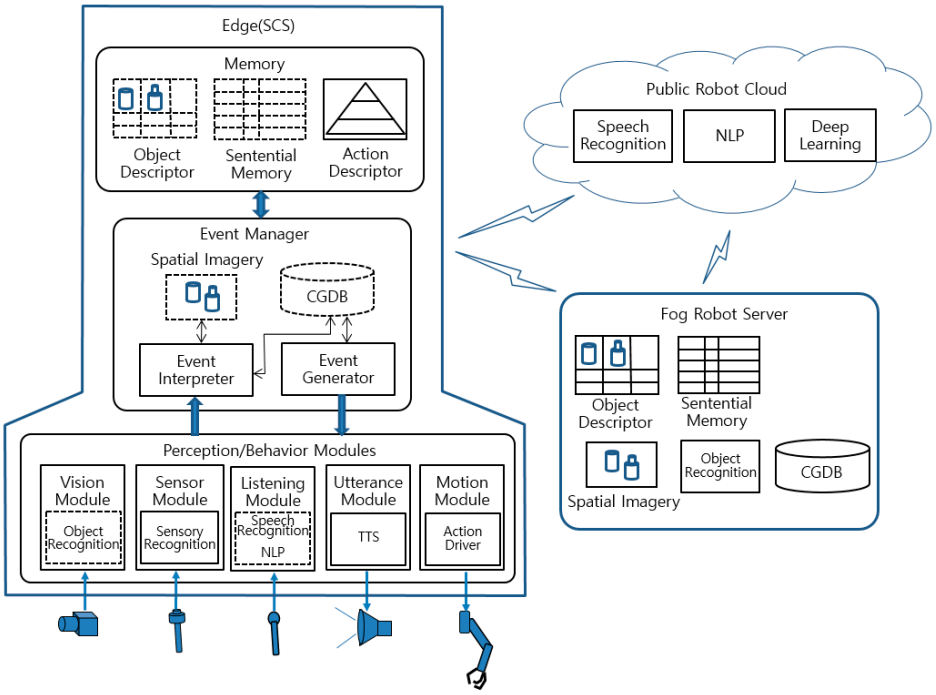



**Figure 5.** The block diagram of the sentential cognitive system (SCS) based on FaaS-FR that offloads the computation of modules. CGDB: cognitive grammar database; TTS: text-to-speech.

In the memory of the SCS, the sentential memory stores a series of sentences describing cognitive information of events as shown in Table 2 [9]. When an event occurs in a module, the system converts the cognitive information of the event to a sentential form and stores it in a sentential memory. Each sentence has a modular and time tags for being used to query the memory for reasoning. SCS uses an object descriptor to store the features of objects, such as labels, shapes, and current poses, for expressing visual events. The motion descriptor stores the information of physical actions of the robot hierarchically [23]. Each module of the memory is related to the privacy and security and indispensable for the essential functions of robots. In the view of FaaS-FR, the memory modules can be worked on the edge when the computing power is enough. If the computing power of the edge is limited, the task of the module can be moved to the fog robot server.

**Table 2.** Examples of sentences stored in sentential memory.

| # | Time | Module | Sentence | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | VERB | AUX | ARG1 | ARG2 | SPACE | TIME |
| 1 | $t_1$ | VM | appeared | - | (NP A cup) | - | (PP at (NP $x$, $y$, $z$)) | |
| 2 | $t_2$ | LM | move | - | - | (NP the cup) | (PP to (NP the front) (PP of (NP the bottle))) | - |
| 3 | $t_3$ | MM | picked up | - | (NP I) | (NP the cup) | - | - |

AUX: auxiliary verb; ARG: argument; VM: vision module, LM: listening module, MM: motion module; NP: noun phrase; PP: propositional phrase.

The event manager controls the interpretation and reasoning of events. The event interpreter interprets the cognitive information obtained from the modules and creates sentences. The event manager stores the sentences in the sentential memory. Schematic imagery is an imitation of a human mental model for spatial reasoning. If the SCS needs a reasoning of the visual situation at a certain time, it produces a virtual scene by placing the models of the objects and derives sentences that express the spatial context of the scene. The cognitive grammar database (CGDB) has grammar rules to generate sentences from the cognitive information of the events. For the purpose of FaaS-FR, the function of the

event manager is essential for the robots to work properly. Therefore, the modules can be worked on the edge or in the robot fog servers.

