Smart City Interoperability

— Fed4IoT Japan-Europe Joint Research —

Hidenori Nakazato
Professor
Department of Communications and Computer Engineering
Waseda University

1. Introduction

With the development of ICT technology, all types of object can now be connected to networks. As implementation of 5th Generation mobile communications network technology (5G) approaches, machine-to-machine (M2M) communication is gaining prominence as one of its applications, connecting computers, sensors, actuators and other devices. One target in this area is called massive machine-type communication (mMTC). mMTC will be used to implement IoT systems, connecting large numbers of devices and computers to the network, and controlling them cooperatively.

IoT systems are anticipated for a wide range of applications, including Smart Cities, involving initiatives to optimize municipal functions and improve convenience. According to an interim summary report from the Ministry of Land, Infrastructure, Transport and Tourism titled, "Toward implementation of Smart Cities"[1], Smart Cities are defined as, "A city or region that utilizes ICT and other new technologies to manage (plan, organize, manage, operate, etc.) various issues it is faced with and to perform overall optimization in a sustainable fashion." Smart City initiatives started to develop in around 2010 throughout the world, with practical R&D and demonstrations. Until recently, these initiatives were applied and operated to solve problems in a particular field. Examples are Smart Grids, which optimize supply and demand in the electricity distribution system, and efforts to optimize transportation system operation.

Following IoT systems in separate fields, R&D toward the next level of Smart Cities now requires these individual IoT systems to be linked together, to optimize and increase efficiencies in the city as a whole. There are already many initiatives to link IoT systems throughout the world. However, it is not a simple matter to link IoT systems from different fields and provide services that span different application fields because these IoT systems were built as separate systems and use different internal representations, terminology and data models. Linking of IoT systems has not yet progressed beyond tests and demonstrations, and implementations with a business model and capable of sustained operation are still in the future.

Existing IoT systems have been integrated vertically as separate systems, and built for an individual IoT service. It is difficult to utilize IoT devices that are already installed for other purposes. To develop and operate any new IoT service from the ground up requires installing IoT devices at the low level, connecting them to the network, establishing the IoT platform and developing the service. The high start-up costs are hurdles that must be overcome to execute a new IoT service and to promote Smart Cities and IoT services.

To deal with these two issues—linking IoT systems and the high startup costs—we are running a joint Japan-Europe Research project under the Ministry of Internal Affairs and Communications, Strategic Information and Communications R&D Promotion Programme (SCOPE), called "Federating IoT and cloud infrastructures to provide scalable and interoperable Smart Cities applications, by introducing novel IoT virtualization technologies (Fed4IoT)"[2]. This article introduces initiatives of the Fed4IoT joint Japan-Europe research project toward realizing Smart City interoperability.

2. Fed4IoT Development Plan/Policy

The aim of Fed4IoT is to build a virtual IoT-cloud platform that will provide interoperability among separate IoT systems (the IoT service domain), using various existing IoT platforms, including oneM2M[3], FIWARE[4], and ETSI-MEC for 5G[5]. Fed4IoT performs sharing on the following three levels, to link IoT service domains and reduce start-up costs (Figure 1).

