Federating Autonomous IoT Silos: Fed4IoT Approach

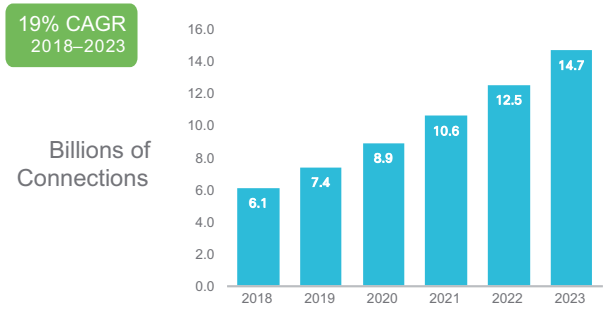
NAKAZATO Hidenori

Waseda University

25 November 2020

IoT is Growing

- The market is expected to reach $ 1,600B globally by 2024[1]
  - Growth is 21.1% annually from 2016 to 2024
- M2M connections will grow 2.4-fold from 2018 to 2023



Source: Cisco Annual Internet Report, 2018–2023

[1] Kenneth Research Report, https://www.kennethresearch.com/report-details/internet-of-things-iot-market/10325467

Smart City

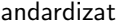
- Smart City: A city or region that utilizes ICT and other new technologies to manage issues it is faced with and to perform overall optimization in a sustainable fashion
- Expenditure for smart city is expected to top $189.5B globally[2]
- Smart city platform market in Japan exceeds JPY 1T in 2023[3]

  (Here smart city platform includes software, services, and devices used to provide smart city services)

[2] IDC, Worldwide Smart Cities Spending Guide, https://www.idc.com/getdoc.jsp?containerId=prJPJ45347819

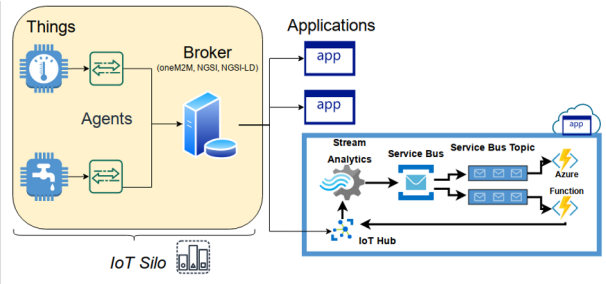
[3] NRI, https://statdb.jp/facts/41525

IoT Platforms

- IoT Platform: A set of common functions used in IoT systems
- oneM2M  : A partnership project to standardize IoT platform
  - 7 standardization bodies from Europe, Japan, US, China, and Korea formed oneM2M in 2012
  - Release-3 specification was published in 2018
- FIWARE  : An open source IoT platform
  - Initially developed in FP7 project FI-PPP
  - FIWARE Foundation promotes FIWARE

IoT Systems are Silos

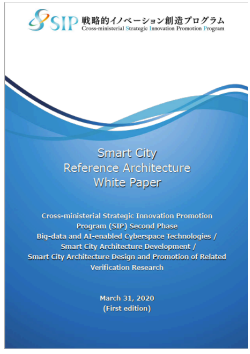
- IoT systems in individual application fields are built
- Most of the IoT systems are silos: they are build ground up



- Starting an IoT service requires large investment
- Interworking among IoT systems is difficult

Federation Required

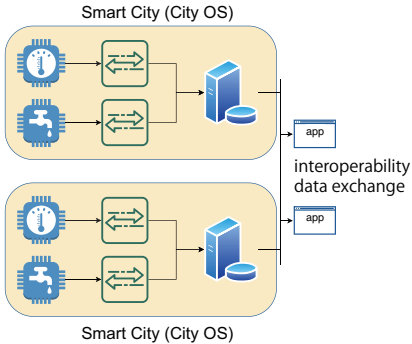
- City OS
  - Defined in Smart City Reference Architecture White Paper published by Cross-ministerial Strategic Innovation Promotion Program of Japanese Government
  - City OS: a set of system functionalities which enable access to a variety of data provided from/to Smart City Assets as well as external systems and then achieves appropriate brokering of such data with Smart City Services
  - Features in City OS are:
    - interoperability
    - data exchange
    - scalability



Last Revised at 17:39 November 25, 2020.