There are perception and behavior modules linked to and from the external world in the lower part of Figure 5. The vision module is used to recognize visual events by capturing scenes by a camera and recognizing objects. The sensor module includes all sensing functions of the robot acquired by data acquisitions that include physical contacts, sound, and temperature. The listening module captures human speeches and transfers it to the cloud to use a speech recognition application to get sentences. Then, it analyzes the acquired sentences via an NLP including syntactic and semantic parsing. The utterance module generates sentences using a sentence generator and utters them with TTS application.

The action module controls the motion of the robot. A physical emergency situation could happen and therefore the motion must be managed and controlled in the edge to keep security and privacy. This approach adopts a hierarchical motion model to provide effective handling of objects by using predefined primitive actions (Table 3). It comprises three levels: episodes, primitive actions, and atomic functions [23]. Episodes could be human commands asking the robot to perform a task via a series of primitive actions. The primitive action calls the predefined atomic functions with the atomic functions in the motion descriptor of an SCS, and physically performs them in the motion module. For example, as shown in Table 3, if a user orders "bring $o_i$ to $p_{oi}$," it can consist of a series of primitive actions: "identify $o_i$," "pick up $o_i$," "move the hand to $p_{oi}$," and "place $o_i$." A primitive action, such as "pick up $o_i$," calls the atomic functions: extend($o_i$), grasp($o_i$), and retract(). The motion descriptor of the SCS stores the elements of each level of the hierarchical model, sustains their linkage, and physically responds to the human speech commands.

**Table 3.** An example of action events with the action descriptor.

| Episodes | Primitive Actions | Atomic Functions |
|---|---|---|
| bring $o_i$ to $p_{oi}$ | identify $o_i$ | (search $o_i$ in the object descriptor) |
| | pick up $o_i$ | extend ($o_i$), grasp ($o_i$), retract () |
| | move the hand to $p_{oi}$ | move_hand_to ($p_{oi}$) |
| | place $o_i$ | open_hand ($o_i$), retract () |

## 5. Implementation and Experimental Results

In this paper, the proposed FaaS-FR model was implemented in a mobile robot and tested object recognition, speech recognition, and object handling motion. Figure 6 shows the schematic of the implemented FaaS-FR. The functions of the robot service were distributed with SAP, PSP, and HPP which worked for their own tasks. Figure 7 shows a two-handed mobile robot as a testbed of the edge. The system of the edge was Raspberry Pi 3 using Linux OS (Ubuntu) that has low computing power, and a desktop computer with Windows 10 was used as the fog robot server. Table 4 shows the system specifications of the edge and the fog robot server. In the vision module, there is an Xtion sensor made by ASUS for acquiring RGB-depth (RGB-D) data. For the listening model, there was a microphone on the edge system to capture the human speech, which was transferred to Google Cloud to get the recognized text data [24]. The acquired text data was transferred to a parsing cloud, Link Grammar Parser server [25], to obtain the parsed sentence. Link Grammar Parser adopts Penn Treebank rules for syntactic parsing in which a sentence is segmented with phrases [26].

The test scenario of the FaaS-FR was that a user asks a speech order to the robot to move an object and place it a specific position. For the execution of the order, the robot used the listening module for understanding the human speech, the vision module for 3D object recognition, and the motion module to bring the object. The FaaS-FR based SCS distributed tasks on the edge, a robot fog server, and clouds. Table 5 shows the functions and fog computing types.

For the listening module, speech recognition was executed with the cloud (Google Cloud), but the NLP was done on the fog server (Link Parser server). For speech recognition, the edge first acquired

a human speech and transferred it to the Google Cloud to get the text of the speech. The speech recognition application utilized Google Cloud speech API as a PaaS. The result of speech recognition was transferred to the fog robot server to recognize the meaning of the sentence with syntactic and semantic parsing. The event interpreter requested a motion to execute the order of the human.

