Towards Open Interfaces of Smart IoT Cloud Services

Kyoung-Sook Kim · Hirotaka Ogawa
Artificial Intelligence Research Center
National Institute of Advanced Industrial Science and Technology
Koto-ku, Tokyo 135-0064, Japan
E-mail : ks.kim@aist.go.jp

ABSTRACT

With the vision of Internet of Things (IoT), physical world itself is becoming a connected information system on the Internet and cyber world is computing as a physical act to sense and respond to real-world events collaboratively. The systems that tightly interlink the cyber and physical worlds are often referred to as Smart Systems or Cyber-Physical Systems. Smart IoT Clouds aim to provide a cyber-physical infrastructure for utility (pay-as-you-go) computing to easily and rapidly build, modify and provision auto-scale smart systems that continuously monitor and collect data about real-world events and automatically control their environment. Developing specifications for service interoperability is critical to enable to achieve this vision. In this paper, we bring an issue to extend Open Cloud Computing Interface for uniform, interoperable interfaces for Smart IoT Cloud Services to access services and build a smart system through orchestrating the cloud services.

Keyword

Cloud Computing, Smart Systems, Software-defined Networking, Open Standards, Open APIs

Ⅰ. Introduction

Today, there is a vision of information and communication technologies providing ubiquitous connectivity of physical and virtual things (including human beings) as “a global infrastructure for the information society, enabling advanced services [1].” More than 26 billion devices will be connected to the Internet by 2020, (according to a Gartner report).

In order to realize large-scale smart systems based on IoT(Smart IoT Systems), providing the cyber-physical components as cloud computing services can address many of the challenges in scalability, reliability, security, timeliness, domain-specific usability, and heterogeneous composability. Cloud computing can support common service/application platforms providing generic enabling capabilities, such as device and service management, security and privacy, and data sharing. Applying the characteristics of cloud computing to Smart IoT Systems, any resource (sensor, actuator, data source, computation resource) is available as a service and its service can be provisioned on demand. Cloud-based services allow the computational and physical resources of smart system to be used in multiple systems providing more efficient use of those resources. The Smart IoT Cloud has the flexibility to change and adapt providing robustness and meeting users changing needs. In this paper, we propose an extension of Open Cloud Computing Interface (OCCI) for developing standard interfaces of a heterogeneous and cross-domain Smart IoT Clouds.

The rest of the paper is structured as follows: Section 2 discusses the basic service types of Smart IoT Clouds with the concept of Cyber-Physical Cloud Computing. Section 3 describes a model of services for Smart IoT Clouds based on the Open Cloud Computing

Interface (OCCI) model. Finally, the conclusion is described in Section 4 with a future plan.

Ⅱ. SERVICE MODELS SMART IOT CLOUDS

A conceptual reference architecture of Cyber-Physical Cloud Coputing (CPCC) as a concept of Smart IoT Clouds is described in [2]. The authors emphasized the characteristics of cloud computing (on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service) to cyber-physical systems. In this architecture, there are four basic types of service (sensor, processor, actuator, and stored data) and orchestration component as shown in Figure 1. The orchestration component composes and conducts these individual services forming a smart system/ application. Here, we refer to this conceptual reference architecture to define smart IoT cloud services with the composability and compositionality of heterogeneous and cross-domain IoT resources.

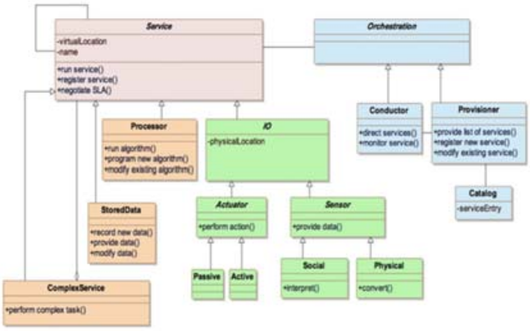

Figure 1. Conceptual Reference Architecture[2]

