엣지 컴퓨팅 표준화 동향 및 오픈소스 기술 동향

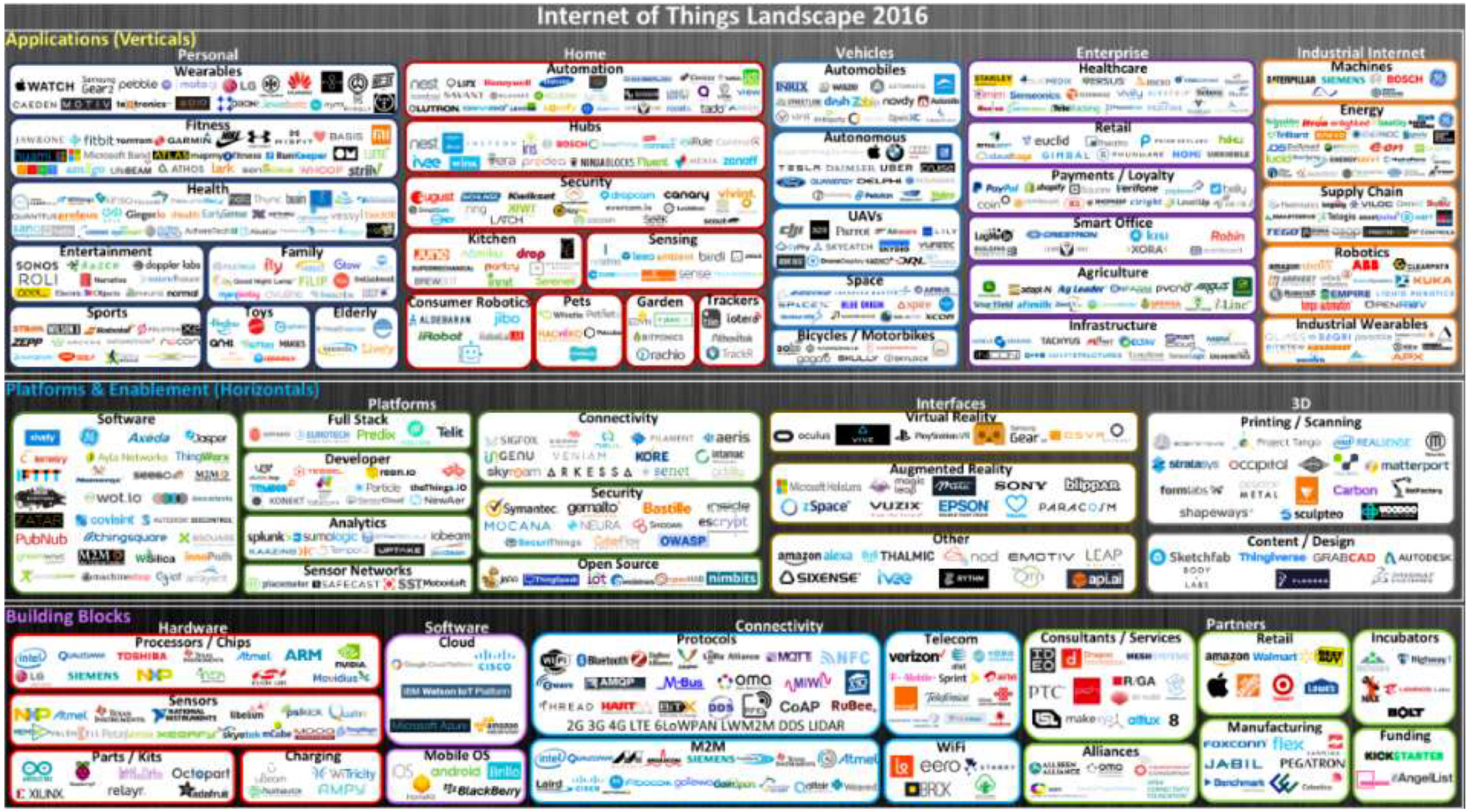
동의대학교
윤주상
jsyoun@deu.ac.kr

Computing 기술의 변화

Cloud
(2000 – 2015)

Edge / Fog
(2016 – ?)

엣지 컴퓨팅의 필요성



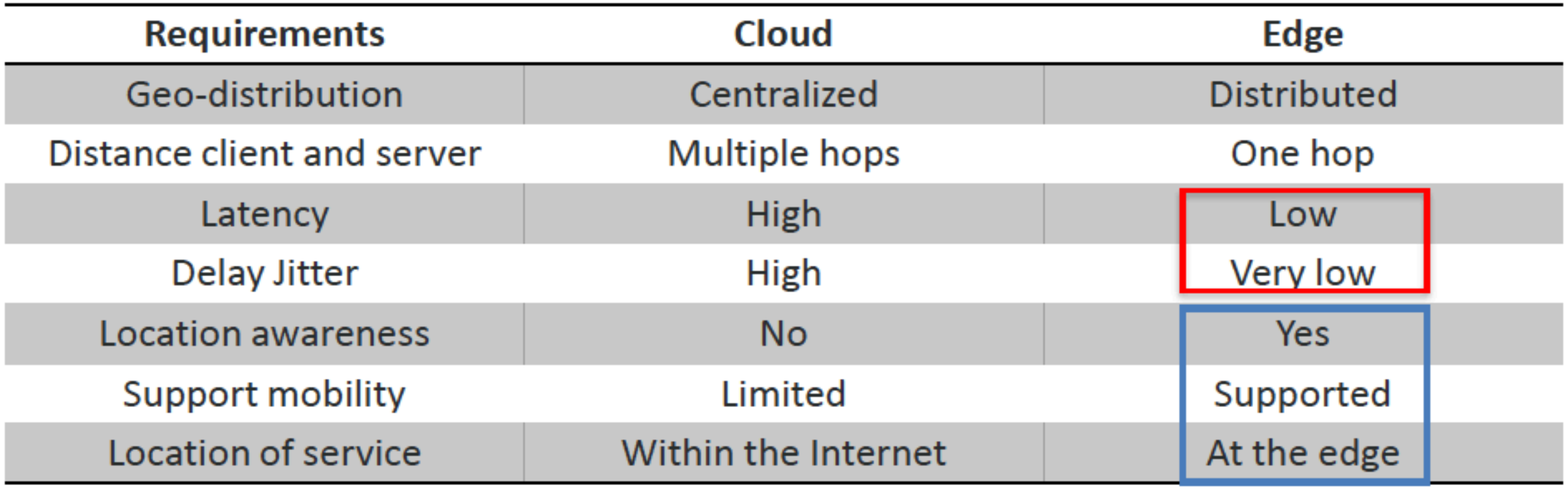
출처: edgexfoundry.org

Edge Computing Definitions

- Edge computing
  - To be a **form of distributed computing** at the **edge** of the network
  - Edge nodes implementing a reduced form of cloud computing in locations near the edge of the network
- Fog computing
  - To be a **system-level horizontal architecture** that distributes resources and services of computing, storage, control and networking anywhere along the continuum from Cloud to Things
    - (http://www.openfogconsortium.org/resources/#definition-of-fog-computing)

엣지 컴퓨팅의 장점

- Low Latency : satisfy requirements from mission-critical apps
- Data Locality : prevent unnecessary exposure of privacy data
- Save Bandwidth : save required BW at Edge instead of Cloud



Requirements	Cloud	Edge
Geo-distribution	Centralized	Distributed
Distance client and server	Multiple hops	One hop
Latency	High	Low
Delay Jitter	High	Very low
Location awareness	No	Yes
Support mobility	Limited	Supported
Location of service	Within the Internet	At the edge

Ref : http://blogs.cisco.com/perspectives/iot-from-cloud-to-fog-computing

표준화 기구 별 엣지 컴퓨팅 정의

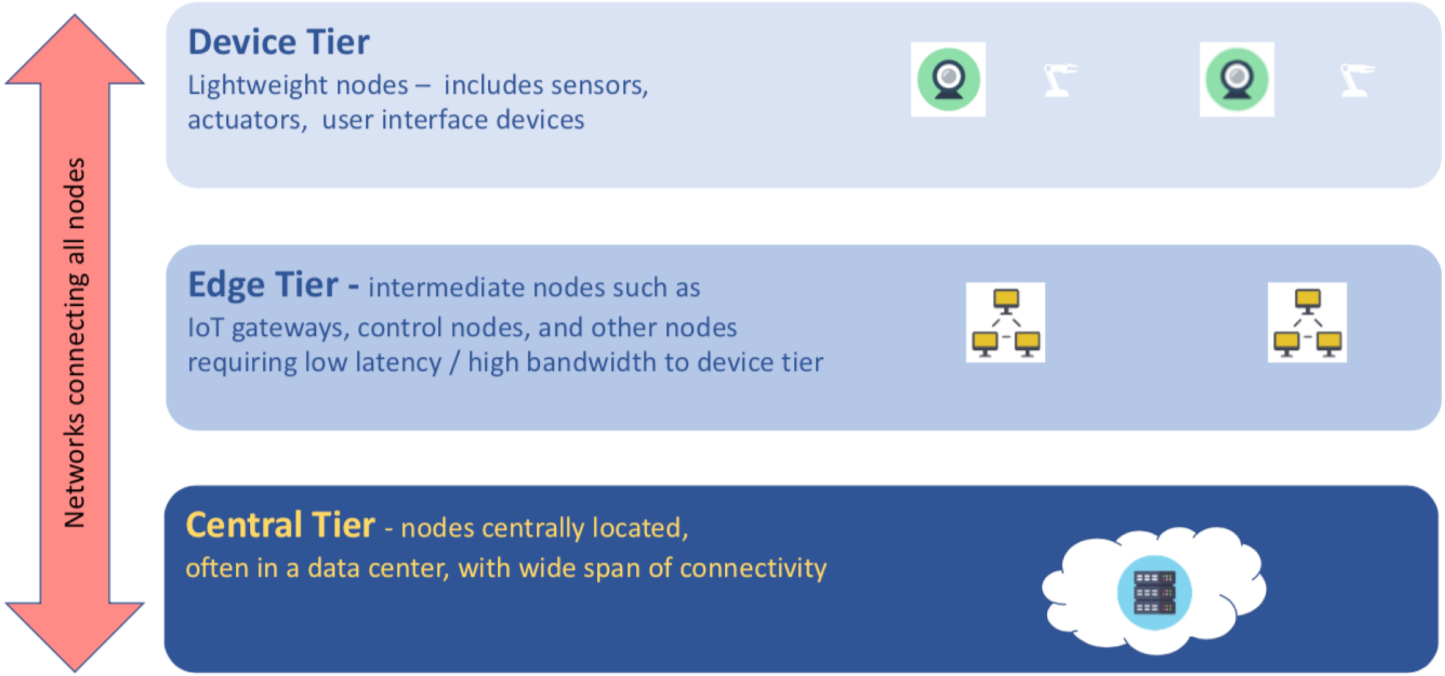
- 표준화 기구(ITU-T/ISO-TR/OpenFog) 별 엣지 컴퓨팅 정의

  - (ISO/IEC JTC 1 SC38) Form of distributed computing
    - in which data processing and data storage takes place on nodes which are near to the edge. The "edge" is marked by the boundary between pertinent digital and physical entities, i.e. between the digital system and the real world, delineated by networked sensors and actuators.

  - (ISO/IEC JTC 1 SC41) IoT Edge Computing
    - IoT systems generally use distributed computing resources
    - To be characterized **by networked systems** in which significant data processing ("**compute**") and information storage ("**storage**") takes place **on devices and nodes near the edge of the network**, rather than in some centralized location

Architectural Foundations of Edge Computing

- Organization of nodes in edge computing



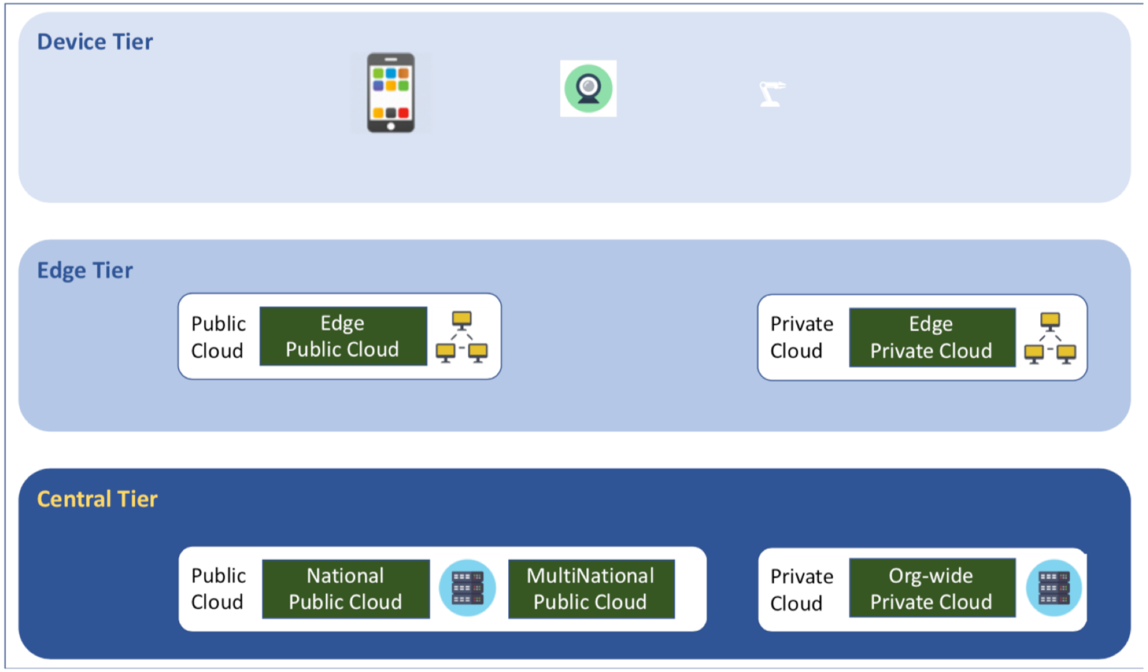
Device Tier
Lightweight nodes – includes sensors, actuators, user interface devices

Edge Tier - intermediate nodes such as IoT gateways, control nodes, and other nodes requiring low latency / high bandwidth to device tier

Central Tier - nodes centrally located, often in a data center, with wide span of connectivity