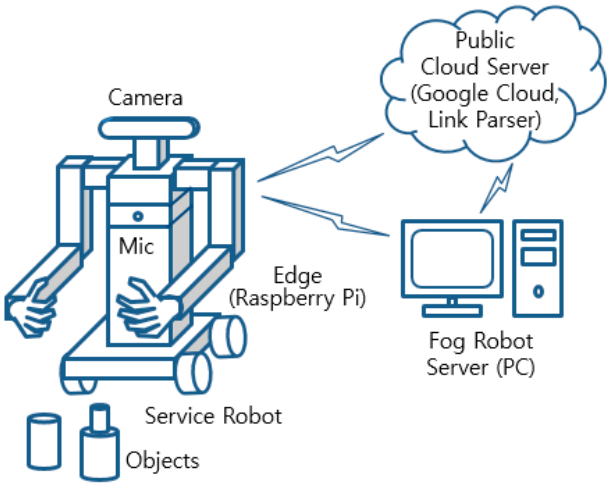



**Figure 6.** The schematic of the FaaS-FR service implementation with an edge of low computing power.

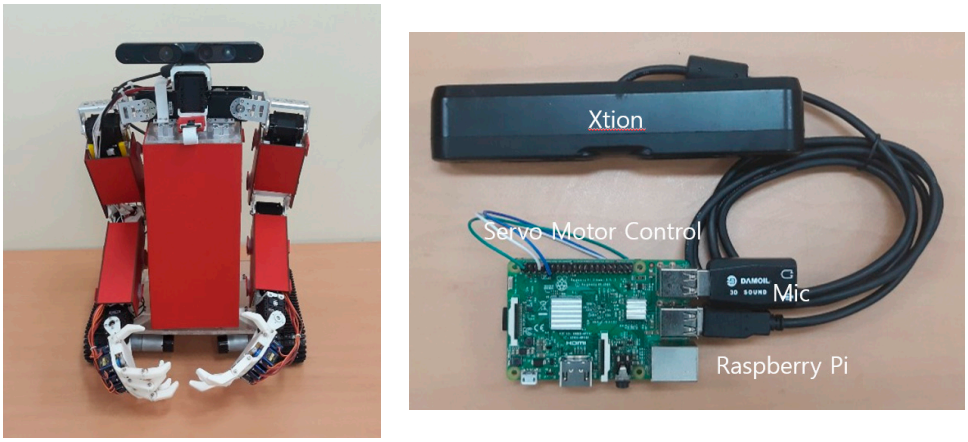
In the case of the vision module, if the scene is changed, the module transfers the captured RGB-D data to the fog server, then the server runs an object recognition application. In this paper, for the object recognition, You Only Look Once (Yolo), a convolution neural network (CNN), was adopted [27]. It produced bounding boxes (BBXs) and labels of the objects in the RGB image. The trained weight files were brought from a cloud (Yolo server), but the object recognition was done on the fog robot server as a SaaS. The depth data were converted to XYZ coordinates for being used for 3D object segmentation, which is done by thresholding on the dot product values of normal vectors of the coordinates as the similarity of the vector orientation. From the segmented 3D data, the real coordinates of the objects were obtained to be handled by the robot.

**Table 4.** The system specification of an edge and a fog robot server.

| Specification | Edge | Fog Robot Server |
|---|---|---|
| System | Raspberry Pi 3 B | Desktop |
| CPU | Quad Cortex A54 | Intel® Core™ i7-770 |
| Clock | 1.2 GHz | 3.6 GHz |
| RAM | 1 GB SDRAM | 8 GB RAM |
| Network | Wifi | LAN |
| Bandwidth | 14.4 Mbit/sec | 94.5 Mbit/sec |

**Table 5.** The functions of the robot and service types.

| Modules | Tasks | Servers | Fog Computation Types | Functionality |
|---|---|---|---|---|
| Listening module | Speech recognition | Google Cloud | Cloud–fog–edge | High performance computation |
| | Semantic parsing | Link Parser | Fog–edge | Computing power |
| Vision module | Object recognition | Yolo | Fog–edge | Computing power |
| Motion module | Object handling | Motion application | Edge | Security and privacy |




(**a**) (**b**)

**Figure 7.** A test bed of FaaS-FR using Raspberry Pi. (**a**) The service robot, (**b**) a Raspberry Pi and perception and behavior modules.