Federation Required

- The features in City OS are fundamental in Smart Cities for efficient delivery and exchange of services and data



Fed4IoT

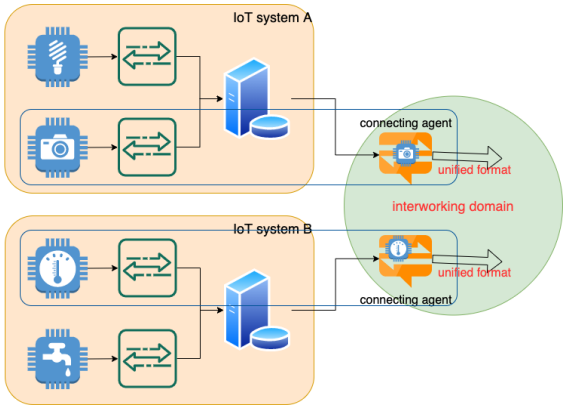
- Fed4IoT[4] is an EU-JP joint research project
- Objective:
  - Interoperability of IoT systems
  - Reduction of initial cost in developing IoT services
- Partners:




[4] The research leading to these results has been supported by the EU-JAPAN initiative by the EC Horizon 2020 Work Programme (2018-2020) Grant Agreement No.814918 and Ministry of Internal Affairs and Communications "Strategic Information and Communications R&D Promotion Programme (SCOPE)" Grant no. JPJ000595, "Federating IoT and cloud infrastructures to provide scalable and interoperable Smart Cities applications, by introducing novel IoT virtualization technologies (Fed4IoT)".

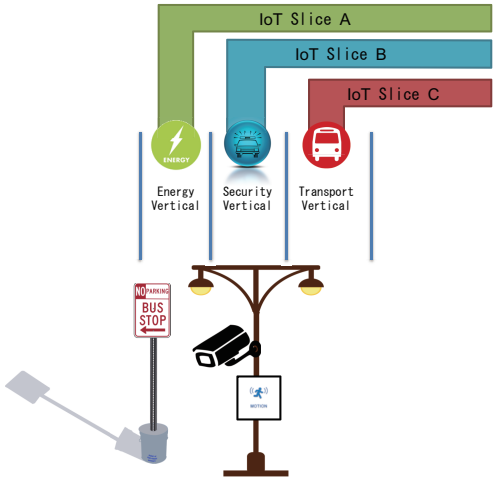
Interoperability of IoT Systems

- Connecting by agents with thing-by-thing basis
- Converting IoT data to a unified format



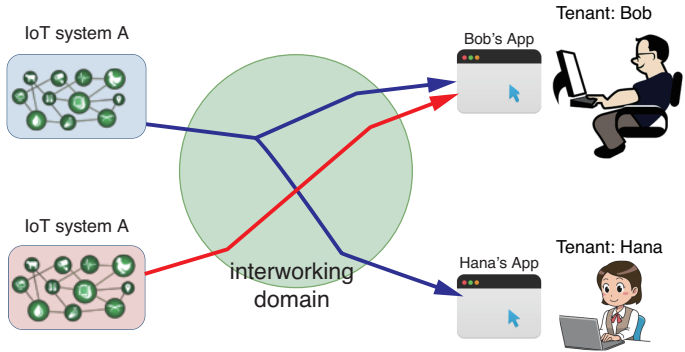
Initial Cost Reduction

- Reuse IoT devices
- One camera can be used for
  - counting # waiting passengers
  - detecting people to turn on the light, and
  - surveillance.