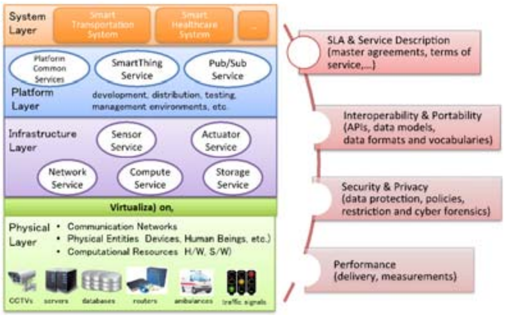
System Layer
Smart Transportation System
Smart Healthcare System
...
Platform Common Services
SmartThing Service
Pub/Sub Service
Platform Layer
development, distribution, testing, management environments, etc
Infrastructure Layer
Sensor Service
Actuator Service
Network Service
Compute Service
Storage Service
Virtualiza) on,
Physical Layer
• Communication Networks
• Physical Entities Devices, Human Beings, etc.)
• Computational Resources H/W, S/W)
CCTVs
servers
databases
routers
ambulances
traffic signals
SLA & Service Description (master agreements, terms of service,...)
Interoperability & Portability (APIs, data models, data formats and vocabularies)
Security & Privacy (data protection, policies, restriction and cyber forensics)
Performance (delivery, measurements)

Figure 2. Service Models and Requirements in Smart IoT Clouds

In general, cloud service models have three different approaches: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS). IaaS provides essential IT resources like computers, storages, and networks abstracted by virtualization. PaaS comprises the environment for developing cloud applications for developers, and SaaS provides complete applications to an end user. As mentioned above, Smart IoT Cloud aims to provide a generic platform of smart systems composing computational and physical components. Thus, IaaS should consider physical resources such as devices, embedded systems, things and even human beings as essential resources. We define basic smart IoT resources in a cloud stack as shown in Figure 2.

- Compute Service: provides computing environment (a virtual server) to host services, e.g., Amazon EC2 and Rackspace are typical processor services.
- Storage Service: delivers the storage capacity for managing, recovering and archiving data, e.g., Google Cloud Storage and Amazon S3 provide storage for the Internet and web-based interfaces to store and retrieve any amount of data.
- Network Service: provides networking capabilities to easily deploy user-defined routing/forwarding and multicast protocols as discussed in [5]. Further, it can process packets on-path such as data aggregation, catching and redundancy elimination to improve end-to-end throughput and reduce network traffic with create isolated overlay networks.
- Sensor/Actuator Service: provides physical sensors and actuators as virtual sensing/ controlling services that can be provisioned on demand. The management, abstraction and virtualization of physical devices in [3] describes a similar concept.

Note that these services are delivered "as a Cloud Service" matching the cloud capabilities to users. Through virtualization, a user can interact with a virtual interface that then forwards the requests to the actual resources in the physical layer.

In the physical layer, various types of smart objects are inter-connected via Internet. The smart objects are small computers or devices with constraints on energy, communication, memory, computing power, etc. [4]. A main topic of IoT is to create communication networks of smart objects with IP addresses that can directly connect to other Internet resources. In Smart IoT Clouds, a smart object is not a small computer or simple sensor/actuator device but it is a virtualized service to collect, store, and act on any data from any connected device or cloud service. Currently a number of IoT platforms, such as Xively, Thingspeak, and ARTIK Cloud, provide easy-to-use APIs and tools that help developers to create a smart system based on smart objects. However, their services are supplied with their own interfaces, cloud customers would need to tailor each system to operate with each vendor interface, an effort would therefore be required to build or modify a virtual system. Thus, a simple and standard-based interface allowing heterogeneous services to be able to cooperate or sometimes compete together on an application is needed. Here, we define two additional middleware services along with the basic services to compose, deploy, host, manage, monitor, and control other services and applications:

- SmartThing Service: represents a cloud service as an abstraction unit of cloud platform resources (hardware and software), such as sensor, actuator, database, web container, programming language, and orchestration, to develop new applications or services in the cloud. A smart-thing service can be scaled depending on internal capabilities. For instance, recent cloud robots are an example of smart things. It should offer development tools to sense, process, share, and take adaptive actions according to surrounding situations by cloud services. Even though there are no existing standards for this, a few studies [5] and [6] show the interest in designing standards supporting smart things.
- Pub/Sub Service: provides real-time communication capabilities with various protocols between smart things based on IoT networking services regardless of proxy servers and firewalls, e.g., PubNub and Pusher are examples offering simple APIs to send real-time messages between applications. Common communication interfaces and message formats are also an important issue in the platform understructure for secure interactions of loosely coupled smart objects with a scale.