Figure 8 shows the results of the vision module with a fog server. When the module of SCS of the robot captures the RGB-D data by using an Xtion sensor, the SCS executes the vision processing in the fog cloud by transferring the acquired data. The fog server receives the data and processes an object recognition algorithm that needs a relatively higher computing power and then transfers the results of the processing to the cognitive system. Figure 8a,b shows the RGB and depth data of the Xtion sensor. Figure 8c shows the result of object recognition using Yolo. The vision module had the results of BBX and labels of the object were offloaded on the cloud fog server. Figure 8d shows the 3D view of the scene using OpenGL library to get x, y, z coordinates of the cloud points from the acquired RGB-D data, which were processed in the fog robot server. Figure 8e shows the result of 3D object recognition obtained by using 3D segmentation on the BBX area of the object. It provided x, y, z coordinates of the object to be used for the handling of the robot hand

For the action to handle objects, the action module analyzed the meaning of the ordered sentence. The argument of the sentence was linked with the cup in the object descriptor and searched the position and pose of the object. Figure 9 shows the motion executing an episode, "bring the cup to the front of the bottle." The order was an episode and divided with primitive actions: (a) "identify the cup," (b) "pick up the cup," (c) "move the hand to the front of the bottle," and (d) "place the cup." These primitive actions are executed with atomic functions.

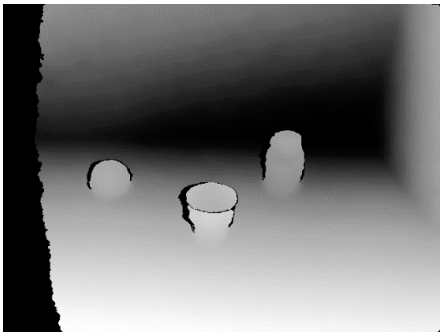
In this paper, FaaS-FR model was tested by comparing the two fog computing types. Table 6 shows two sentences and their speech signals for testing FaaS-FR. Two sentences, "bring the cup" and "bring the cup to the front of the bottle" were tested with the link parser cloud in textual syntactic parsing and Google Cloud for speech recognition. Figure 10 shows the average service times of 20 times of trial with two types of FaaS-FR models to utilize the Google Cloud for speech recognition.

The service time was measured with the duration between the start of packet sending and the end of result receiving in the edge applications. The graph shows that the cloud–fog–edge type is better in the service time. Figure 11 shows link parser service time. We can see the fog–edge type is best to reduce the service time.

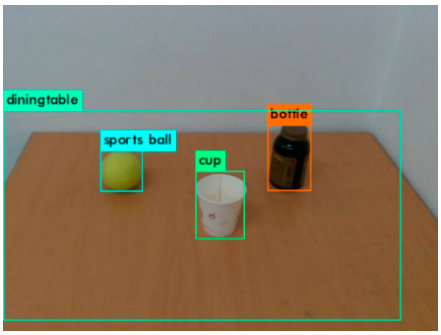
From the results of the two cases, the service time of the services is not proportional to the size of the data. The syntactic parsing shows that the service time performance between cloud–fog–edge and cloud–edge is largely different, but the speech recognition produces a relatively small difference. It could be related to networking delays including processing, queuing, transmission, and propagation delays as well as the computing time of the clouds. Therefore, when one selects a fog computation type, a previous performance test is needed.



(**a**)



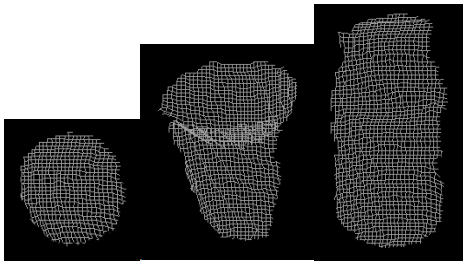
(**b**)




(**c**)