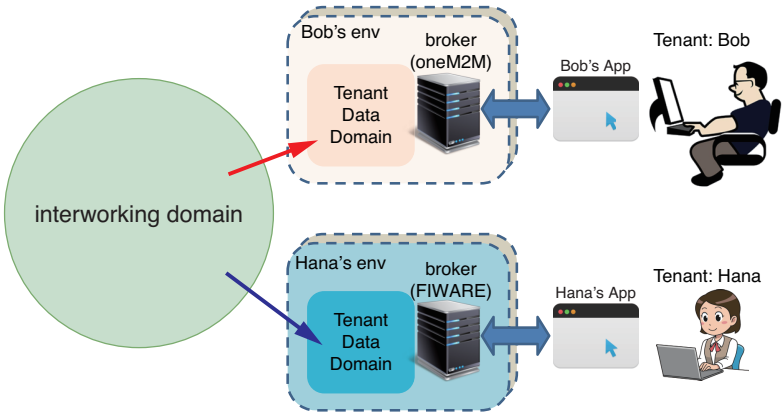
Initial Cost Reduction

- Share IoT devices and IoT infrastructure



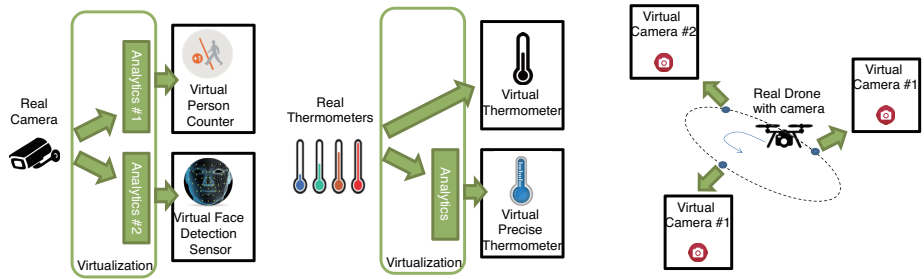
Initial Cost Reduction

- Isolated IoT service development/execution environment



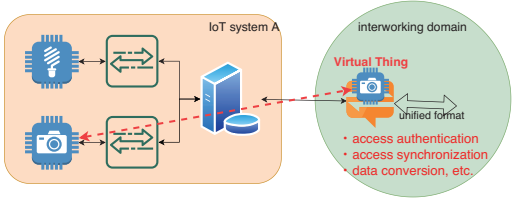
Virtual Things

- *Virtual Things* are emulation of *real things*
  - Real things, or IoT devices, are virtualized as Virtual Things



Virtual Things

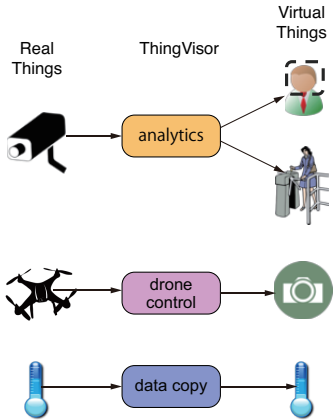
- Virtual Thing is a concept to buffer accesses to real things
- The buffer provides the opportunity to insert access authentication, access synchronization, data conversion, etc.
  - Access control and access synchronization are required for sharing
    → IoT device sharing
  - Access control regulates access to real things thing-by-thing basis
    → IoT service domain federation
  - Data conversion is necessary to connect heterogeneous IoT service domains to a unified communication platform
    → IoT service domain federation



Last Revised at 17:39 November 25, 2020.

ThingVisor

- A ThingVisor is a software component
- ThingVisors virtualize IoT devices to Virtual Things
- ThingVisors connect heterogeneous IoT service domains to a unified communication platform by inserting data format conversion
- It bridges between shared IoT device(s) and IoT service application(s) for sharing

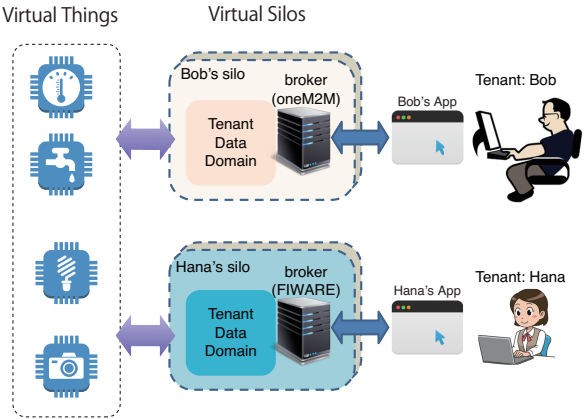


Virtual Silos

- Isolated IoT service application development/execution environment
  - Isolation is important to develop software
  - Interference from other programs cause unexpected outcome
  - Controlling actuation is a key for isolation
- Providing development environment familiar to each developer