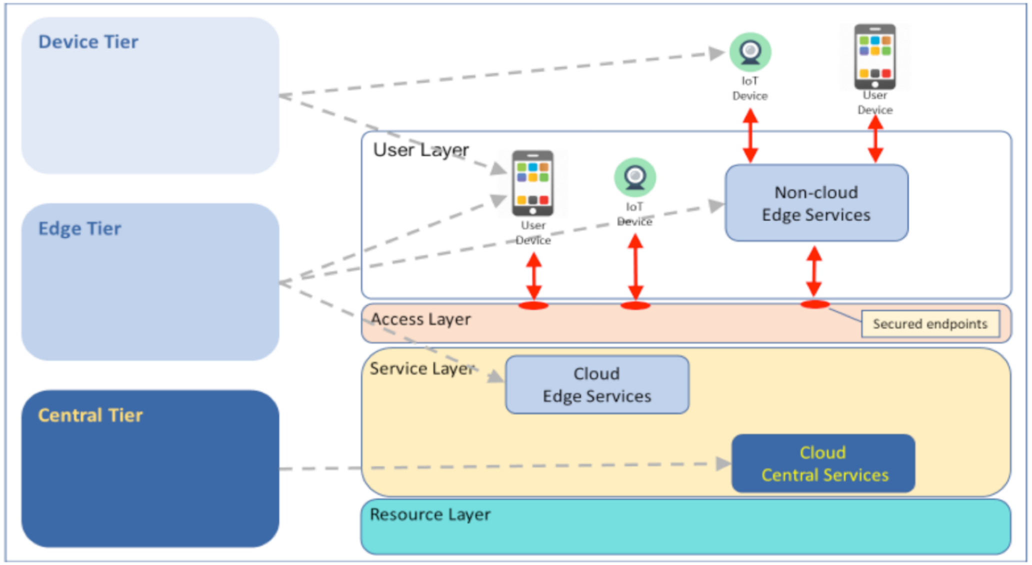
The Relationship of Edge Computing to Cloud Computing

- Edge computing can exist on its own, without any relationship to cloud computing.
- However, for many systems cloud computing is used in one or more of the tiers



Relationship of Edge Computing Tiers to Cloud Computing

- To show the relationship of edge computing to cloud computing as described in the ISO/IEC 17789 Cloud Computing Reference Architecture
  - Many edge computing systems are effectively implemented by CSCs rather than CSPs
  - The elements in the user layer are implemented and under the control of the CSC - not the CSP

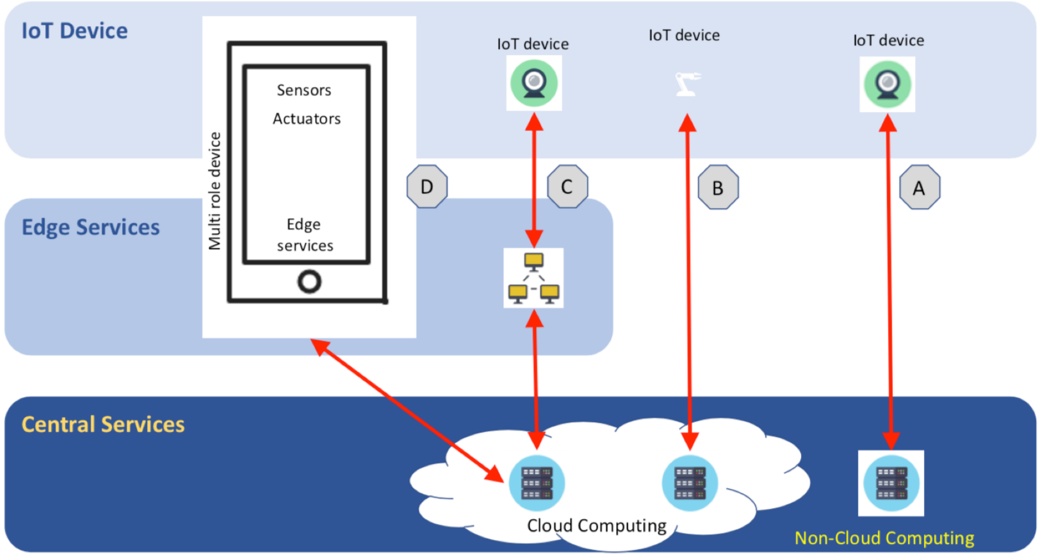


CSP Function to Support Edge Computing

- Some CSPs offer cloud services that have capabilities that are designed to support edge computing
- Many of these cloud services are particularly designed to support IoT systems and include capabilities such as:
  - Device registration
  - Device management and operation
  - Device communication (e.g. support of lightweight protocols)
  - Device data storage

The Relationship of Edge Computing to IoT

- Edge computing is central to IoT systems
  - Many IoT systems do include an edge tier, containing IoT gateways and control nodes
- The IoT, edge computing, and cloud computing concepts are independent. While often deployed together



Networking and Edge Computing

- Types of communications network, which are broadly applicable to edge computing
  - Proximity network
    - Proximity networks are local and limited in range
    - Proximity networks often use specialized protocols
  - Access network
  - Services network
    - Services networks typically connect applications and services running in the central tier
    - Services networks typically exist within a cloud data centres
  - User network

Deployment Models

- Private deployment
  - This is either owned and managed by a single organization, or where a provider makes the resources available exclusively to a single organization (i.e. no sharing of resources)
- Public deployment
  - This is owned and managed by a provider who makes the resources available to a wide range of customers (i.e. resources are shared)
- Community deployment
  - This is owned and managed on behalf of a group of organizations who have a relationship with one another and a shared set of requirements (i.e. the resources are shared amongst the community
- Hybrid deployment
  - where a given tier has a combination of the other deployment models

Capabilities Types

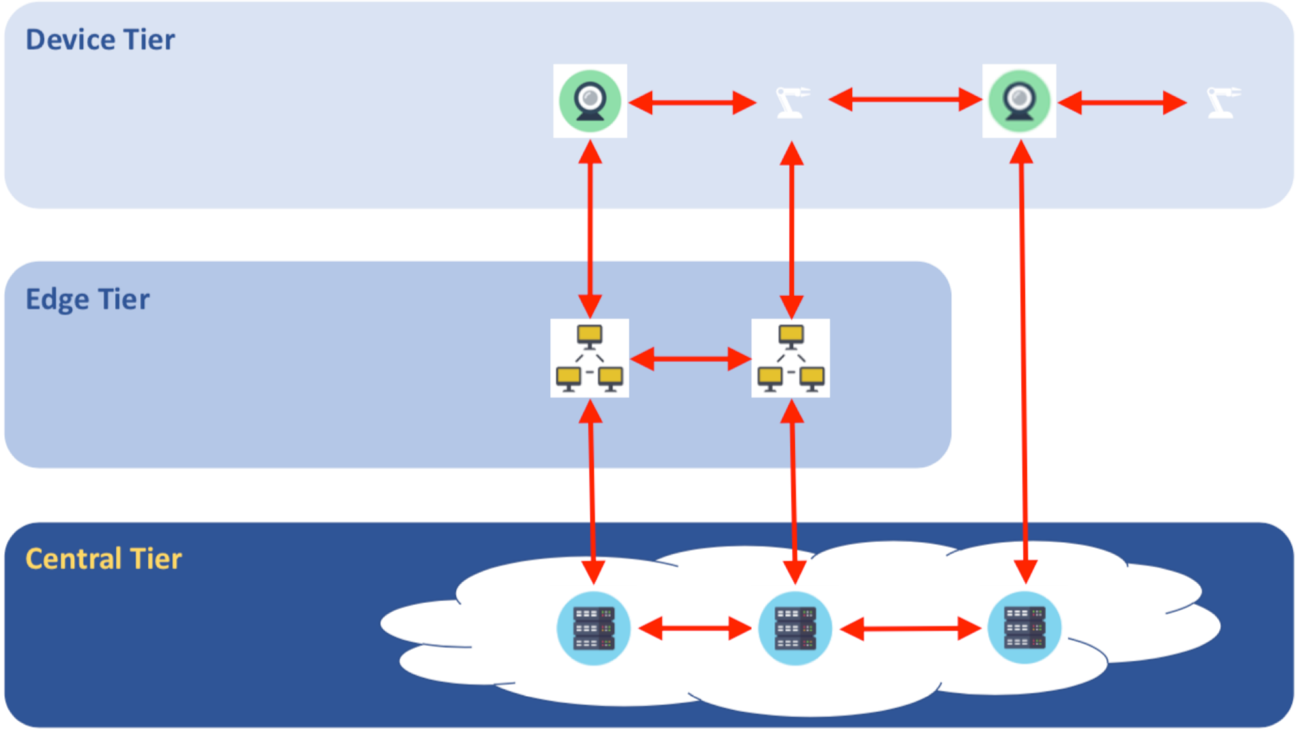
- As for cloud computing, the capabilities provided by any node in any of the tiers can be one of the three:
  - Infrastructure
    - where the user of the node can provision and use processing, storage or networking resources
  - Platform
    - where the user can deploy, manage and run user-created or user-acquired applications using one or more execution environments supported by the node
  - Application
    - where the user can make use of one or more complete applications provided by the node

Service Categories

- Analytics
- Image processing
- Entity tracking
- Control processing
- Content distribution
- IPTV
- Platform services such as running containers, object stores, databases

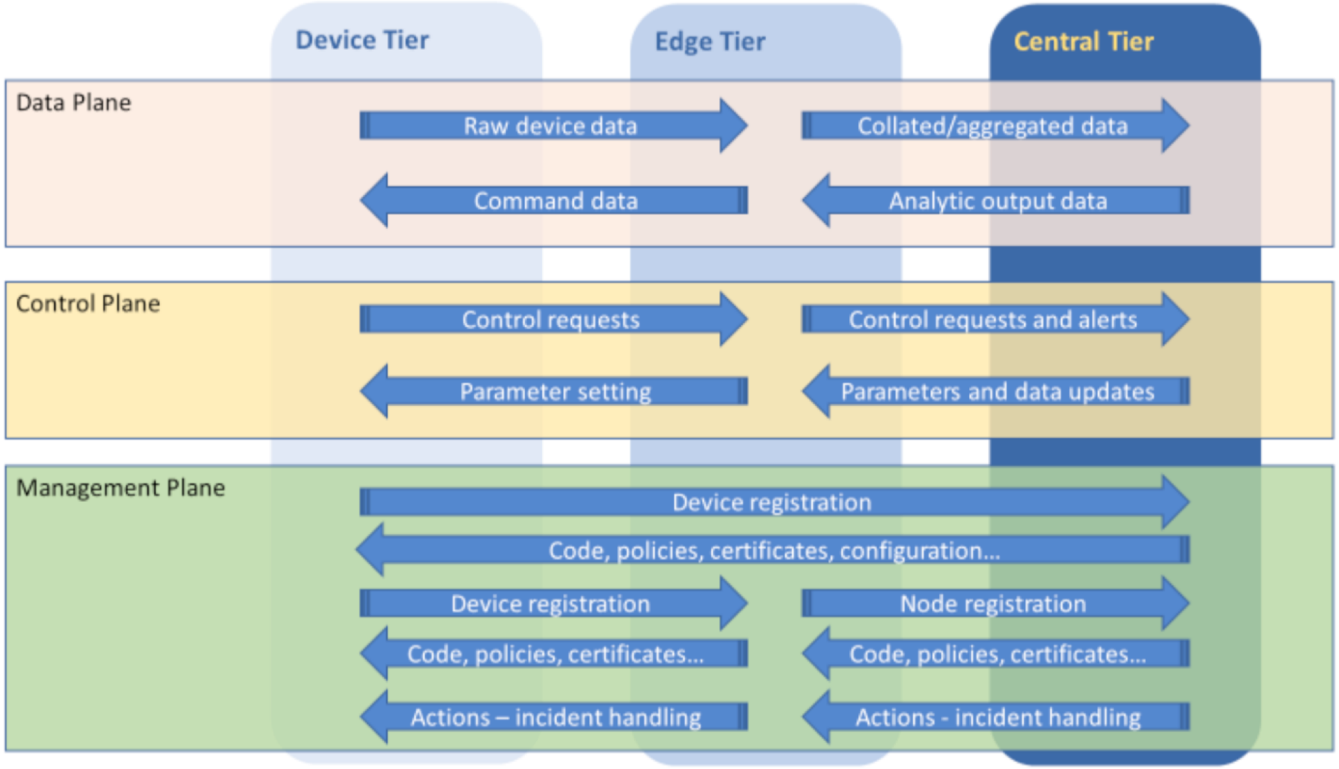
Data Flow

- Data flow can take place between two or more nodes within one edge computing tier