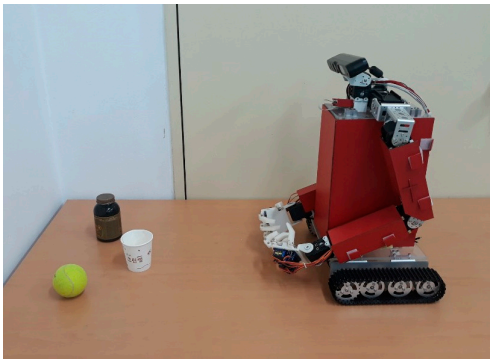


(**d**)

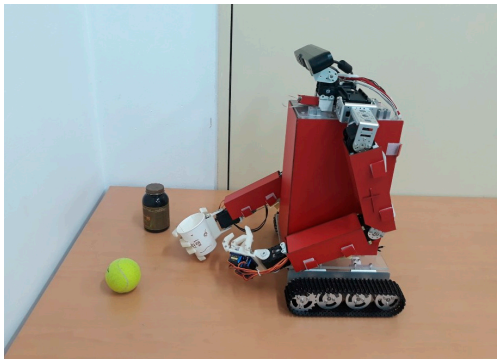


(**e**)

**Figure 8.** The result of object recognition with the fog robot service: (**a**) an RGB image; (**b**) a depth image; (**c**) the result of object recognition with bounding boxes (BBXs) and labels; (**d**) a 3D view of the scene; (**e**) the result of 3D object recognition after 3D segmentation on the BBX areas.

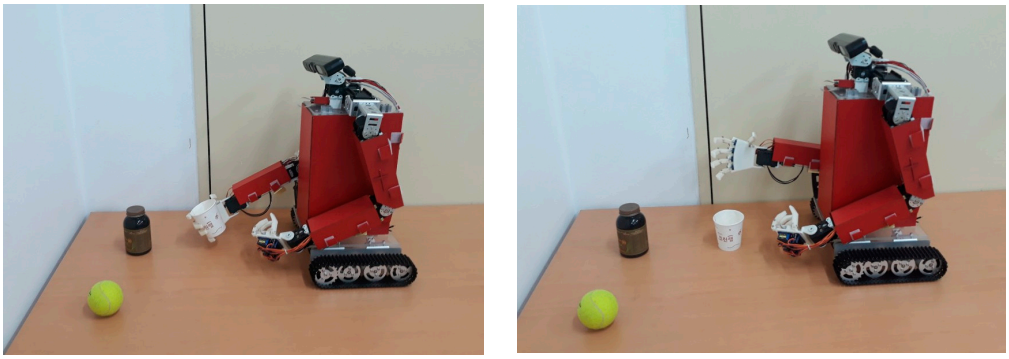


(**a**)



(**b**)

**Figure 9.** *Cont.*



(**c**) (**d**)

**Figure 9.** The motion of the service robot executing "bring the cup to the front of the bottle": (**a**) identify the cup; (**b**) pick up the cup; (**c**) move the hand to the front of the bottle; (**d**) place the cup.

**Table 6.** The sentences and speech signals for testing FaaS-FR.

| | Sentence_1 | Sentence_2 |
|---|---|---|
| Text | "bring the cup" | "bring the cup to the front of the bottle" |
| Time (s) | 1.55 | 2.93 |
| File size (Kbyte) | 135 | 180 |