III. A POSSIBLE EXTENSION OF OCCI

In this study, we focus on common APIs to access cloud resources for the interoperability of Smart IoT IaaS through HTTP-based communication by extending existing standards as a starting point of our work. The standardization of platforms boundary interfaces and APIs is fundamental to implement interoperable cloud technologies. Different standards will apply to different kind of cloud services and different integration strategies. Thus, we need to establish appropriate standards for the Smart IoT environment.

Many existing standards for Web services, such as WS-* specifications or RESTful interfaces, play basic structures to support broad accessibility and interoperability of cloud services. Further, cloud-specific standards have been proposed in industry and academic communities, such as the Open Virtualization Format (OVF), the Cloud Data Management Interface (CDMI), the Open Cloud Computing Interface (OCCI), the Cloud Infrastructure Management Interface (CIMI), or the Topology and Orchestration Specification for Cloud Applications (TOSCA). Among them, we look at two specifications: DMFT's CIMI [7] and OGF's OCCI [8] for cloud infrastructure managements. These specifications show goals similar to ours. Both of them describe interface models and how to map and render the models using RESTful HTTP-based protocol. Even though the CIMI specification covers core functionality to configure, deploy, maintain, and monitor infrastructure resources and seems more mature and more detailed, we decided to focus on the OCCI model and protocol first for our extension. The OCCI protocol gave us a simpler way with a higher degree of extensibility from its core meta-model to define our own resources and APIs, i.e. a virtual sensor, actuator and controlled network.

Figure 3 presents our proposed extension (green color) of the OCCI model (yellow color) for a cloud-based smart IoT infrastructure. We here assume every communication is based on Internet Protocol (IP) based networks because smart objects will be inter-connected with each other or to the Internet by IP as shown in [4]. In the model, we add two Mixin types: SDNController and SDNSwitch. They are added to the network resource instance at creation-time and/or run-time with a software-defined networking (SDN) approach. SDN controllers offer northbound APIs to support application developers for orchestration of networks and services and southbound APIs

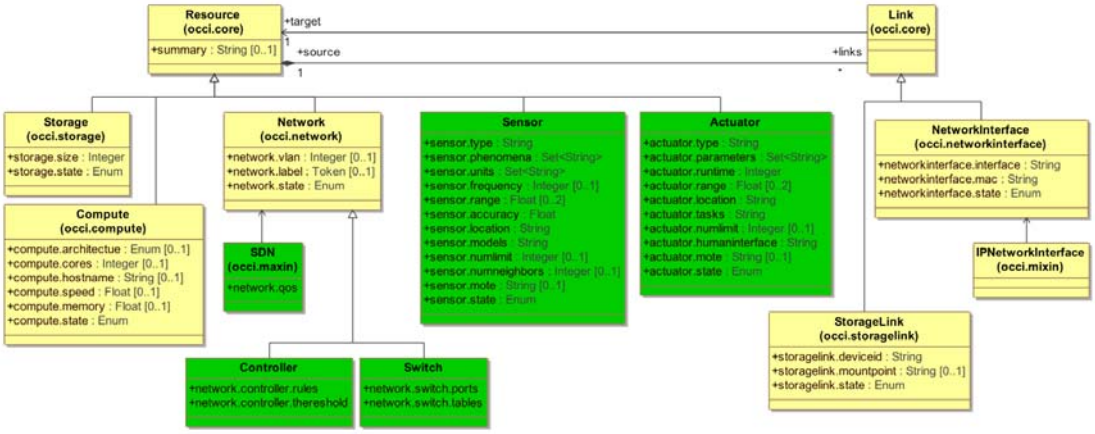
Resource
(occi.core)
+summary : String [0..1]
+target
1
+source
1
Link
(occi.core)
+links
*
Storage
(occi.storage)
+storage.size : Integer
+storage.state : Enum
Compute
(occi.compute)
+compute.architectue : Enum [0..1]
+compute.cores : Integer [0..1]
+compute.hostname : String [0..1]
+compute.speed : Float [0..1]
+compute.memory : Float [0..1]
+compute.state : Enum
Network
(occi.network)
+network.vlan : Integer [0..1]
+network.label : Token [0..1]
+network.state : Enum
SDN
(occi.maxin)
+network.qos
Controller
+network.controller.rules
+network.controller.thereshold
Switch
+network.switch.ports
+network.switch.tables
Sensor
+sensor.type : String
+sensor.phenomena : Set<String>
+sensor.units : Set<String>
+sensor.frequency : Integer [0..1]
+sensor.range : Float [0..2]
+sensor.accuracy : Float
+sensor.location : String
+sensor.models : String
+sensor.numlimit : Integer [0..1]
+sensor.numneighbors : Integer [0..1]
+sensor.mote : String [0..1]
+sensor.state : Enum
Actuator
+actuator.type : String
+actuator.parameters : Set<String>
+actuator.runtime : Integer
+actuator.range : Float [0..2]
+actuator.location : String
+actuator.tasks : String
+actuator.numlimit : Integer [0..1]
+actuator.humaninterface : String
+actuator.mote : String [0..1]
+actuator.state : Enum
NetworkInterface
(occi.networkinterface)
+networkinterface.interface : String
+networkinterface.mac : String
+networkinterface.state : Enum
IPNetworkInterface
(occi.mixin)
StorageLink
(occi.storagelink)
+storagelink.deviceid : String
+storagelink.mountpoint : String [0..1]
+storagelink.state : Enum