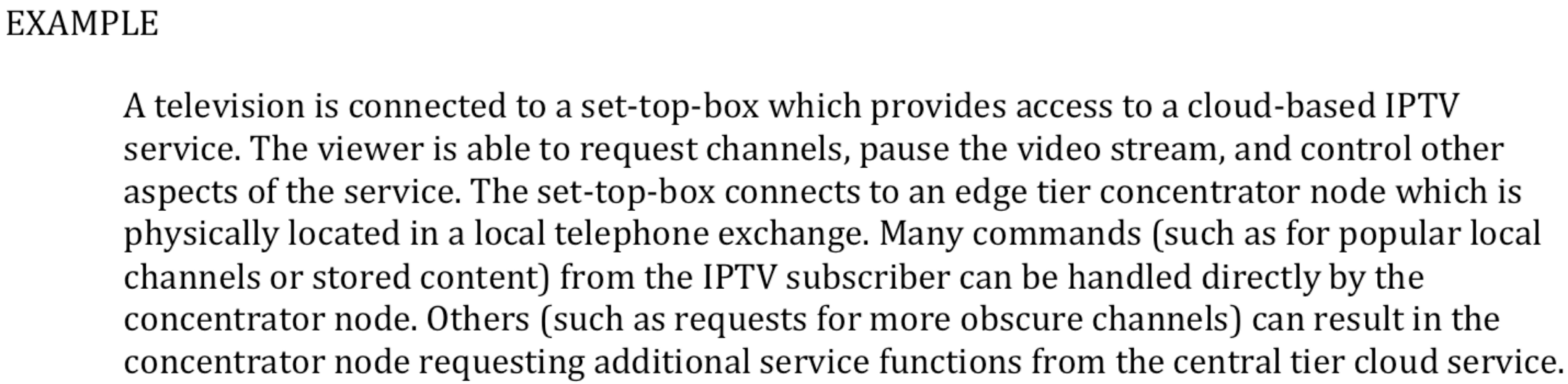
Capability of Edge Computing

- Relationship of capability planes to edge computing tiers



Cloud-based Management and Control

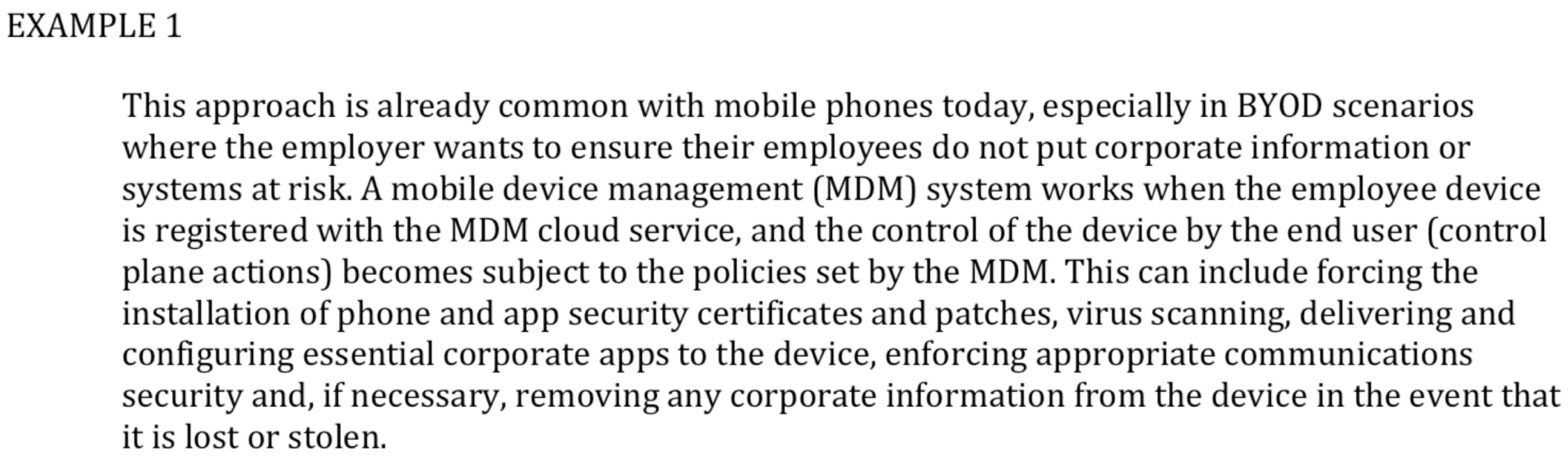
- Cloud-based **management** and **control** of **edge tier nodes** and **device tier devices**
- Control of services from a device



EXAMPLE

A television is connected to a set-top-box which provides access to a cloud-based IPTV service. The viewer is able to request channels, pause the video stream, and control other aspects of the service. The set-top-box connects to an edge tier concentrator node which is physically located in a local telephone exchange. Many commands (such as for popular local channels or stored content) from the IPTV subscriber can be handled directly by the concentrator node. Others (such as requests for more obscure channels) can result in the concentrator node requesting additional service functions from the central tier cloud service.

- Management of devices and edge nodes from a cloud service



EXAMPLE 1

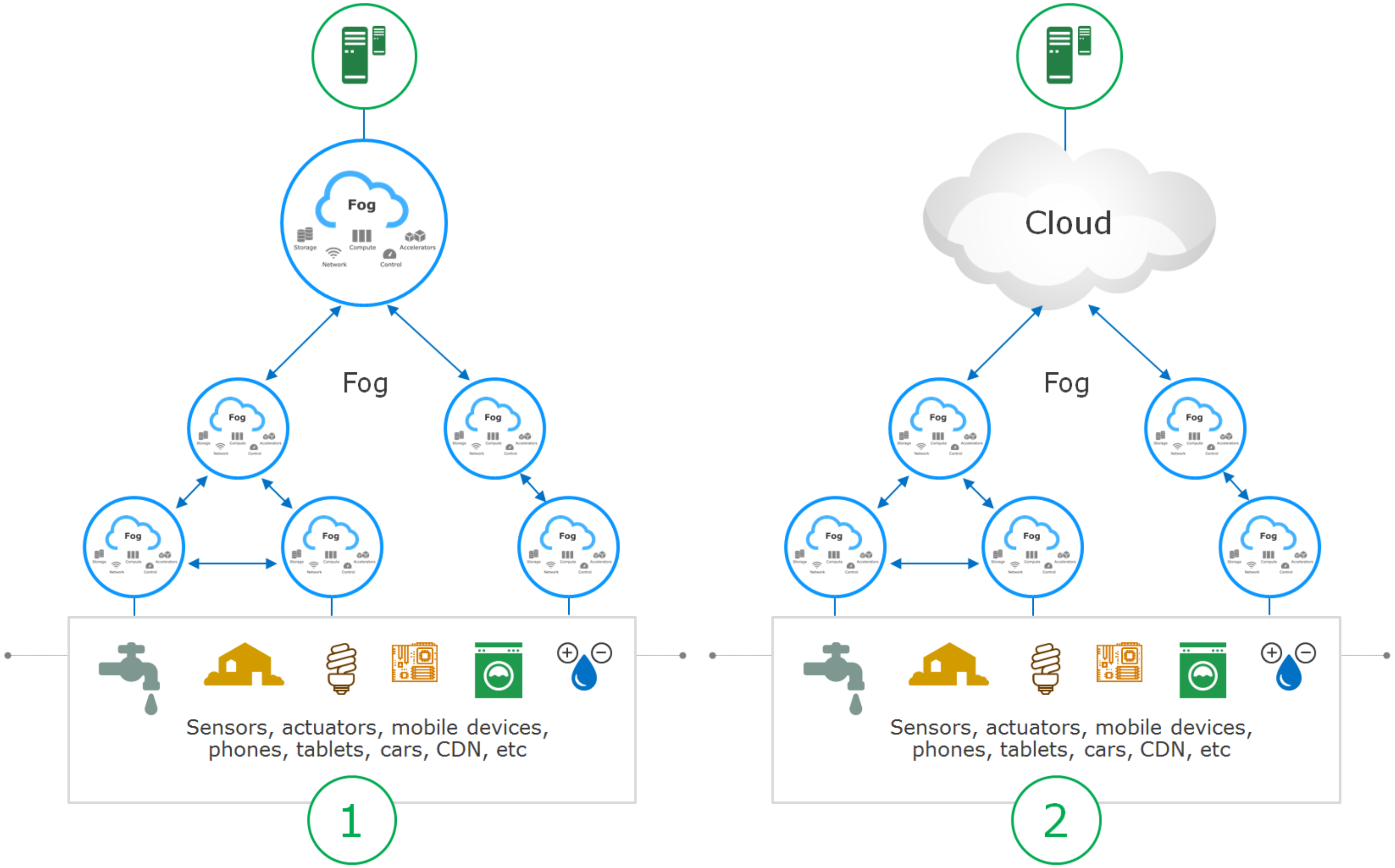
This approach is already common with mobile phones today, especially in BYOD scenarios where the employer wants to ensure their employees do not put corporate information or systems at risk. A mobile device management (MDM) system works when the employee device is registered with the MDM cloud service, and the control of the device by the end user (control plane actions) becomes subject to the policies set by the MDM. This can include forcing the installation of phone and app security certificates and patches, virus scanning, delivering and configuring essential corporate apps to the device, enforcing appropriate communications security and, if necessary, removing any corporate information from the device in the event that it is lost or stolen.

ISO/IEC JTC 1 SC38 엣지 기능

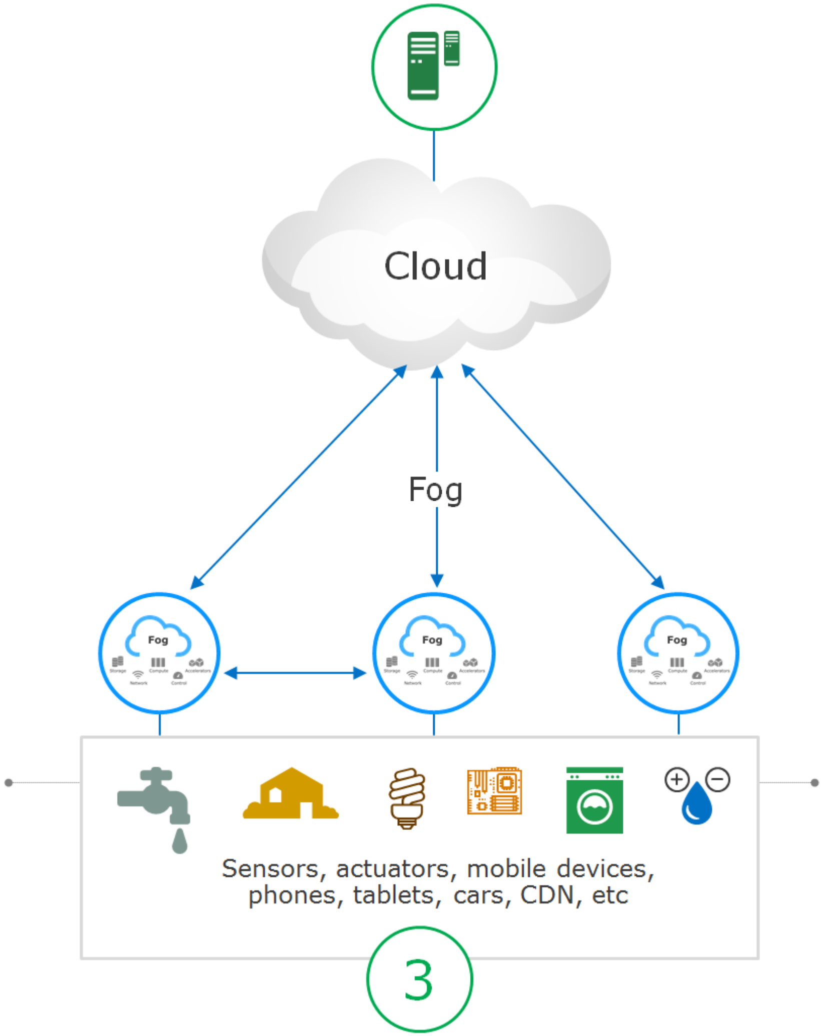
- The **edge tier** typically sits close to the device tier
  - It role is to provide direct support to the nodes in the device tier
  - One type of node in the edge tier is the **gateway**
  - The role of the gateway is to connect nodes in the device tier to the wider network
  - The gateway also typically provides a means for **managing** the nodes in the device tier.
  - Another type of node in the edge tier is the **control node**.
    - The control node receives data from nodes in the device tier and responds by issuing instructions to other nodes in the device tier.