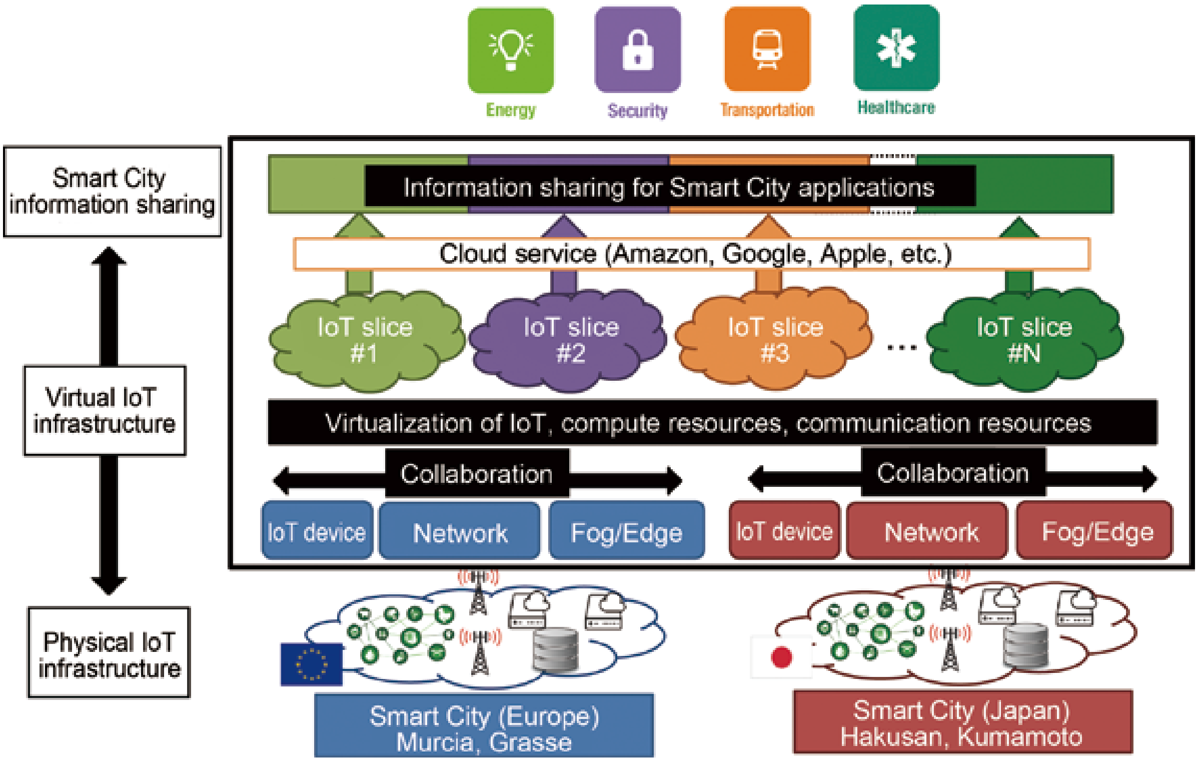
- Data Level
- Platform Level
- Device Level

For interoperability in the IoT service domain, one issue is that different terminology and data models are used to represent IoT data in each IoT service domain. For interoperability, individual IoT services must first share the terminology and data models as metadata. This will enable the Fed4IoT collaboration platform to convert data from individual IoT service domains to its own internal representation, referring to this metadata. The administrator for each IoT service domain is responsible for managing the data handled by the service for that domain. The administrator controls publication of the IoT service domain information to the platform when the domain is linked to the Fed4IoT collaboration platform. Access control information related to publication of this information must also be shared as metadata.

Sharing at the platform level involves connecting each IoT service domain, and an adapter is needed to connect each IoT domain to the Fed4IoT collaboration platform. These adaptors will be described below where device level sharing is described.

Two methods for communication between the multiple IoT service domains and the Fed4IoT collaboration platform are being considered. The first is publish/subscribe communication, and the second is content-centric or information-centric

Figure 1: Overview of Fed4IoT R&D issues


Energy
Security
Transportation
Healthcare
Smart City information sharing
Virtual IoT infrastructure
Physical IoT infrastructure
Information sharing for Smart City applications
Cloud service (Amazon, Google, Apple, etc.)
IoT slice #1
IoT slice #2
IoT slice #3
…
IoT slice #N
Virtualization of IoT, compute resources, communication resources
Collaboration
Collaboration
IoT device
Network
Fog/Edge
IoT device
Network
Fog/Edge
Smart City (Europe) Murcia, Grasse
Smart City (Japan) Hakusan, Kumamoto

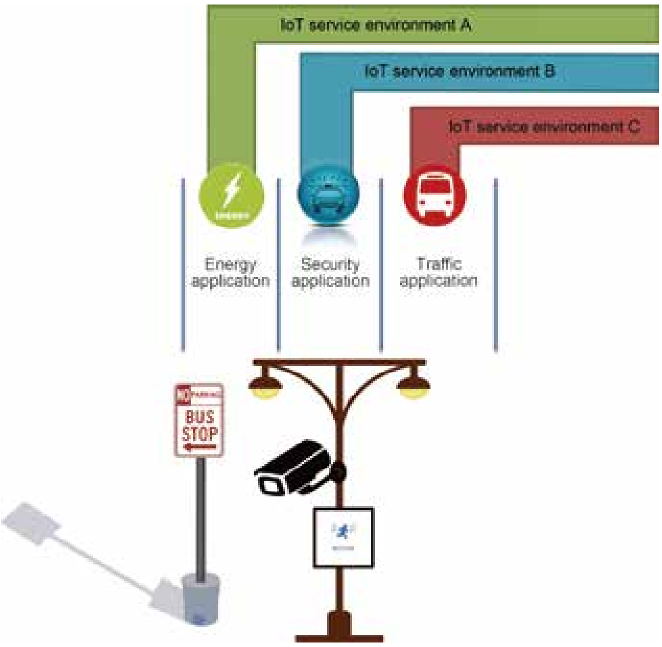
networking (ICN). Both communication methods use abstract identifiers, such as keywords or content names, and are suitable for implementing communication specifying various IoT devices and services.