Figure 3. An Extension of OCCI for Smart IoT Infrastructures

for the communication between the controller and the network resources. Cloud providers have started to employ SDN technologies to manage traffic loads and specific types of flows for cloud tenants. Currently, many different RESTful northbound APIs already exist and they are still evolving based on "open" development environments. However, we need a standard for interoperable northbound APIs to improve benefits of cloud computing and a new working group was started in Open Networking Foundation (ONF11). This work could be a possible approach for its design. In addition, there are Actions defined for the two Resources Sensor and Actuator. The Action element defines an invocable operation of Entity, which is an abstract representation of the Resource and Link elements. For instance, we can define the actions ‘shutdown’, ’start’, and ’restart’ for the Sensor and Actuator resource. The abstraction and virtualization allow cloud users only to consider the logical elements for their service or application, such as sampling frequency, type of sensor and observation, deployment place, data process model, etc.

Ⅳ. CONCLUSIONS AND FUTURE WORK

Extending existing cloud interface specifications to support Smart IoT Cloud functionality minimizes interoperability issues between existing cloud infrastructure and cyber and physical resources being developed. Similar extensions to the OCCI specification described in this paper for the basic smart IoT infrastructure services can quickly bring a new smart system based on existing IoT resources. For our future work, combining the extended interface spec, with standardized service capability descriptions and implementing a robust orchestration component will be a high priority.

References

[1] Recommendation ITU-T Y.2060 (2012), Overview of Internet of Things (Pre-published).

[2] E. Simmon, K.-S. Kim, E. Subrahmanian, R. Lee, F. de Valux, Y. Murakami, K. Zettsu, and R. D. Sriram, A Vision of Cyber-Physical Cloud Computing for Smart Networked Systems, In NISTIR 7951 (July 2013)

[3] S. Distefano, G. Merlino, A. Puliafito, A. Vecchio, A hypervisor for infrastructure-enabled sensing Clouds, In: Proc. IEEE International Conference on Communications Workshops, pp.1362–1366 (2013)

[4] The IPSO Alliance, http://www.ipso-alliance.org

[5] M. Blackstock and R. Lea, IoT mashups with the WoTKit, In: Proc. International Conference on the Internet of Things, pp.159–166 (2012)

[6] M. Koster and S. McNeil, IoT Toolkit and the Smart Object API (April 2013)

[7] Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol (DSP0263)

[8] Open Cloud Computing Interface : Core (GFD.183)