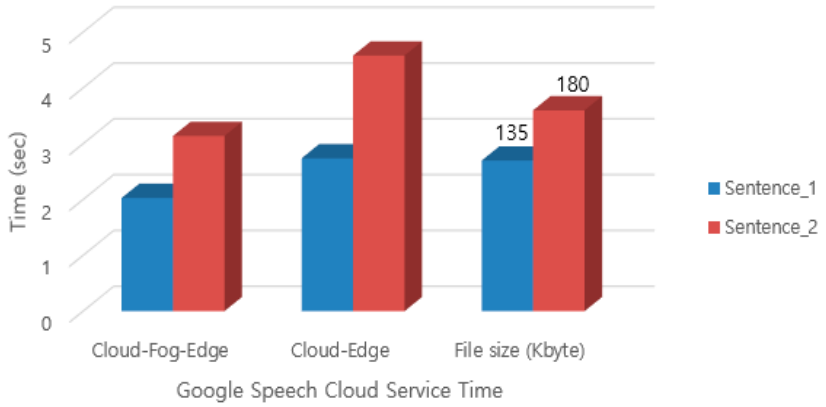



**Figure 10.** Google speech cloud service time according to the fog service model.

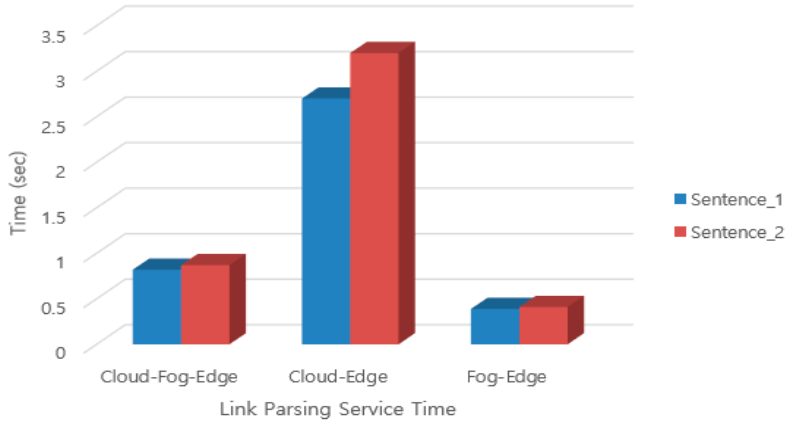



**Figure 11.** Link parser service time according to the fog service model.

## 6. Conclusions

In this paper, a FaaS-FR model for cognitive robots is proposed. The functions of cognitive system are categorized as SAP, PSP, and HPP according to functionality of security, privacy, high performance computation, and needed computing power. The modular functions of SCS of the robot are divided into classes apt to be proper to edges, fog robot servers, and public robot clouds. FaaS-FR was implemented on Raspberry Pi as an edge, and PCs as a fog robot server, and Google Cloud and Link Parser server as robot clouds. From the test of objects handling, the edge system of the robot worked successfully even it had a low cost Raspberry Pi in speech recognition, 3D object recognition, and object handling motion. The test showed that the robot can work more efficiently even in the cases of low specification edges by properly selecting the computation types. The proposed FaaS-FR model can be an alternative selection for low cost but high performance service robots. In the future, the issue of an autonomous selecting of fog computation types needs to be studied to produce the best performance, and even low cost, edges of cognitive robots.

**Funding:** This research was funded by Tongmyong University Research Grants 2016 and the Basic Research Project of the Korea Institute of Geoscience and Mineral Resourcesfunded by the Ministry of Science, ICT and Future Planning of Korea.

**Acknowledgments:** This research was supported by Tongmyong University Research Grants 2016 (2016A017) and the Basic Research Project (Development of Object Recognition and 3D Informatizing in Life Space for Responding to Geological Disaster) of the Korea Institute of Geoscience and Mineral Resources (KIGAM) funded by the Ministry of Science, ICT and Future Planning of Korea.

