一般社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report

[Special Invited Talk] Introduction to Fed4IoT architecture and its related technologies for Smart city

Andrea Detti

CNIT, Elect. Eng. Dept., University of Rome "Tor Vergata", Italy
E-mail: andrea.detti@uniroma2.it

Abstract In this paper we present the Cloud of Things architecture conceived by the Fed4IoT EU-JP project. Although there are many IoT cloud platforms, our Cloud of Things virtualizes and offers as-a-service what is still real, i.e. the IoT infrastructure, while at the same time addressing interoperability and expenditure issues that small stakeholders encounter when they decide to deploy large-scale services IoT applications such as Smart city one.

Keywords Cloud, Virtualization, IoT, interoperability

1. Introduction

Modern cities need to evolve and become structured and interconnected ecosystems in which thousands of IoT components belonging to systems of different sectors (energy, health, mobility, buildings, water management, lighting, waste management, environment, etc.) work together. The resulting huge amount of heterogeneous information, its correlation and intelligent processing is the enabling factor for decision making, services and cross-cutting applications that provide citizens with the feeling of living in an intelligent environment. The implementation of such pervasive and large-scale systems of heterogeneous IoT devices and data poses problems of interoperability and expense.

Interoperability problems arise because there is as yet no consensus on what, if any, and what is the scope of the final IoT standard for devices and/or data, allowing interoperability between different providers. Expenditure problems arise in large-scale IoT implementations, such as smart cities, because a valuable amount of capital (CAPEX) and operating costs (OPEX) is required to implement the necessary IoT infrastructure.

In fact, IoT application developers usually need to install their own infrastructure, or Silo, made of things, agents and (possibly) information brokers, as shown in Fig.1. Things are heterogeneous sensors or actuators that send or receive information from applications. Agents are adapters that manages the interoperability among things and applications (or brokers) with respect to different aspects including programming/network interfaces and information models. The broker is an optional component (or distributed system) that manages the lifecycle of the information of the things of the Silo, so that applications have a single point of access to the whole information set.

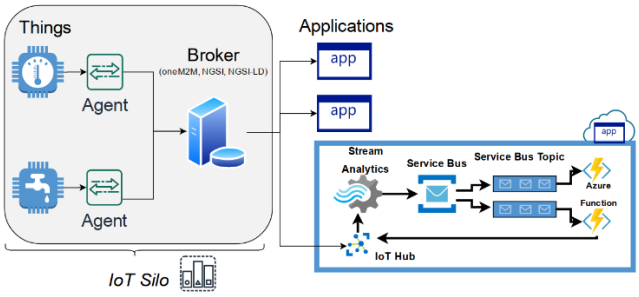

Fig. 1 Example of IoT Silo and applications

Many standardization efforts focus on the API and the information model used by these brokers. OneM2M [4] and NGSI [2] are two very popular standards, and the latter is evolving towards NGSI-LD [3].

The applications can be built either from scratch or can exploits upstream IoT cloud platforms offered by many providers including Amazon, Microsoft and Google. These cloud services provide the means to simply compose processing pipelines made up of analytics modules, functions, information hubs, etc. Fig. 1 shows an example of this processing pipeline based on services provided by Microsoft Azure.

Fed4IoT is a joint Europe-Japan project that addresses the interoperability and expenditure issues mentioned above in a single **Cloud of Things** platform, called VirIoT. VirIoT extends the scope of cloud services to what is not yet virtualized - the Silo (Fig.1). VirIoT provides users with Virtual Silo (vSilos): isolated and secure environments made of Virtual Things (vThings) whose data is accessible through standard information brokers offered as a service. IoT application developer can simply rent a Virtual Silo with the necessary Virtual Things and select the IoT standard they prefer to access the data. VirIoT takes care of solving interoperability between different standards and virtualizing the IoT infrastructure.