Data from IoT devices is stored in a repository, such as the Data Management & Repository, which is a Common Service Function of oneM2M, or the Context Broker of FIWARE. Most IoT devices are normally in a sleep state to conserve power. They wake up to collect data, send it to a repository, and then return to a sleep state. This is referred to as "push" communication from the IoT device. IoT applications such as fault detection are required to respond immediately to data from the IoT device. A publish-subscribe model must be used to send and receive data to implement this sort of coordination. On the other hand, it would be wasteful to constantly send values read from the IoT device for applications that require low latency. Request/response form of communication, as with ICN, is appropriate.

IoT devices are supplied by various vendors, and each operates in an IoT service domain. The various IoT devices in each IoT service domain can be used and shared through the unified Fed4IoT collaboration platform. In doing so, IoT devices in each IoT service domain are presented to IoT applications on the Fed4IoT collaboration platform as virtual IoT devices through the repositories of each IoT service domain, rather than present them directly as physical devices. By going through this virtual IoT device software layer, access can be synchronized and data can be distributed among the various IoT applications sharing an IoT device. The software structure implementing virtual IoT devices is also able to perform adapter functions for connecting the various IoT service domains to the Fed4IoT collaboration platform.

IoT device sharing through virtualization is useful in cases such as the one shown in Figure 2. In the figure, the electric company has installed smart light bulbs and person sensors in street lights to control the illumination according to the presence of people, and these are connected to the cloud. A security company has also installed surveillance cameras on the lamp posts to monitor people in the street, and these are also connected to the cloud. The bus company could then use the surveillance camera to monitor the wait times and the number of people waiting at the bus stop next to the lamp post, in order to optimize bus

Figure 2: IoT device sharing

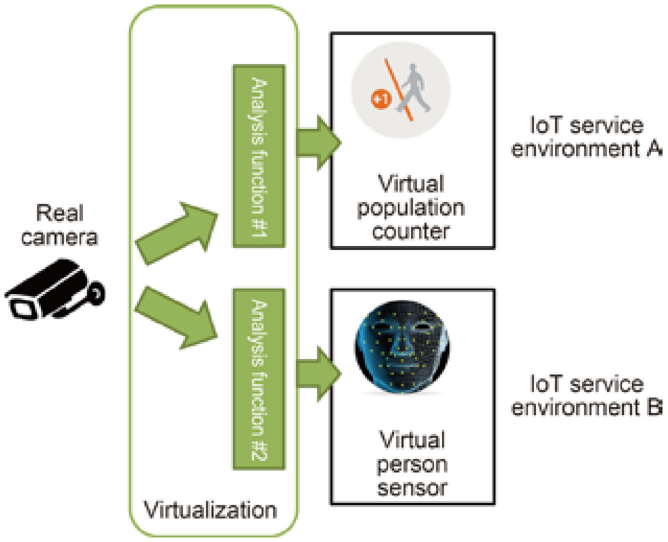

IoT service environment A
IoT service environment B
IoT service environment C
Energy application
Security application
Traffic application
BUS STOP

operations. This sharing of devices could help reduce the start-up and operating costs of providing various services.

It is also possible to configure multiple virtual IoT devices with different functions from a single real IoT device (Figure 3). In this figure, virtual IoT devices with different functions, person-counter and person-detector, are derived from a single surveillance camera.

To promote sharing of IoT devices through virtualization, the Fed4IoT project considers it important to make these multiple IoT service environments (the run-time environment for IoT applications) independent of each other, without interference among them.

Figure 3: Providing multiple functions with virtual IoT device



3. System architecture

A conceptual diagram of the Fed4IoT virtual IoT-cloud collaboration platform, called “VirIoT,” is shown in Figure 4. Various IoT service domains are shown on the left side of the figure. These IoT service domains are built according to oneM2M or FIWARE standards. They contain repositories that store IoT data, and provide external access to some of this data.