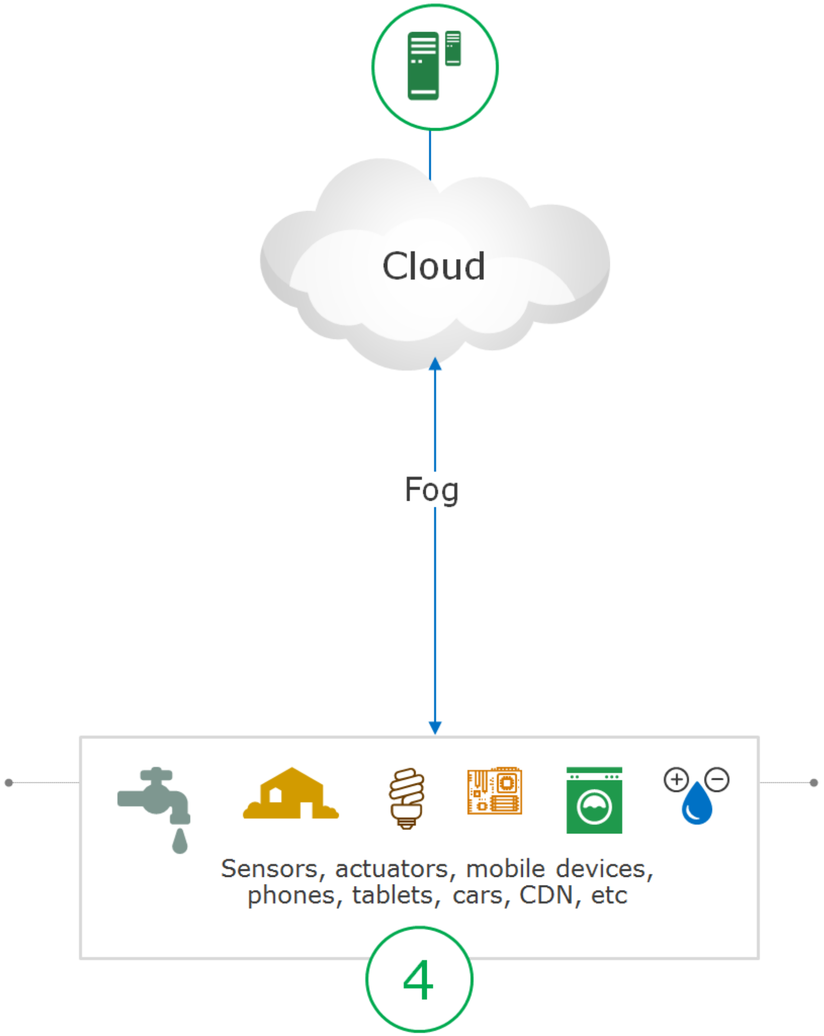
OpenFog Consortium

- Edge-Fog-Cloud Network Model

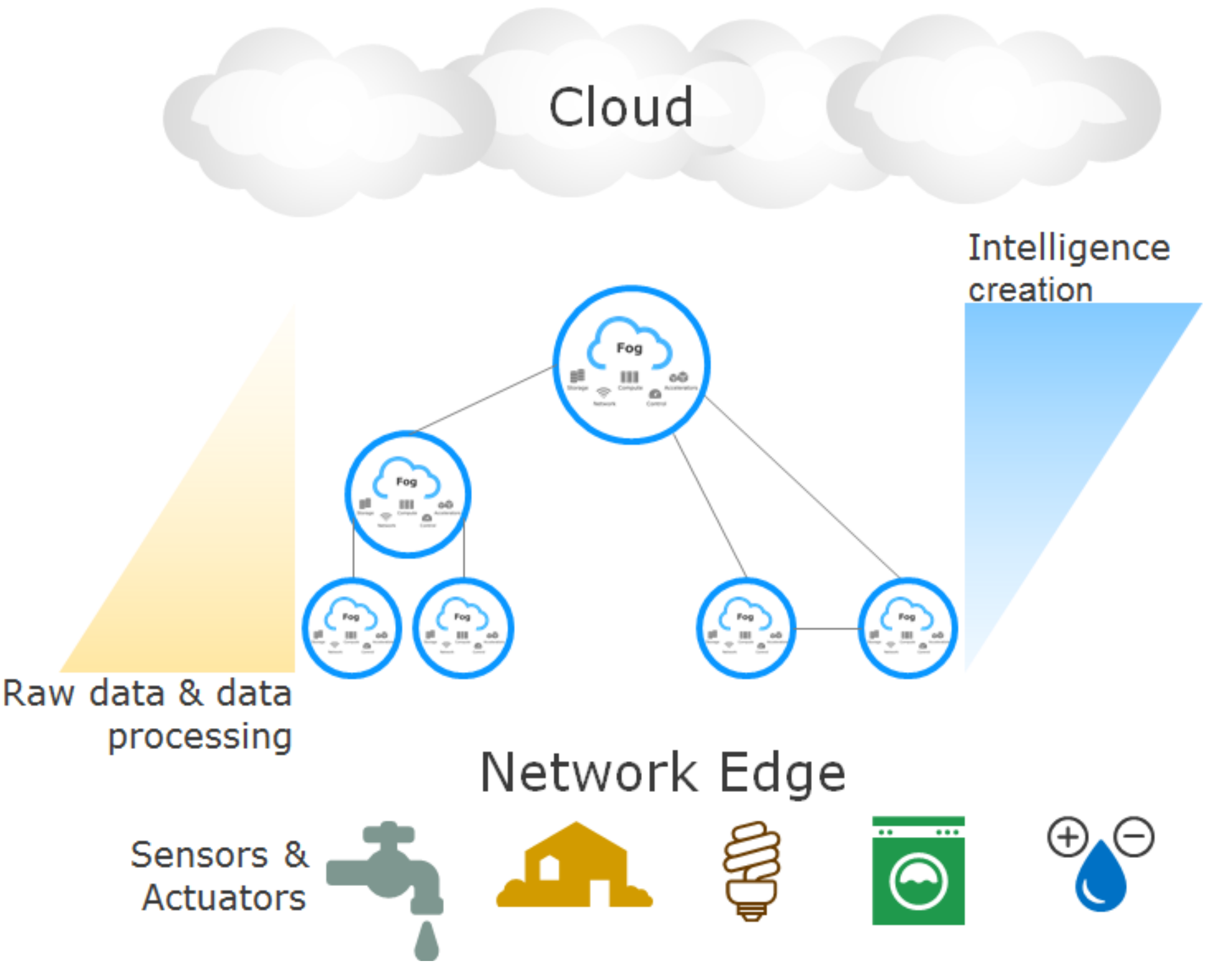

Fog
Storage
Compute
Accelerators
Network
Control
Fog
Sensors, actuators, mobile devices, phones, tablets, cars, CDN, etc
1
Cloud
Fog
Sensors, actuators, mobile devices, phones, tablets, cars, CDN, etc
2

Edge-Fog-Cloud Network Model (2/2)

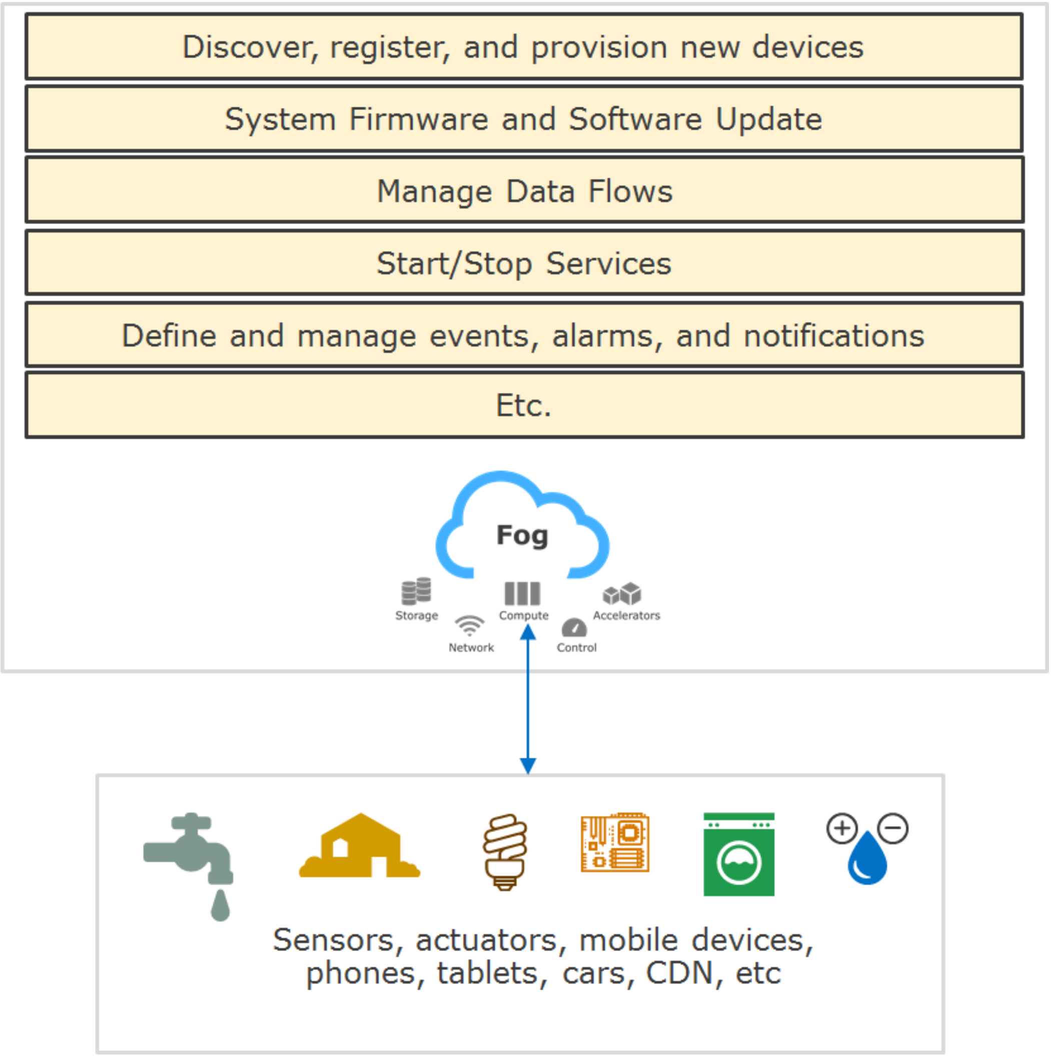

Cloud
Fog
Fog
Fog
Fog
Sensors, actuators, mobile devices, phones, tablets, cars, CDN, etc
3


Cloud
Fog
Sensors, actuators, mobile devices, phones, tablets, cars, CDN, etc
4

Intelligence from Data in Edge-Fog-Cloud Network



Management Layer in Edge-Fog Node



Discover, register, and provision new devices

System Firmware and Software Update

Manage Data Flows

Start/Stop Services

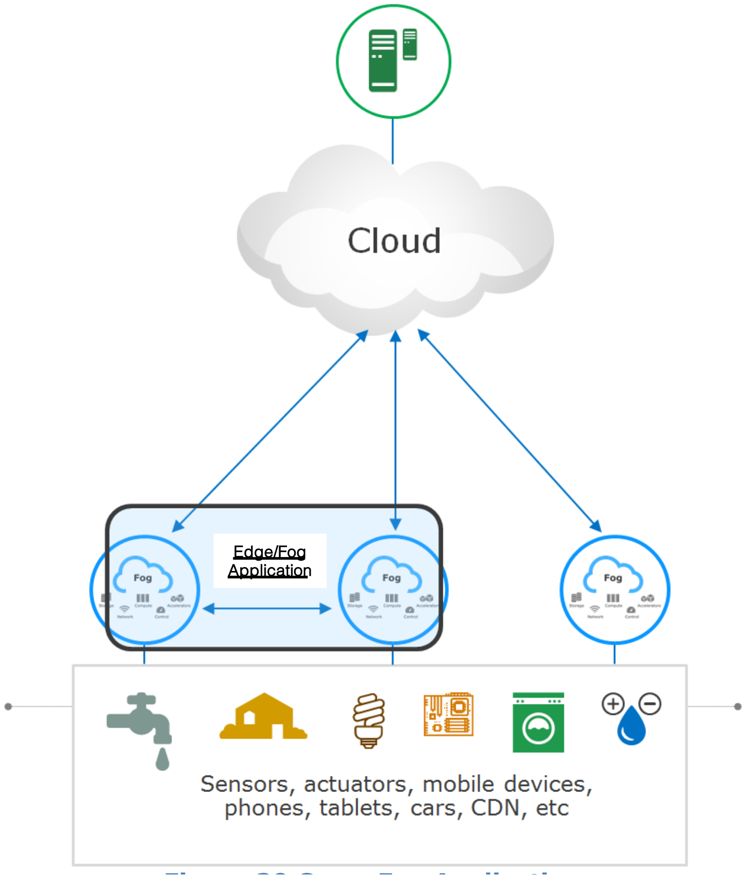
Define and manage events, alarms, and notifications

Etc.

Sensors, actuators, mobile devices, phones, tablets, cars, CDN, etc

Open Topics (1/2)

- Convergence across multiple use cases and verticals
- **Interworking between edge-fog and cloud**
- Design techniques for cloud-native applications (e.g. , for IoT) to be deployable in a distributed cloud environment (e.g., IoT device)
- **Extension of IP routers to become edge/fog nodes** (e.g. Cisco IOX)


Cloud
Edge/Fog Application
Fog
Fog
Fog
Sensors, actuators, mobile devices, phones, tablets, cars, CDN, etc

Cross-Edge

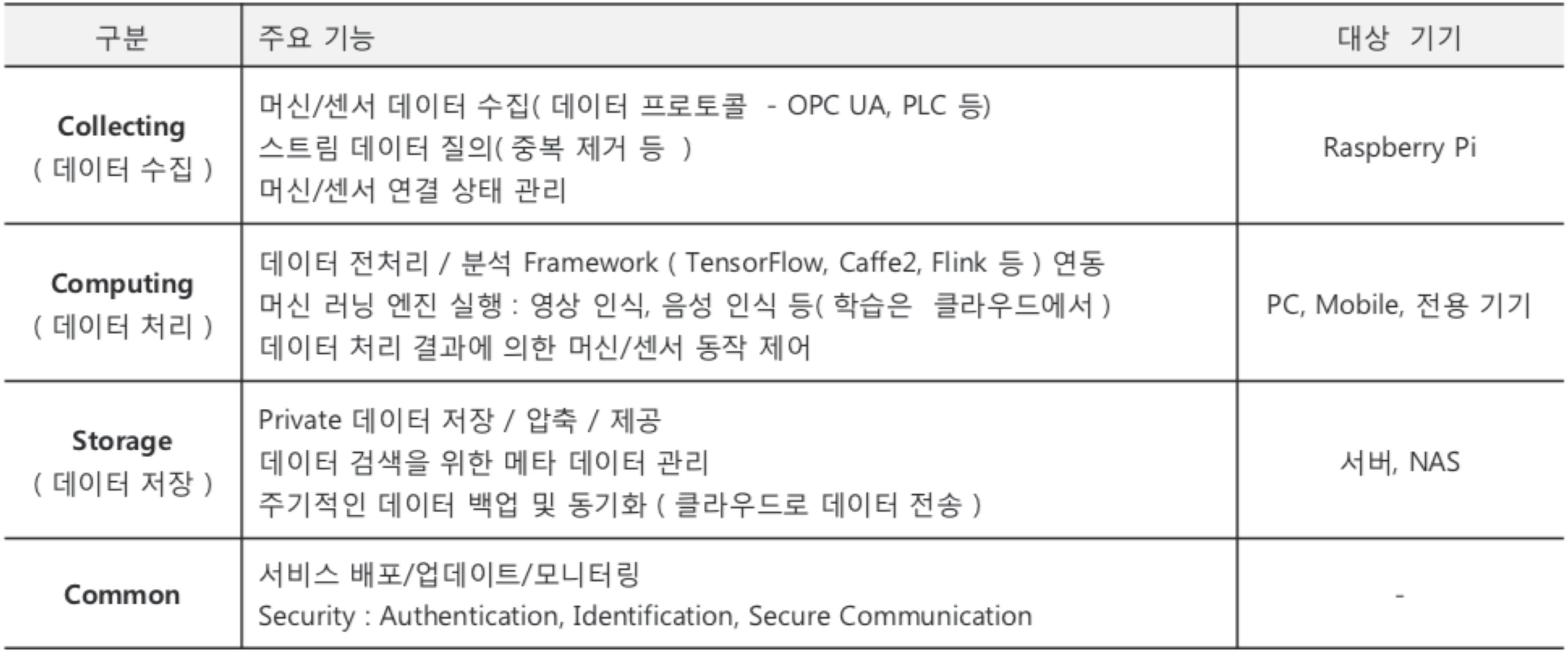
Open Topics (2/2)

- How to realize a scalable and resilient EC network?
- How to achieve connections with requirement of bounded latency **through EC**?
- How to **orchestrate** the resources at the edge, e.g. computing, storage, communication?
- How to **achieve the interoperability** between edge computing nodes of different vertical industries?
- How to **build** edge computing platforms?
- How to **guarantee** the data privacy and security of the vertical operation?
- How to **define the frontier between the edge and the cloud** ?