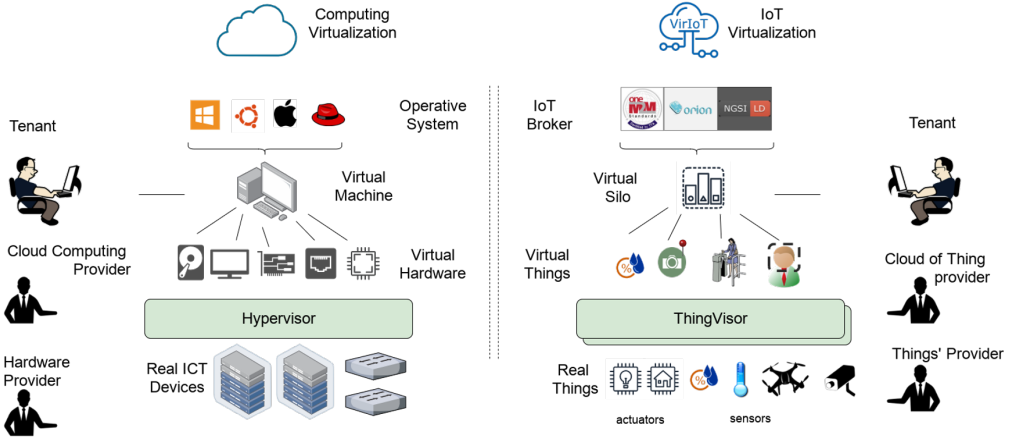
Virtual Silos

- Each Virtual Silo contains an IoT broker of tenant's choice
- Virtual Things are added to Virtual Silos to use/control the Virtual Things



Last Revised at 17:39 November 25, 2020.

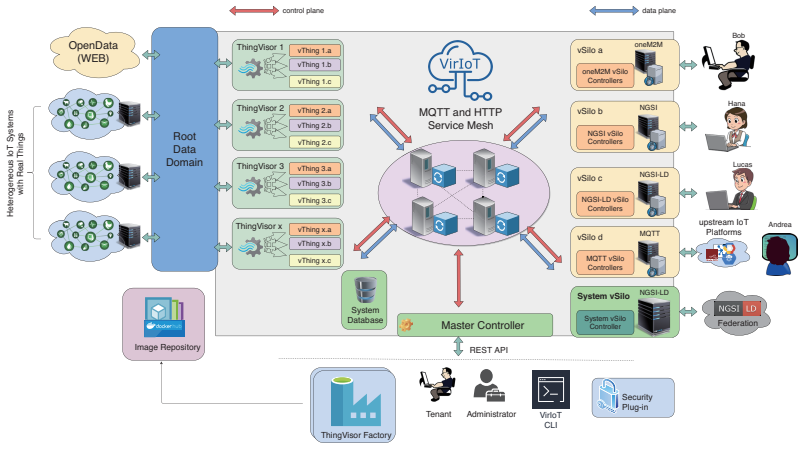
Cloud Computing vs Cloud of Things (VirIoT)



Last Revised at 17:39 November 25, 2020.

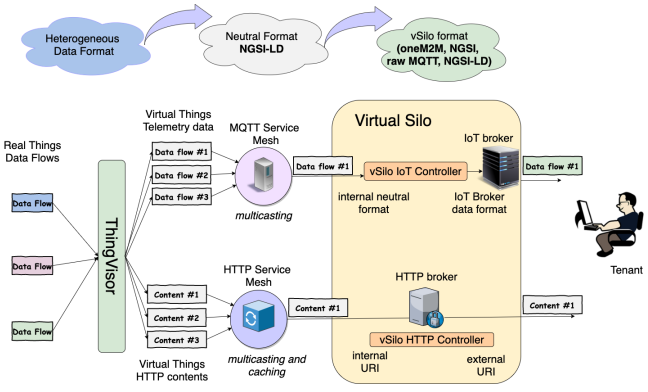
VirIoT Architecture

- Microservice
- Edge-computing
- Efficient data dissemination
- Kubernetes
- Open-source