IoT data from various IoT service domains along with open data and other data sources comprise the “Root Data Domain,” and this Root Data Domain forms the data set for VirIoT operation. The “Virtual Things” are virtual IoT devices, which are implemented by “ThingVisor” software.

VirIoT provides virtual IoT systems called “Virtual Silos” to IoT service providers, or “tenants.” A Virtual Silo is an environment provided to a specific tenant, providing an independent execution environment, which does not interfere with any other Virtual Silos, for an IoT application. A tenant can use a virtual IoT device, a Virtual Thing, by selecting the ThingVisor that implements the Virtual Thing and adding it to their own Virtual Silo.

Each Virtual Silo also has a repository for managing IoT data to be used within the Virtual Silo, similar to the linked IoT service domains, and the data in this repository can be opened to and shared with other Virtual Silos. Virtual Silos can also be connected with real IoT devices.

In the example in Figure 4, tenant Bob wants to build an irrigation system at his home. Bob has experience using FIWARE’s Orion broker (repository), so he creates a virtual silo incorporating Orion broker. He connects his own temperature sensors and sprinkler devices to the Virtual Silo he has created, and since he does not have a humidity sensor, he creates a virtual

Figure 4: Fed4IoT virtual IoT platform: VirIoT

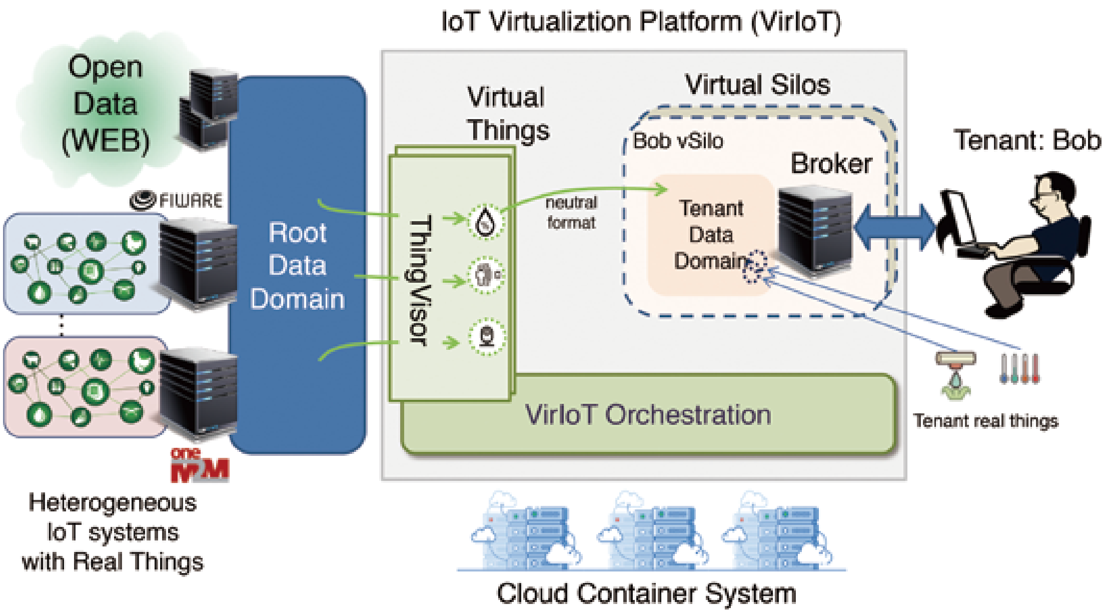
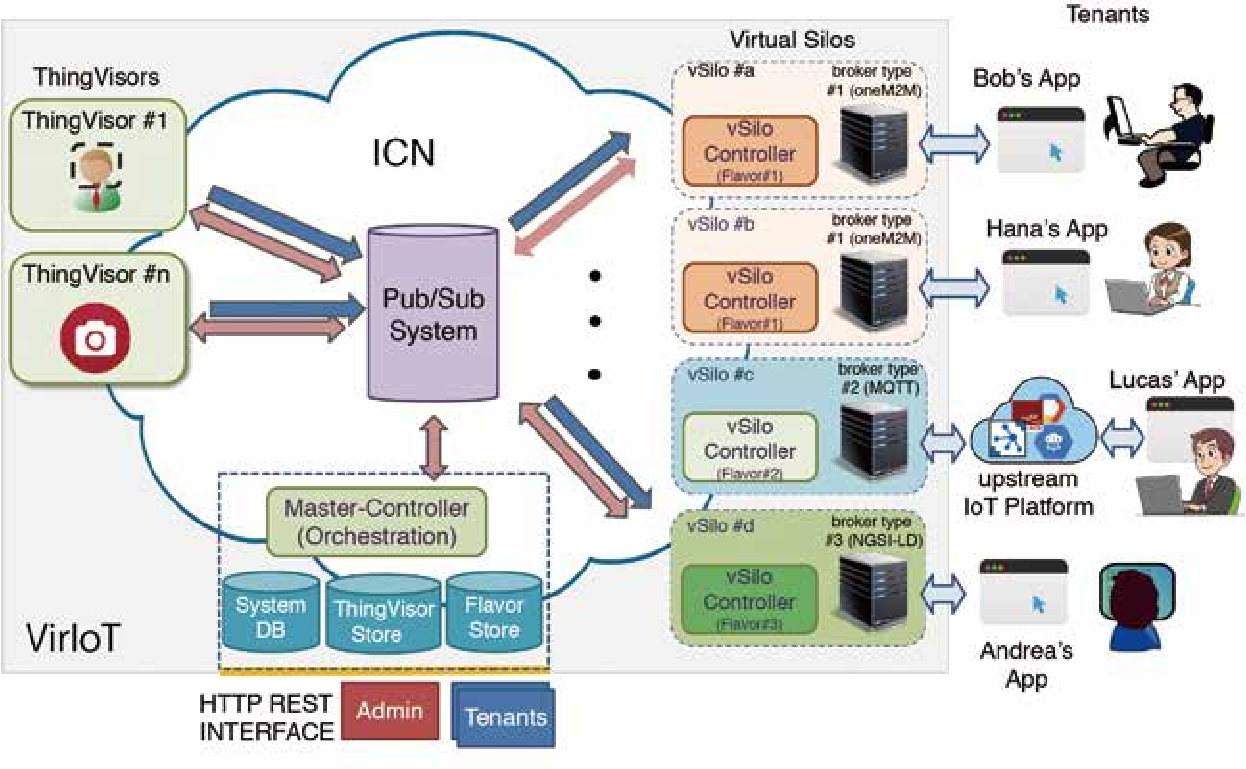