오픈소스 기술 동향

Edge Platform 기능

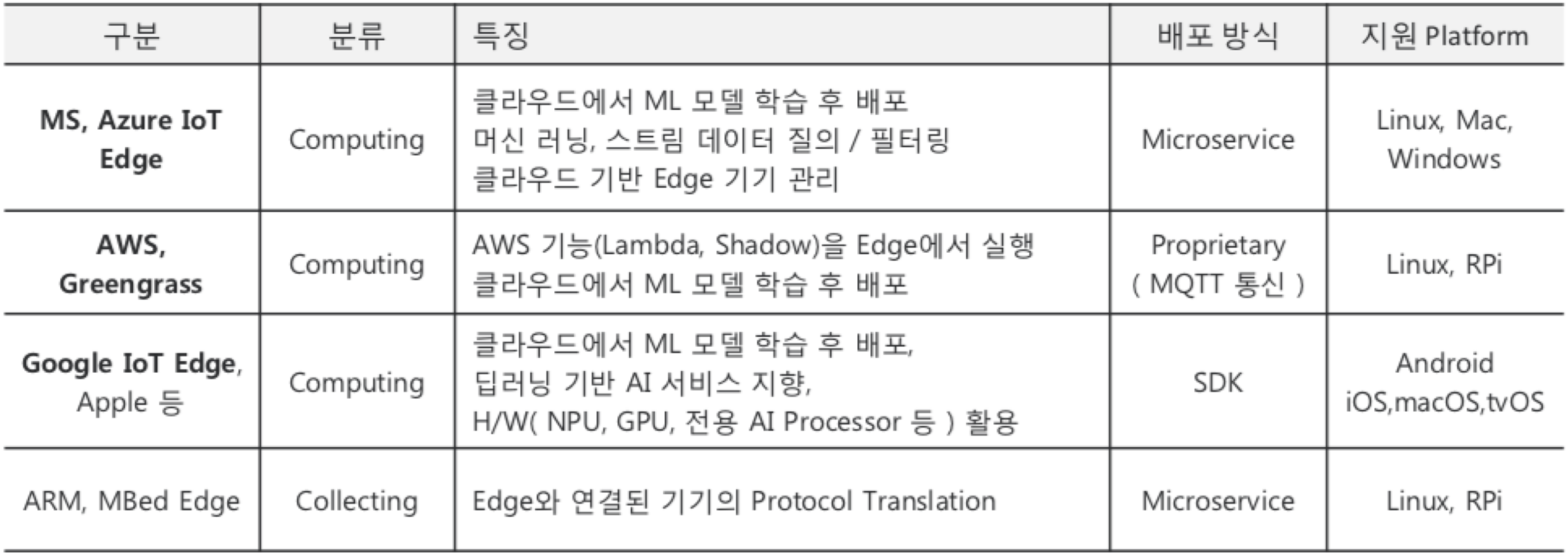
- 기능 분류



구분	주요 기능	대상 기기
Collecting (데이터 수집)	머신/센서 데이터 수집(데이터 프로토콜 - OPC UA, PLC 등) 스트림 데이터 질의(중복 제거 등) 머신/센서 연결 상태 관리	Raspberry Pi
Computing (데이터 처리)	데이터 전처리 / 분석 Framework (TensorFlow, Caffe2, Flink 등) 연동 머신 러닝 엔진 실행 : 영상 인식, 음성 인식 등(학습은 클라우드에서) 데이터 처리 결과에 의한 머신/센서 동작 제어	PC, Mobile, 전용 기기
Storage (데이터 저장)	Private 데이터 저장 / 압축 / 제공 데이터 검색을 위한 메타 데이터 관리 주기적인 데이터 백업 및 동기화 (클라우드로 데이터 전송)	서버, NAS
Common	서비스 배포/업데이트/모니터링 Security : Authentication, Identification, Secure Communication	-

Edge Platform Trends

- 솔루션 구분



구분	분류	특징	배포 방식	지원 Platform
MS, Azure IoT Edge	Computing	클라우드에서 ML 모델 학습 후 배포 머신 러닝, 스트림 데이터 질의 / 필터링 클라우드 기반 Edge 기기 관리	Microservice	Linux, Mac, Windows
AWS, Greengrass	Computing	AWS 기능(Lambda, Shadow)을 Edge에서 실행 클라우드에서 ML 모델 학습 후 배포	Proprietary (MQTT 통신)	Linux, RPi
Google IoT Edge, Apple 등	Computing	클라우드에서 ML 모델 학습 후 배포, 딥러닝 기반 AI 서비스 지향, H/W(NPU, GPU, 전용 AI Processor 등) 활용	SDK	Android iOS,macOS,tvOS
ARM, MBed Edge	Collecting	Edge와 연결된 기기의 Protocol Translation	Microservice	Linux, RPi

Edge Platforms

- Edge computing platform
  - Authentication and authorization
  - Offloading management
  - Location services
  - System monitoring
  - Resource management
  - Scheduling (VM, container)

- The key limitation to using edge devices effectively is the lack of a platform ecosystem that allows generic and **distributed applications** to be designed, deployed and executed on them

What technologies are needed at the Edge

- Containerization
- Orchestration
- AI & Machine Learning
- 5G
- Functional safety
- Time-sensitive Networking
- Secure device onboard

**Open Source Summit 2017, by Intel

Summary

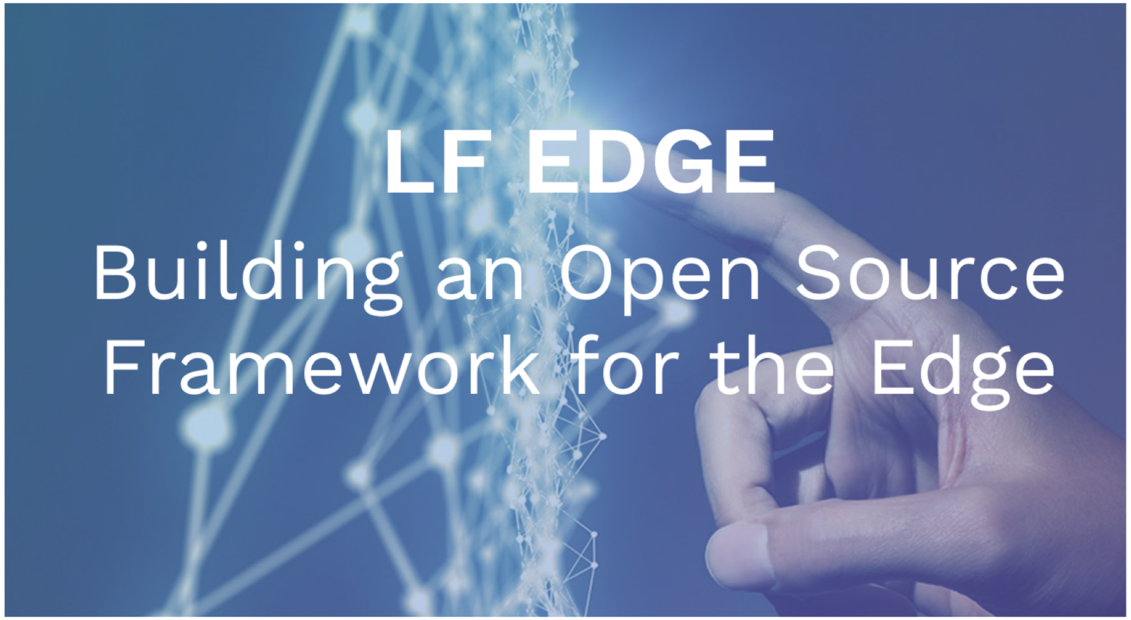
	EdgeX	Cloudlet	FogLamp	Flogo	Kura	Macchina	Kaa
Virtualization	Docker Container	VM, Openstack	VM	Docker Container	Java/OSGi	Linux	Sandbox, VM
Communities	Dell,	Open Edge Computing			Eclipse		Eclipe
Service area	IoT, IIoT	Mobile comm., IoT,	IoT, IIoT	IoT	IoT	IoT, IIoT	IoT
Applications	Java, Python, Golang, C/C++	Python,	REST, python, g++	Java	Java	C++, Java, Clang++, Python	Java, C, C++, Objective-C
Orchestrations	SL, security, management, etc.	Using Openstack	scalability, elasticity and resilience	Edge, App integration	Hardware, S/W platform integration	Hardware, S/W platform integration	Clustering, control,
Products	Dell, Chip						
License	Apache2	Apache2	Apache2	BSD 3-Clause License	Eclipse Public License	Apache2	Apache2

Open Source Edge/Fog Platform for IoT

- LF Edge
  - https://www.lfedge.org/
  - EdgeX (https://www.edgexfoundry.org/)
- FogLamp
  - http://dianomic.com/platform/foglamp/
- Flogo
  - https://www.flogo.io/
- Kura
  - https://www.eclipse.org/kura/
- Macchina
  - https://macchina.io/
- Kaa
  - https://www.kaaproject.org/
- Azure IoT Edge
  - https://azure.microsoft.com/en-us/services/iot-edge/

LF EDGE

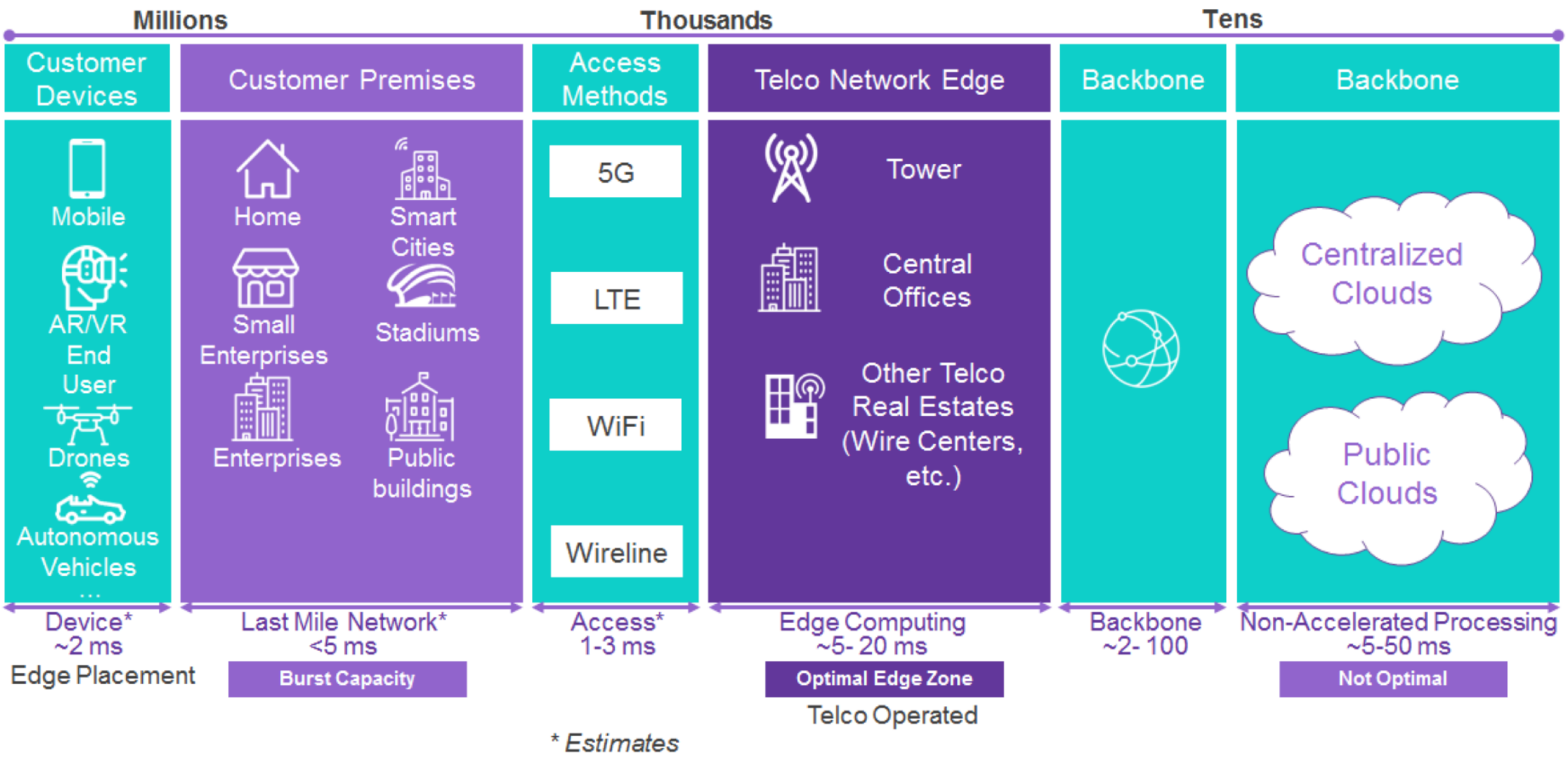
- Aims to establish an **open, interoperable framework** for edge computing independent of hardware, silicon, cloud, or operating system