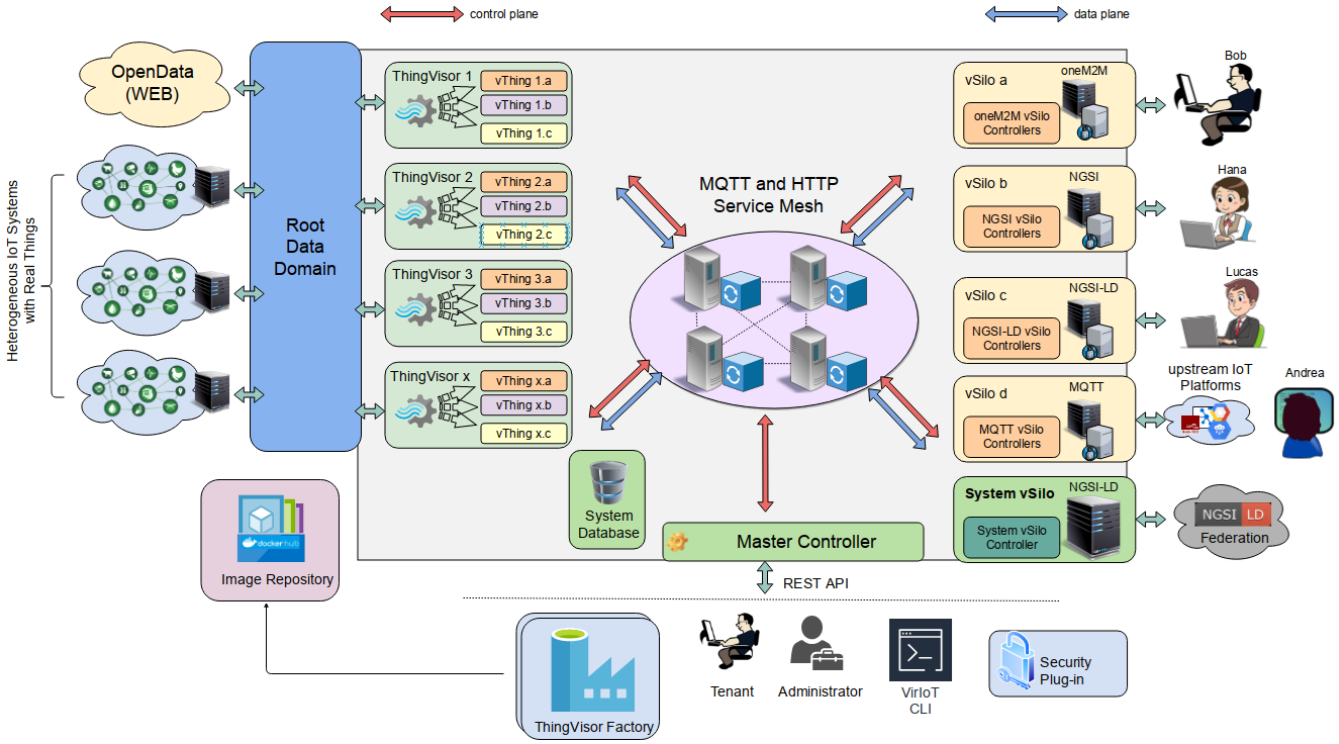

Fig. 2 VirIoT system architecture

2. VirIoT platform

Fig. 2 shows the architecture of the VirIoT platform [1]. *ThingVisors* are components that interact with real things (sensors or actuators) of heterogeneous IoT systems and produce data related to vThings, which are emulations of real things. For example, a person counter based on the processing of video streams from a real camera can be a vThing. Virtual Things can also be actuators, like lamps or door locks; in this case the supporting ThingVisor acts as a proxy between the system user and the real actuator.

The vThings can produce or consume telemetry data (e.g. sensor measurements or actuation commands) and large HTTP contents (e.g. video streams). Internally, telemetry data is compliant with the NGSI-LD information model, deemed to behave as a *neutral-format* that can be simply converted in other standard data formats. The ThingVisor solves the interoperability issues between external data format and the internal NGSI-LD one. Telemetry data is shared with interested vSilos through an internal service mesh consisting of a cluster of MQTT servers. The HTTP data is instead shared through another service mesh whose nodes are HTTP proxies. Both service meshes provide data multicasting and caching to reduce bandwidth consumption and latency.

Users can create vSilos in which vThings can be added on-demand. The telemetry data of included vThings is exposed to the user through an IoT broker server of user's choice, e.g. a oneM2M, NGSI, or NGSI-LD server. Within the vSilo, an IoT controller performs the translation of the data from internal NGSI-LD format into the format of the IoT broker, and manages the operations of the control plane (e.g. add/remove vThings). Therefore, the IoT controller solves the interoperability between the internal format and the format of the vSilo IoT broker. The HTTP data of the vThings of the vSilo is exposed to the user through an internal HTTP server without any format translation. A special vSilo, called *System vSilo*, is used to federate the platform with external platforms based on NGSI-LD technology, thus enabling the platform to be part of a wider IoT ecosystem.

ThingVisors and vSilos are devised as autonomous micro-services (Kubernetes PODs) whose deployment is controlled by a *Master Controller*, which exploits an underlying distributed Kubernetes system and supports edge computing in the sense that the user can decide in which zone to deploy a specific component e.g. to reduce access latency. A *System Database* maintains the status of the system components. The development of ThingVisors can take advantage of additional tools, called ThingVisor Factories. The basic security of the platform is based on JSON Web Token (JWT) technology but can be improved with specific plug-in.

Acknowledgments

This work is supported in part by H2020 EU-JP Fed4IoT project (www.fed4iot.org, EU contract 814918). The document reflects only the author's view, European Commission and Japanese MIC are not responsible for any use that may be made of the information it contains.

References

[1] VirIoT web site, https://github.com/fed4iot/VirIoT

[2] NGSI v2 tutorials, https://fiware-tutorials.readthedocs.io/en/latest/

[3] NGSI-LD tutorials, https://ngsi-ld-tutorials.readthedocs.io/en/latest/

[4] ETSI oneM2M, https://www.etsi.org/committee/1419-onem2m