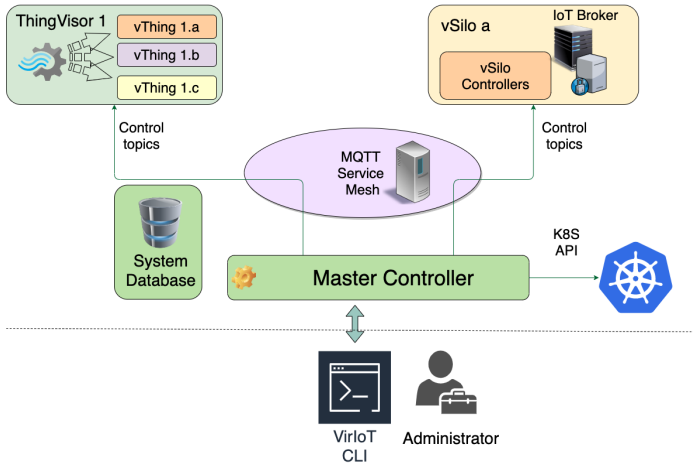
Data Plane

- Telemetry data plane
  - Data topics on a MQTT service mesh made of a cluster of MQTT servers
  - NGSI-LD as internal neutral format translated by vSilo IoT Controllers
- Content data plane (e.g. video streaming)
  - Raw contents on HTTP service mesh made of a cluster of HTTP Proxies



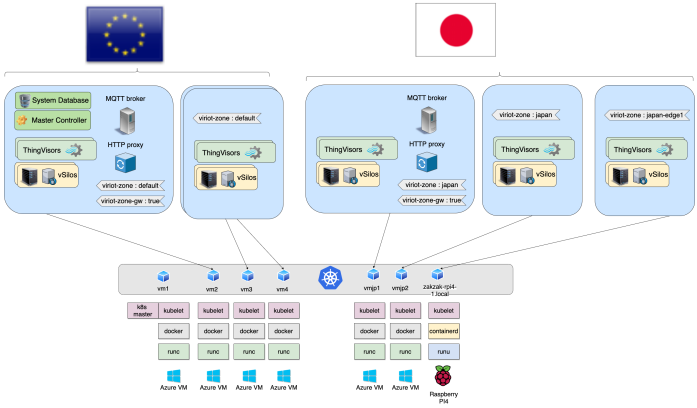
Control Plane

- Control Topics on the MQTT service mesh
- Control Commands and Procedures



VirIoT over Kubernetes: EU-JP Deployment

- Kubernetes
- VPN
- Zone labels to support edge computing
- Service topology feature to route traffic towards closest MQTT/HTTP server
  - Low latency
  - Low bandwidth consumption



Cloud of Things (VirIoT): Essential Characteristics

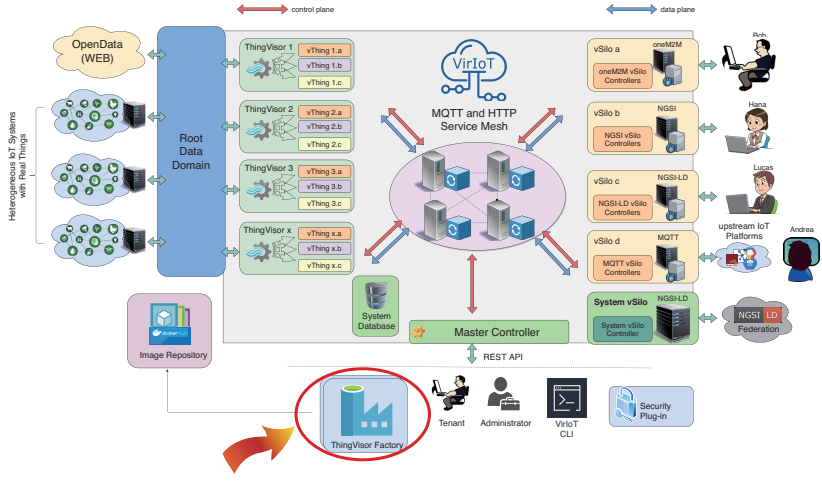
Inspired from NIST definition of cloud computing