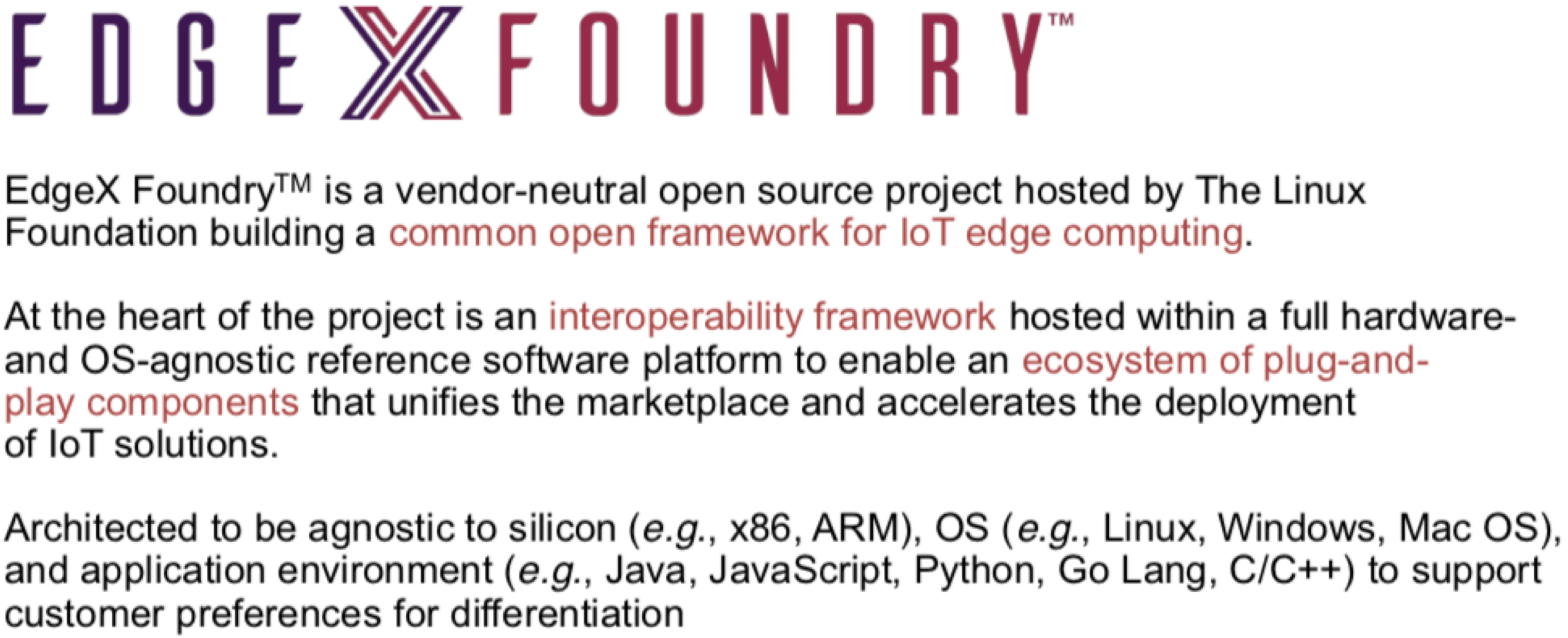
LF EDGE Project

- Akraino Edge Stack
  - 클라우드 백엔드에서 엣지 서비스를 실행을 위한 스택
- EdgeX Foundry
- **Home Edge**
  - Service layer for home service
- **Project EVE**
  - Edge Virtualization Engine
  - 사용자가 엣지 하드웨어를 가상화 하는 기능 지원
- Open Glossary of Edge Computing

Optimal Zone for Edge Placement



EdgeX



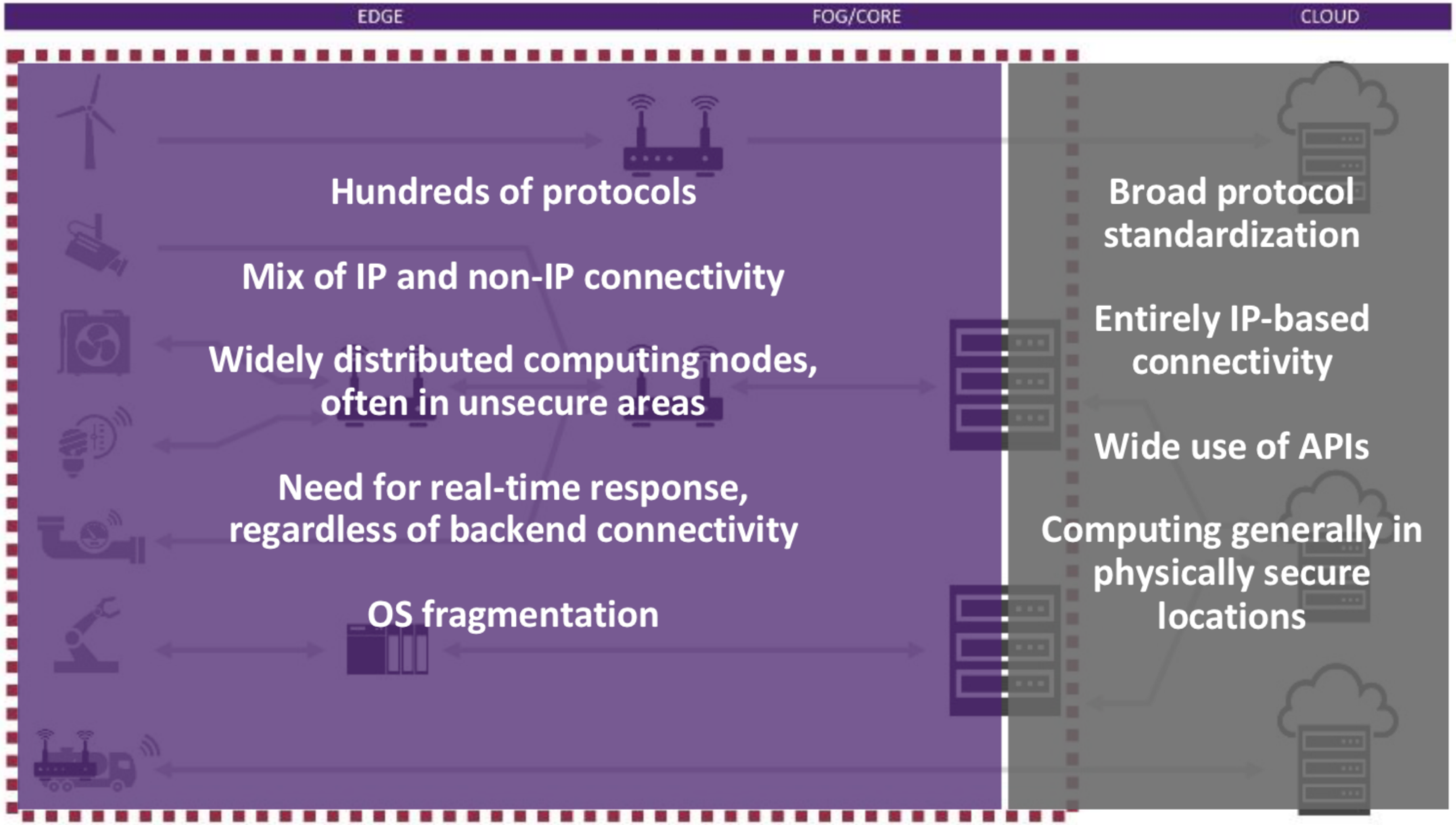
EDGEXFOUNDRY™

EdgeX Foundry™ is a vendor-neutral open source project hosted by The Linux Foundation building a common open framework for IoT edge computing.

At the heart of the project is an interoperability framework hosted within a full hardware- and OS-agnostic reference software platform to enable an ecosystem of plug-and-play components that unifies the marketplace and accelerates the deployment of IoT solutions.

Architected to be agnostic to silicon (*e.g.*, x86, ARM), OS (*e.g.*, Linux, Windows, Mac OS), and application environment (*e.g.*, Java, JavaScript, Python, Go Lang, C/C++) to support customer preferences for differentiation