Figure 5: Fed4IoT system architecture



one by borrowing one from another IoT sensor domain. This enables him to build an IoT application to irrigate automatically.

The Fed4IoT architecture is shown in Figure 5. It uses a micro-service design method, and components, including ThingVisor and VirtualSilo, are independent subsystems with their own network interfaces. This architecture enables IoT service developers to develop their components independently. These independent components are currently being created as Linux containers. By making each Virtual Silo an independent Linux container, Virtual Silos realize IoT application execution environments that cannot interfere with each other.

As mentioned earlier, the communication environment within VirIoT provides communication using both publisher/subscriber model and content-centric networking. In particular, implementations using the content-centric networking can use service function chaining to link ThingVisors. Thus, the environment is designed to enable tenants to easily build the IoT service they desire by linking various service functions that have been prepared earlier as ThingVisors. We hope to promote development and spread of IoT services by providing an environment that makes IoT services easy to develop. For example, in smart home environments, many different IoT devices are installed in each home. How these are controlled can be expected to differ depending on the environment they are in, and home owners will want to create their own IoT services. As such, a simple environment for building IoT service is an essential function of an IoT system.

4. Conclusion

In this article, we have discussed how it is necessary to link IoT systems built for separate fields in order to implement overall optimization in Smart Cities. We have also introduced Fed4IoT, a joint research project between Japan and Europe that is working to enable such linking. There is much R&D being done to link IoT systems, and we expect that user-friendly IoT systems supporting highly efficient services will be implemented in the near future.

References
[1] Ministry of Land, Infrastructure and Transport. Toward realization of Smart Cities [Intermediate summary], https://www.mlit.go.jp/report/press/toshi07_hh_000126.html, Aug. 2018.
[2] Fed4IoT Web page, https://fed4iot.org.
[3] oneM2M Web page, http://www.onem2m.org
[4] FIWARE Web page, https://www.fiware.org.
[5] Multi-access edge computing (MEC), https://www.etsi.org/technologies/multi-access-edgecomputing.