- On-demand self-service
  - a consumer can unilaterally provision IoT resources, such as Virtual Things, IoT Broker and Virtual Silos, as needed automatically without requiring human interaction with each service provider.
- Broad IoT access
  - IoT resources are available over the network and accessed through heterogeneous IoT standard mechanisms, such as those specified by NGSI, NGSI-LD, oneM2M, etc.
- Broad support of IoT devices
  - Support of heterogeneous IoT sensors and actuators producing or consuming telemetry data and generic HTTP contents (images, streams, etc.)
- Resource pooling
  - Underlying computing and IoT resources (real things, open data, etc.) are pooled to serve multiple consumers with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

Further Development

- ThingVisor Factory
- ThingVisor Chaining
- Application of Information Centric Networking

ThingVisor Factory



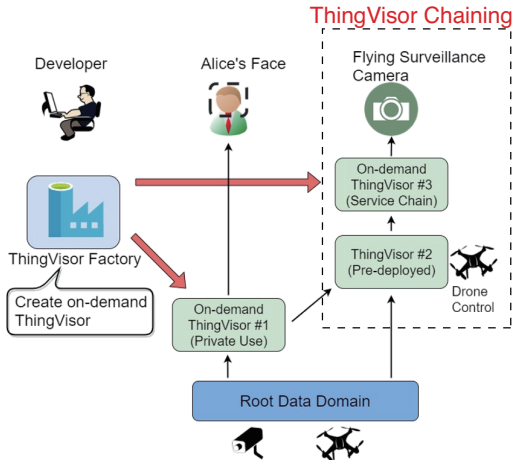
Last Revised at 17:39 November 25, 2020.

ThingVisor Factory

- Support implementation and deployment of ThingVisors on demand
- Support Docker/Kubernetes environment
- Equipped with GUI
  - GUI of ThingVisor Factory is an extension of Node-RED
- Achieve efficient resource utilization for k8s/VirIoT computing nodes
  - Virtual Things are created with “Service Function Chaining”

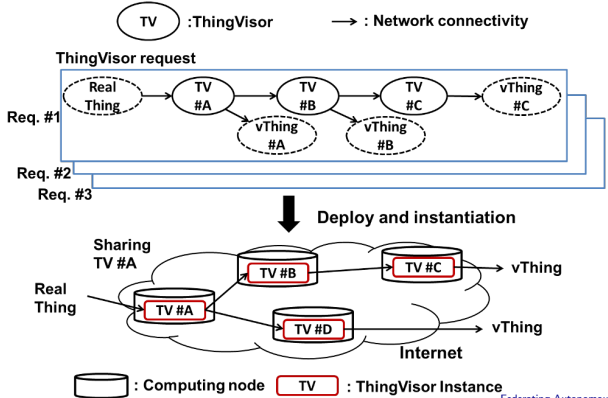
ThingVisor Chaining

- ThingVisor Factory creates Virtual Things using a chain of ThingVisors

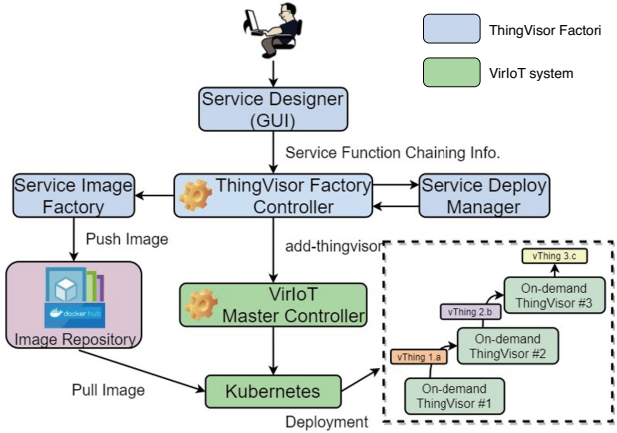


ThingVisor Chaining

- ThingVisor developers design a chain of ThingVisors using ThingVisor Factory
- The chain of ThingVisors are deployed over a network
  - Computing resources are distributed over the network