Challenges



Hundreds of protocols

Mix of IP and non-IP connectivity

Widely distributed computing nodes, often in unsecure areas

Need for real-time response, regardless of backend connectivity

OS fragmentation

Broad protocol standardization

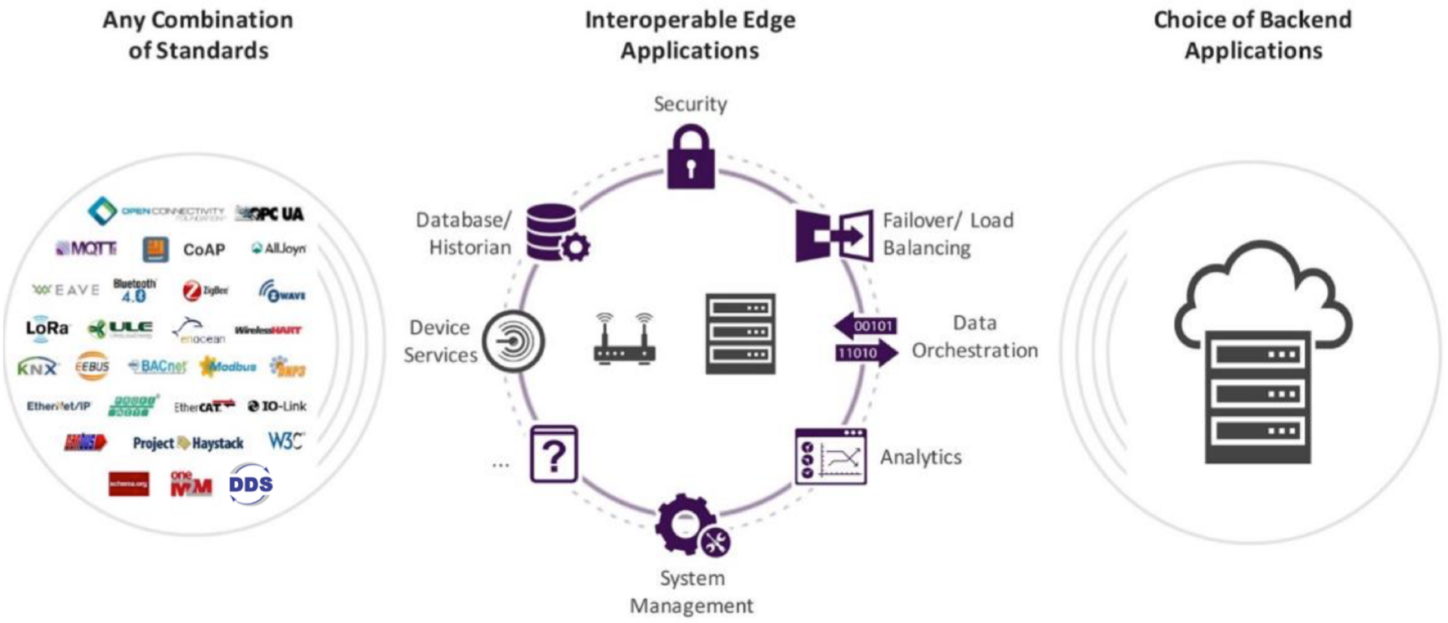
Entirely IP-based connectivity

Wide use of APIs

Computing generally in physically secure locations

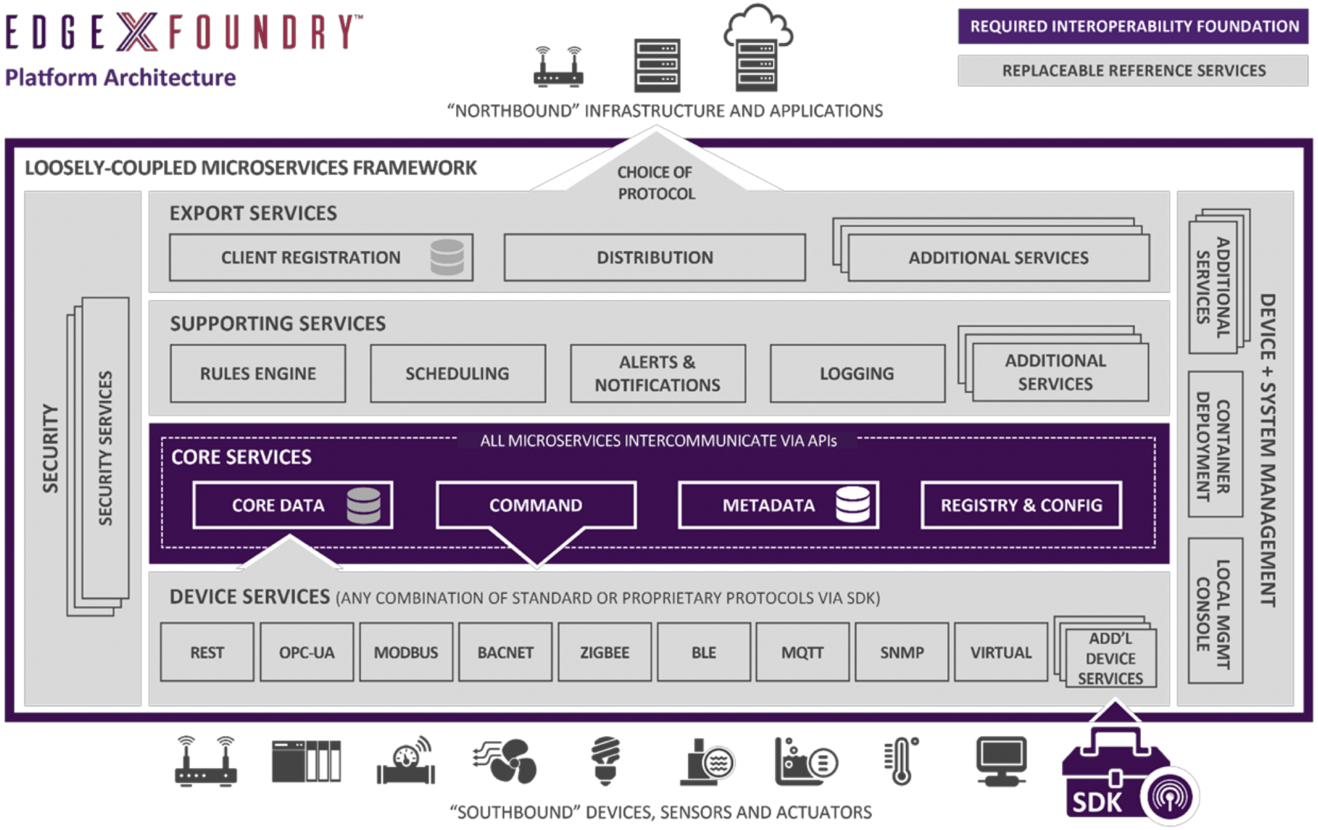
Support

- EdgeX Foundry
  - For user, developer
  - SDK for device services

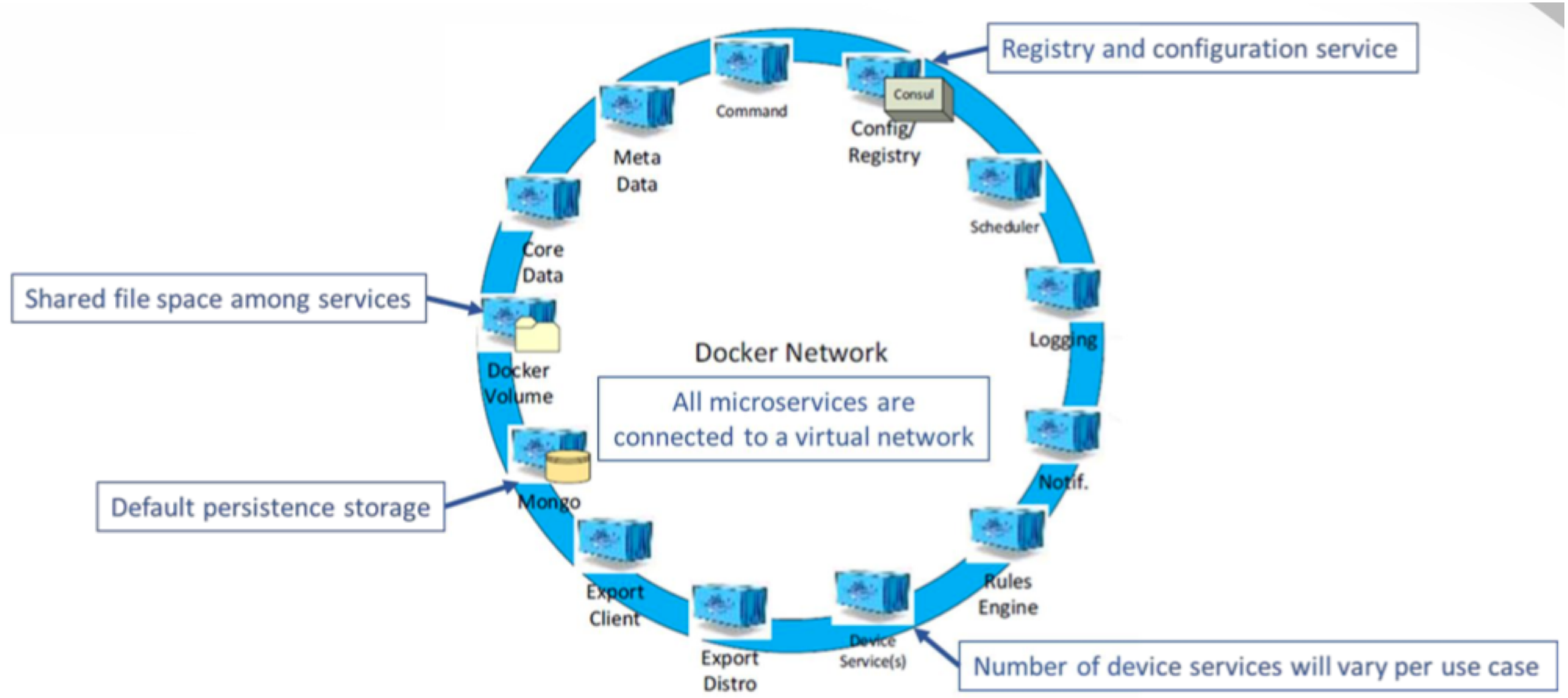


EdgeX Architecture

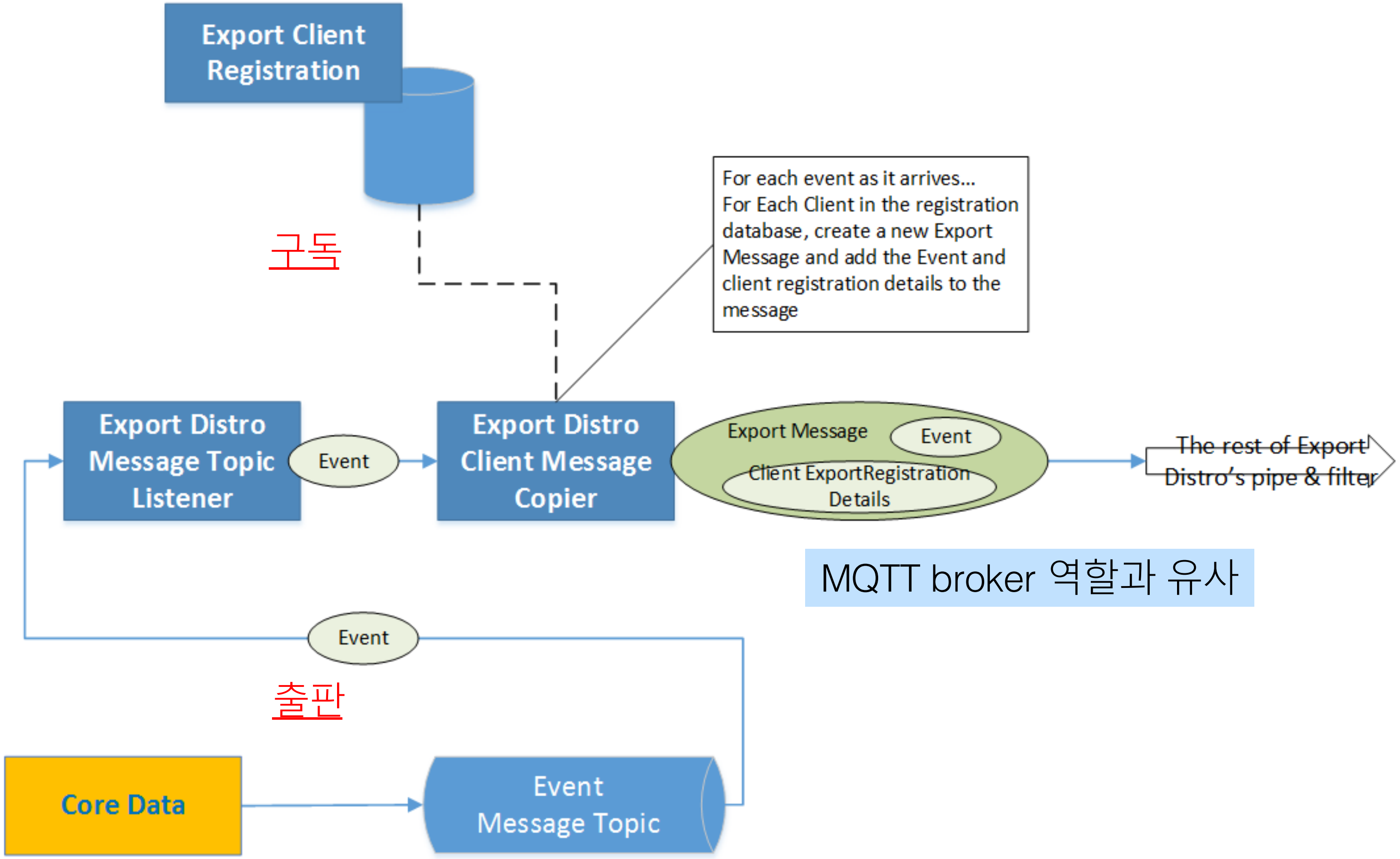
- Common open framework for IIoT Edge Platform



EdgeX 서비스 구성


Registry and configuration service
Consul
Command
Config/ Registry
Meta Data
Scheduler
Core Data
Shared file space among services
Logging
Docker Volume
Docker Network
All microservices are connected to a virtual network
Notif.
Default persistence storage
Mongo
Rules Engine
Export Client
Export Distro
Device Service(s)
Number of device services will vary per use case

Export Distribution

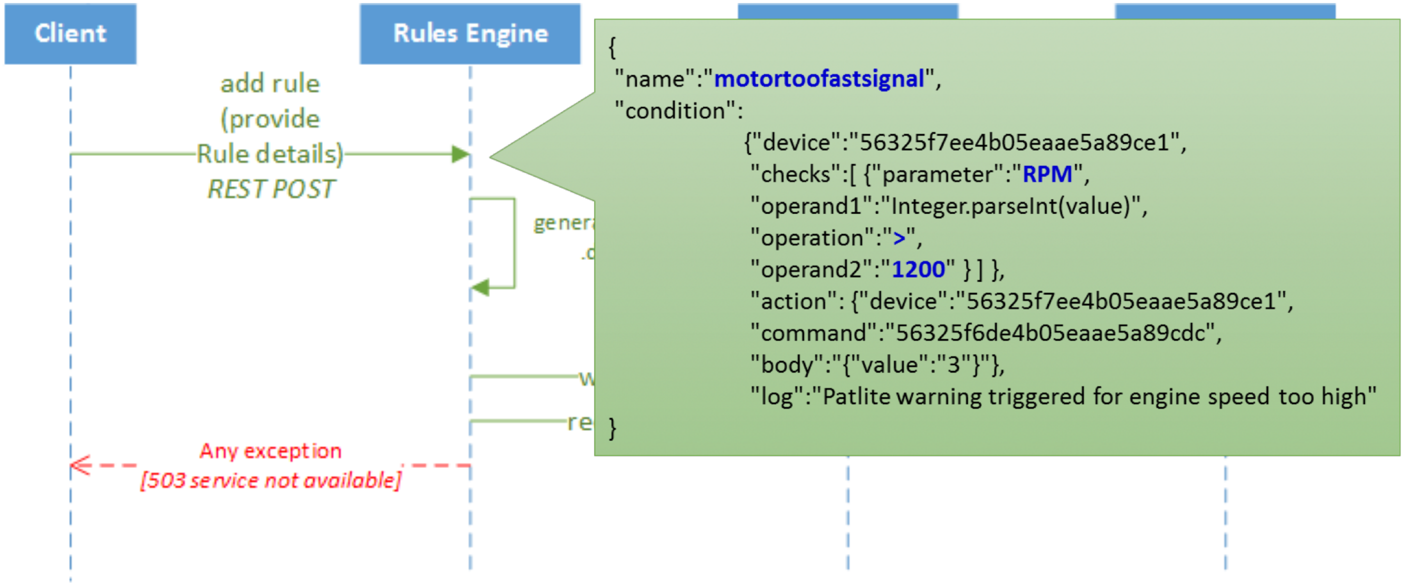


Export Client Registration
구독
For each event as it arrives...
For Each Client in the registration database, create a new Export Message and add the Event and client registration details to the message
Export Distro Message Topic Listener
Event
Export Distro Client Message Copier
Export Message
Event
Client ExportRegistration Details
The rest of Export Distro's pipe & filter
Event
출판
Core Data
Event Message Topic

MQTT broker 역할과 유사

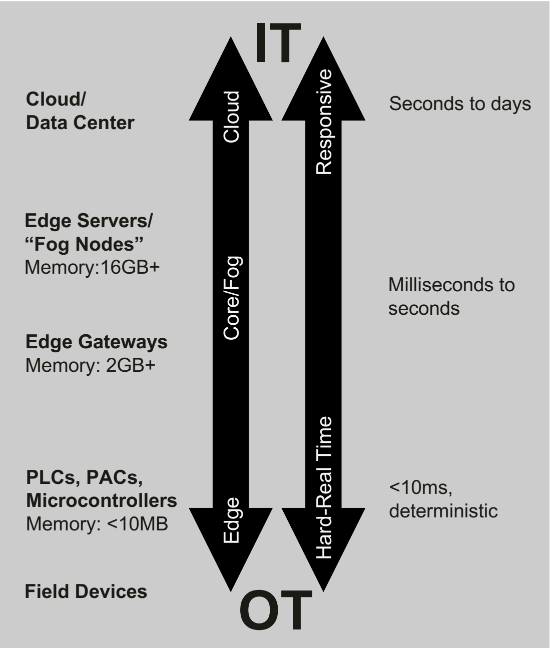
Rules Engine

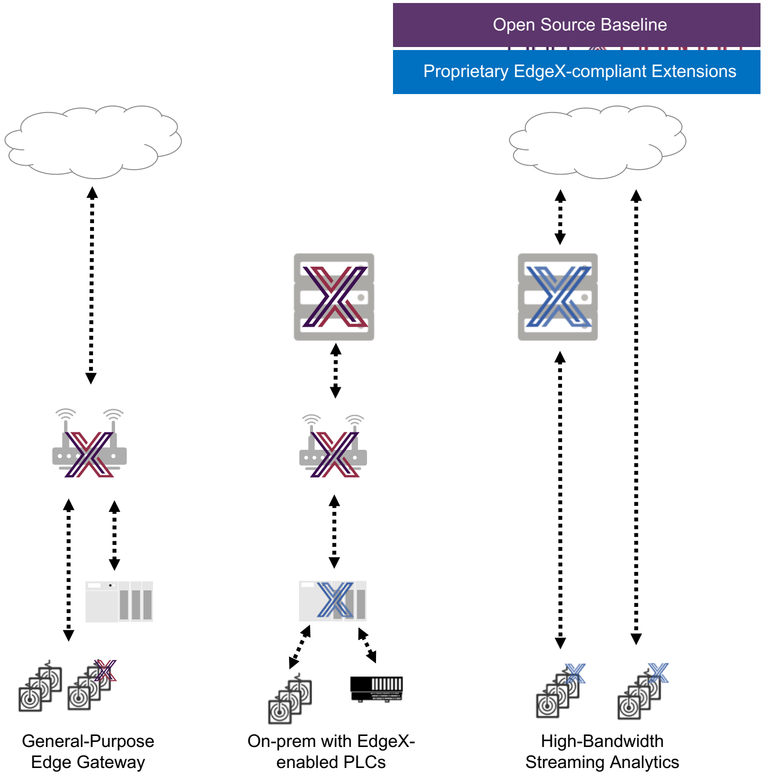
- “Intelligence” 서비스를 위한 빠른 응답 시간 제공
  - Cloud와 연동 없이 즉각적인 이벤트 처리
  - Open source rules engine provided by the JBoss community



```
{
 "name":"motortoofastsignal",
 "condition":
              {"device":"56325f7ee4b05eaae5a89ce1",
               "checks":[ {"parameter":"RPM",
               "operand1":"Integer.parseInt(value)",
               "operation":">",
               "operand2":"1200" } ] },
               "action": {"device":"56325f7ee4b05eaae5a89ce1",
               "command":"56325f6de4b05eaae5a89cdc",
               "body":"{"value":"3"}"},
               "log":"Patlite warning triggered for engine speed too high"
}
```

Summary of Example Use Cases





Real Time Enabled Via Code Extensions

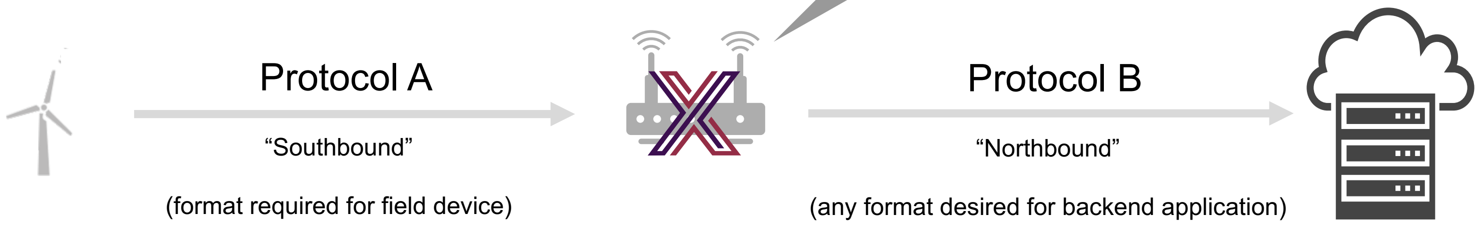
	Through Community Extensions	Open Source Baseline
	Real Time	**Soft Real Time (“Relevant Time”)**
Response Time	High bandwidth, QoS, sub-millisecond, deterministic	Milliseconds+
OS	RTOS	Traditional Linux or Windows
Example Use Cases	Smart Building, Energy Management, Factory Optimization, Predictive Maintenance, Quality Control, Supply Chain Management, Remote Asset Management, Fleet Management, Logistics, Environmental Monitoring	
	High-speed Process Control, Robotics, Safety Systems, Autonomous Vehicles	

Simple Linking Device

- A minimal deployment of EdgeX can function as a linking device which simply converts one protocol into another
- Typical protocol combinations vary by vertical and installation, some typical examples:
  - Energy: DNP3 to MQTT, Modbus to REST
  - Manufacturing: Profibus to OPC-UA
  - Buildings: BACnet MSTP (serial) to BACnet IP, MQTT, etc.