**Conflicts of Interest:** The author declares no conflict of interest.

## References

1. RoboEarth. Available online: http://roboearth.ethz.ch/ (accessed on 1 September 2019).
2. Ansari, F.Q.; Pal, J.K.; Shukla, J.; Nandi, G.C.; Chakraborty, P. A Cloud Based Robot localization technique. In Proceedings of the international Conference on Contemporary Computing, Noida, India, 6–8 August 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 347–357.
3. Tanwani, A.K.; Mor, N.; Kubiatowicz, J.; Gonzalez, J.E.; Goldberg, K. A Fog Robotics Approach to Deep Robot Learning: Application to Object Recognition and Grasp Planning in Surface Decluttering: ICRA 2019. *arXiv* **2019**, arXiv:1903.09589.
4. Chinchali, S.; Sharma, A.; Harrison, J.; Elhafs, A.; Kang, D.; Pergament, E.; Cidon, E.; Katti, S.; Pavone, M. Network Offloading Policies for Cloud Robotics: A Learning-based Approach, Robotics: Science and Systems 2019, Freiburg im Breisgau. *arXiv* **2019**, arXiv:1902.05703.
5. Gudi, S.; Ojha, S.; Clark, J.; Johnston, B.; Williams, M.-A. Fog robotics: An introduction. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, BC, Canada, 24–28 September 2017.
6. Roy, D. Semiotic Schemas: A Framework for Grounding Language in Action and Perception. *Artif. Intell.* **2005**, *167*, 170–205. [CrossRef]
7. Coradeschi, S.; Saffiotti, A. An introduction to the anchoring problem. *Robot. Auton. Syst.* **2003**, *43*, 85–96. [CrossRef]
8. Ahn, H. A CaaS Model Based on Cloud/IoT Service for Cognitive Robots. In Proceedings of the ICGHIT, Clark, Philippines, 24–26 February 2016; pp. 106–107.
9. Ahn, H. A sentential cognitive system of robots for conversational human–robot interaction. *J. Intell. Fuzzy Syst.* **2018**, *35*, 6047–6059. [CrossRef]
10. Kuffner, J. Cloud Enabled Humanoid Robots. Humanoids 2010 Workshop Talks. Available online: https://www.scribd.com/doc/47486324/Cloud-Enabled-Robots (accessed on 1 September 2019).
11. Guizzo, E. Robots with their heads in the clouds. *IEEE Spectrum.* **2011**, *48*, 16–18. [CrossRef]
12. Mell, P.; Grance, T. *The NIST Definition of Cloud Computing*; NIST Special Publication: Gaithersburg, MD, USA, 2011; Volume 800, p. 7.
13. Mouradian, C.; Errounda, F.Z.; Belqasmi, F.; Glitho, R. An Infrastructure for Robotic Applications as Cloud Computing Services. In Proceedings of the IEEE WF-IoT, Seoul, Korea, 6–8 March 2014; pp. 377–382.

14. Gherardi, L.; Hunziker, D.; Mohanarajah, G. A software product line approach for configuring cloud robotics applications. In Proceedings of the IEEE 7th International Conference on Cloud Computing, Washington, DC, USA, 8–11 December 2014; pp. 745–752.
15. Guizzo, E.; Deyle, T. Robotics trends for 2012. *IEEE Robot. Autom. Mag.* **2012**, *19*, 119–123. [CrossRef]
16. Chen, Y.; Hu, H. Internet of intelligent things and robot as a service. *Simul. Model. Pract. Theory* **2013**, *34*, 159–171. [CrossRef]
17. DeMarinis, N.; Tellex, S.; Kemerlis, V.; Konidaris, G.; Fonseca, R. Scanning the internet for ROS: A view of security in robotics research. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 8514–8521.
18. Fog Robotics. Available online: https://sites.google.com/view/fogrobotics (accessed on 1 September 2019).
19. Tian, N.; Tanwani, A.K.; Chen, J.; Ma, M.; Zhang, R.; Huang, B.; Goldberg, K.; Sojoudi, S. A Fog Robotic System for Dynamic Visual Servoing: ICRA 2019. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20 May 2019.
20. Wang, X.V.; Wang, L.; Mohammed, A.A.; Givehchi, M. Ubiquitous manufacturing system based on Cloud: A robotics application. *Robot. Comput. Integr. Manuf.* **2017**, *45*, 116–125. [CrossRef]
21. Google Cloud. Available online: https://cloud.google.com/ (accessed on 1 September 2019).
22. Stamey, L. IaaS vs. PaaS vs. SaaS Cloud Models. 2017. Available online: http://www.hostingadvice.com/how-to/iaas-vs-paas-vs-saas/ (accessed on 1 September 2019).
23. Ahn, H.; Ko, H. Natural-Language-Based Robot Action Control Using a Hierarchical Behavior Model. *IEIE Trans. Smart Process. Comput.* **2012**, *1*, 192–200.
24. Google Cloud Speech-to-Text. Available online: https://cloud.google.com/speech-to-text/ (accessed on 1 September 2019).
25. Link Grammar. Available online: http://www.link.cs.cmu.edu/link/index.html (accessed on 1 September 2019).
26. Marcus, M.P.; Santorini, B.S.; Marcinkiewicz, M.A. Building a Large Annotated Corpus of English: The Penn Treebank. *Comput. Linguist.* **1993**, *19*, 313–330.
27. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger, CVPR 2017. Available online: https://pjreddie.com/publications/ (accessed on 1 September 2019).




© 2019 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).