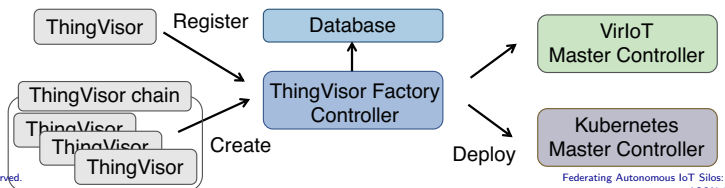
Architecture of ThingVisor Factory



ThingVisor Factory Controller

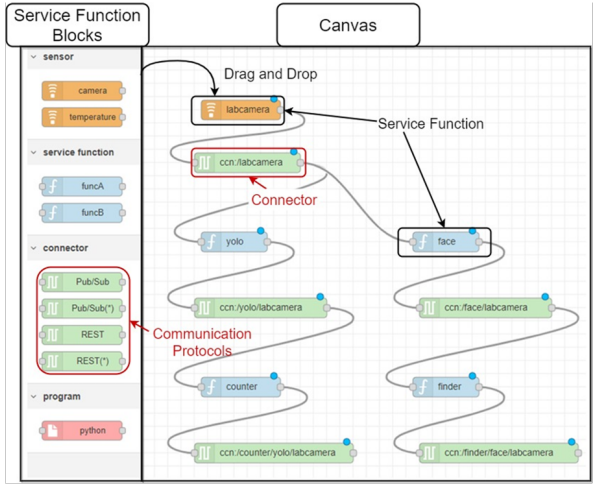
Provide APIs to call function blocks for ThingVisor Factory, VirIoT Master Controller and Kubernetes

- CRUD for ThingVisors and ThingVisor chains
  - For designing on-demand ThingVisors
- Schedule for ThingVisor deployment
  - Determine deployment nodes and allocated resources
- Add-thingvisor
  - For deployment of ThingVisors to VirIoT system
- CRUD for Kubernetes deployment
  - For deployment of ThingVisors to a native Kubernetes environment



Service Designer

- Provide high usability interface for designing ThingVisors
- Extend from Node-RED
  - Node-RED
    - Local environment
    - JavaScript
  - ThingVisor Factory
    - Specify protocols: ICN and Pub/Sub
    - Network wide
    - Python
- Output ThingVisor chain request by formatting JSON



Service Deploy Manager

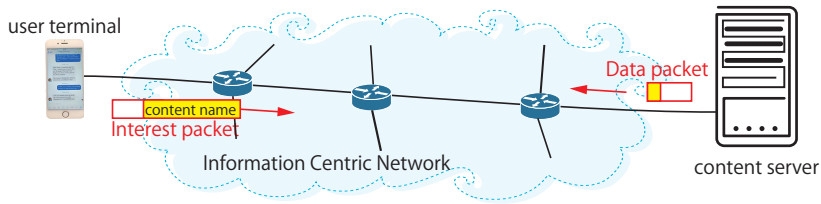
- Performance Monitoring
  - Computing resources: Kubernetes APIs (e.g., kubectl top)
    - CPU and Memory usage
  - Network and other performances: ThingVisor logger
    - Throughput, input and output data size, and processing time
- Optimize ThingVisor deployment
  - Determine optimal deployment nodes and resource allocations
- Solve routing for ThingVisor chaining

Application of Information Centric Networking

- Information Centric Networking (ICN) forwards packets with content names instead of IP address
- Name-based forwarding is handy for executing a chain of ThingVisors
  - It is not the case in Pub/Sub, i.e., MQTT
- In-network caching capability can also be exploited in executing chains of functions

Communication Model in ICN

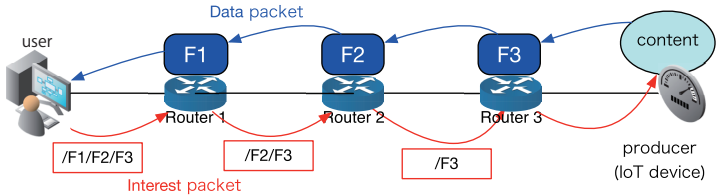
Communication is performed using two types of packets: Interest packet and Data packet



- A user requests a content using an Interest packet
- Content names in Interest packets specifiy the packet destinations
- The content server with the requested content returns a Data packet containing the requested content

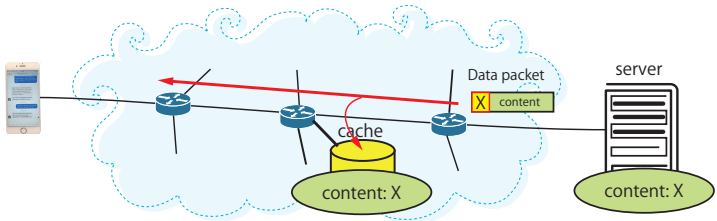
ThingVisor Chaining with ICN

- Extension of ICN for service function chaining, or ThingVisor Chaining
  - Interest packet can specify a sequence of functions and a content
  - After deriving the content, Data packet goes through functions to process the content



In-Network Caching

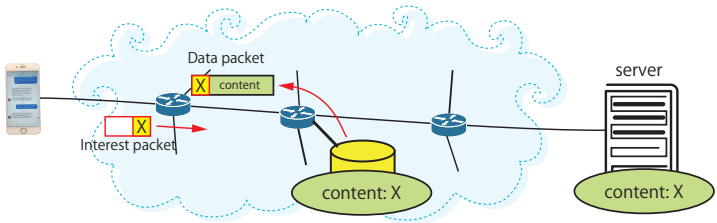
Routers can cache contents



- Routers temporarily hold the content included in Data packets passing by

In-Network Caching

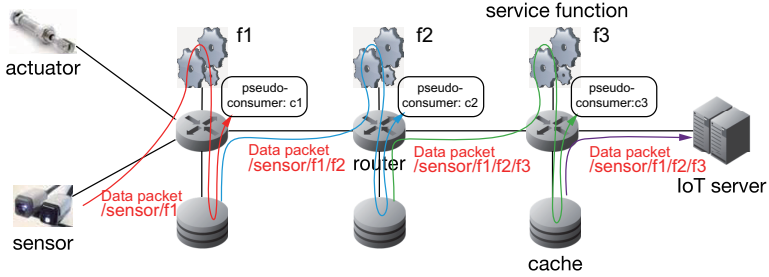
- A requested content can be derived without going all the way to its server using this mechanism



- It can avoid congestion around servers and their surrounding network
- Content caching became available due to name-based packet delivery and packet signature
  - Without content name, cached content cannot be matched with requested content
  - Without par-packet signature, suspicious contents may be cached and delivered

Caching in ThingVisor Chaining

- ThingVisors are typically executed periodically due to the nature of IoT systems
- Each ThingVisor in a chain may be periodically executed autonomously and cache its execution result
- Without explicit sequential execution, an execution chain can be formed
- Latency may be reduced



Conclusion

- IoT systems are ballooning and Smart Cities are being developed
- Federating those systems is the next step development
- Fed4IoT is an EU-JP joint research project to federate IoT systems and introduces the concepts:
  - *Virtual Things*,
  - *ThingVisors*, and
  - *Virtual Silos*

  to share IoT devices and infrastructures, and to promote IoT data exchange
- Fed4IoT project is hoping to encourage development of IoT services by lower CAPEX
- Software is avaiable at GitHub

Reference

- Project web site : `https://fed4iot.org`
- GitHub of software : `https://github.com/fed4iot/VirIoT`
- Video of VirIoT software in Action : `https://fed4iot.org/wp-content/uploads/ftp/VirIoT-in-Action.mp4`

Thank you
Questions ?