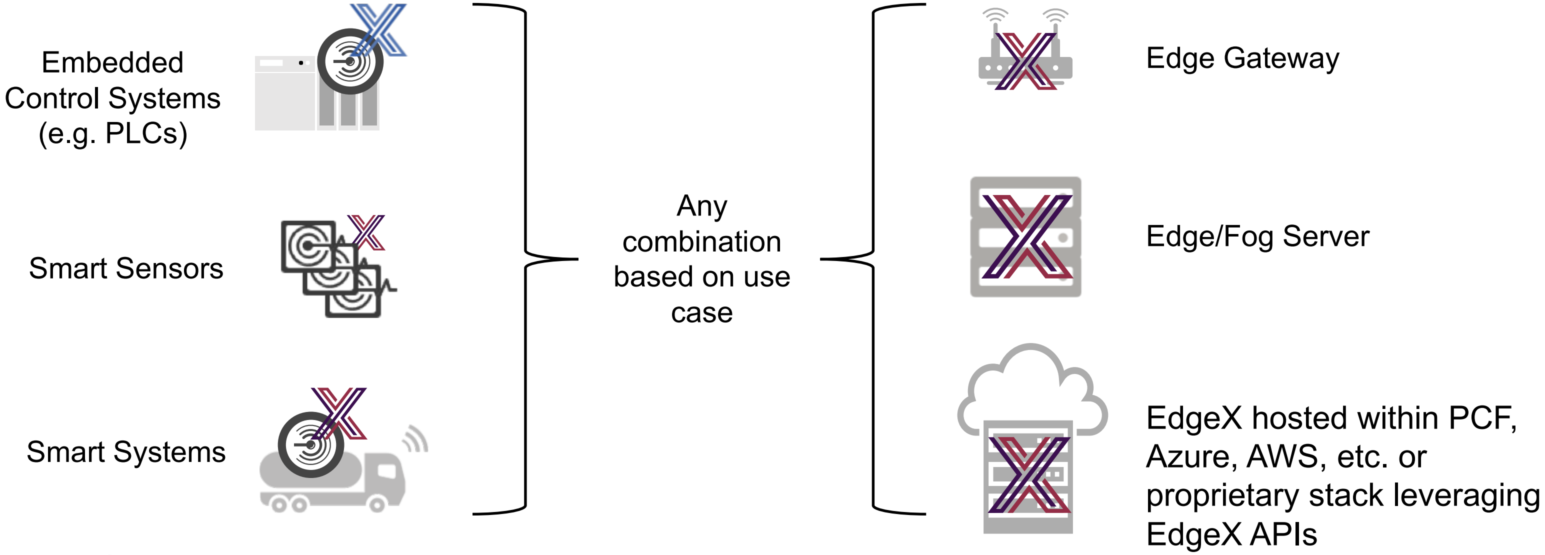
Deployed Microservices:

– Single Device Service
– Core Services
– Single Export Service
– Basic security and manageability


Protocol A
"Southbound"
(format required for field device)
Protocol B
"Northbound"
(any format desired for backend application)

Embedded Device Services

- Planned work will enable C-based Device Services to be embedded in constrained microcontrollers running a RTOS for real-time use cases (e.g. within a smart sensor or PLC)
- Due to loosely-coupled architecture, baseline EdgeX-compliant Device Services can be deployed directly on smart sensors or systems capable of hosting a microservice (via container or VM)
- IP-capable sensors with an EdgeX Device Service / APIs can communicate directly with Core Services running on any other compute node such as a gateway, server or directly to the cloud



Embedded Control Systems (e.g. PLCs)

Smart Sensors

Smart Systems

Any combination based on use case

Edge Gateway

Edge/Fog Server

EdgeX hosted within PCF, Azure, AWS, etc. or proprietary stack leveraging EdgeX APIs

Tiered Deployment in Smart Buildings

Number of deployed microservices and functionality increases higher in tier

Field Devices | Simple Edge GWs | Intelligent Edge GWs | Edge Servers | Cloud

Room Level

- Ingestion for local temperature and occupancy data
- Simple rules engine to control temperature and lighting settings

Floor Level

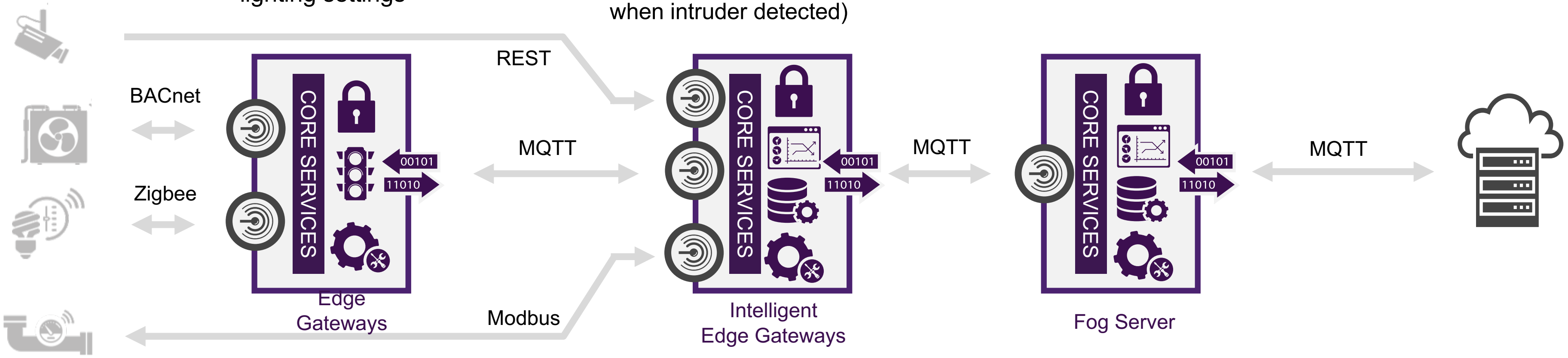
- Integration of temp and occupancy plus add'l events from surveillance cameras and overall energy usage data
- Basic ML/CEP for reacting to local events (e.g. alert security when intruder detected)

Building Level

- Aggregated data for analytics of overall building performance
- Streaming data from all floors, more complex analytics

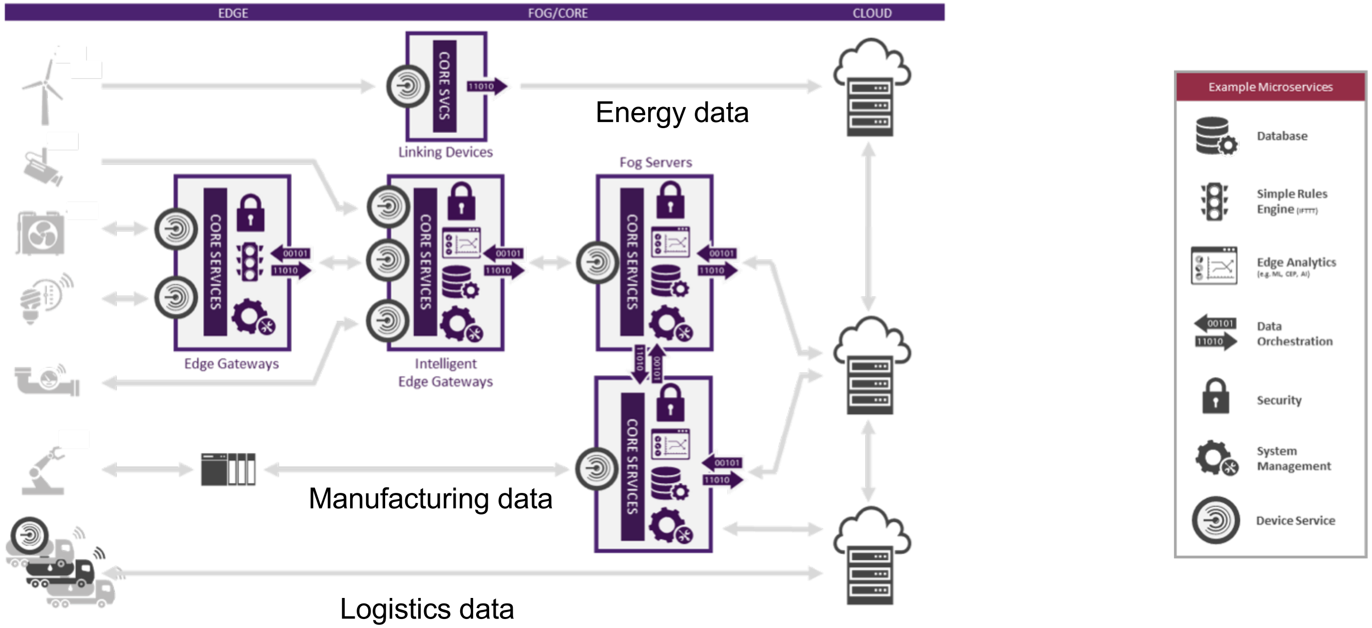
Portfolio Level

- Deep learning in the cloud to optimize energy usage across entire real estate portfolio



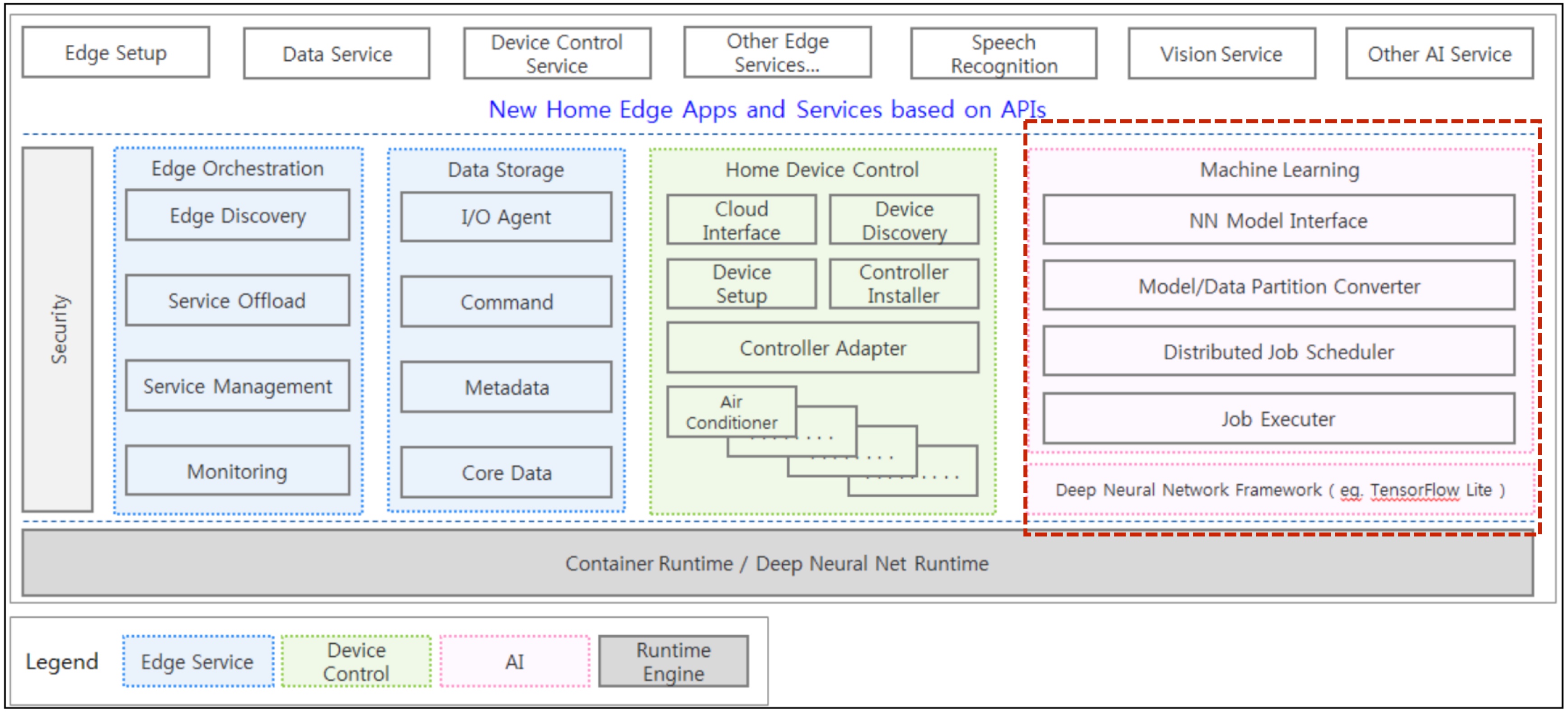
Distributed (e.g. 'Fog') Computing

- Introducing specific microservices to address QoS, failover between nodes, redundancy and "east-west" communication
- Workloads deployed dynamically at different tiers to optimize performance and results.
- In a manufacturing example, data can be coordinated for manufacturing process, building performance energy usage and logistics across various buildings, plants and trucks.



Home Edge Project

- Home Edge Project will provide users with
  - Interoperable, Flexible and Scalable edge computing services platform with a set of APIs that can also run with libraries and runtimes



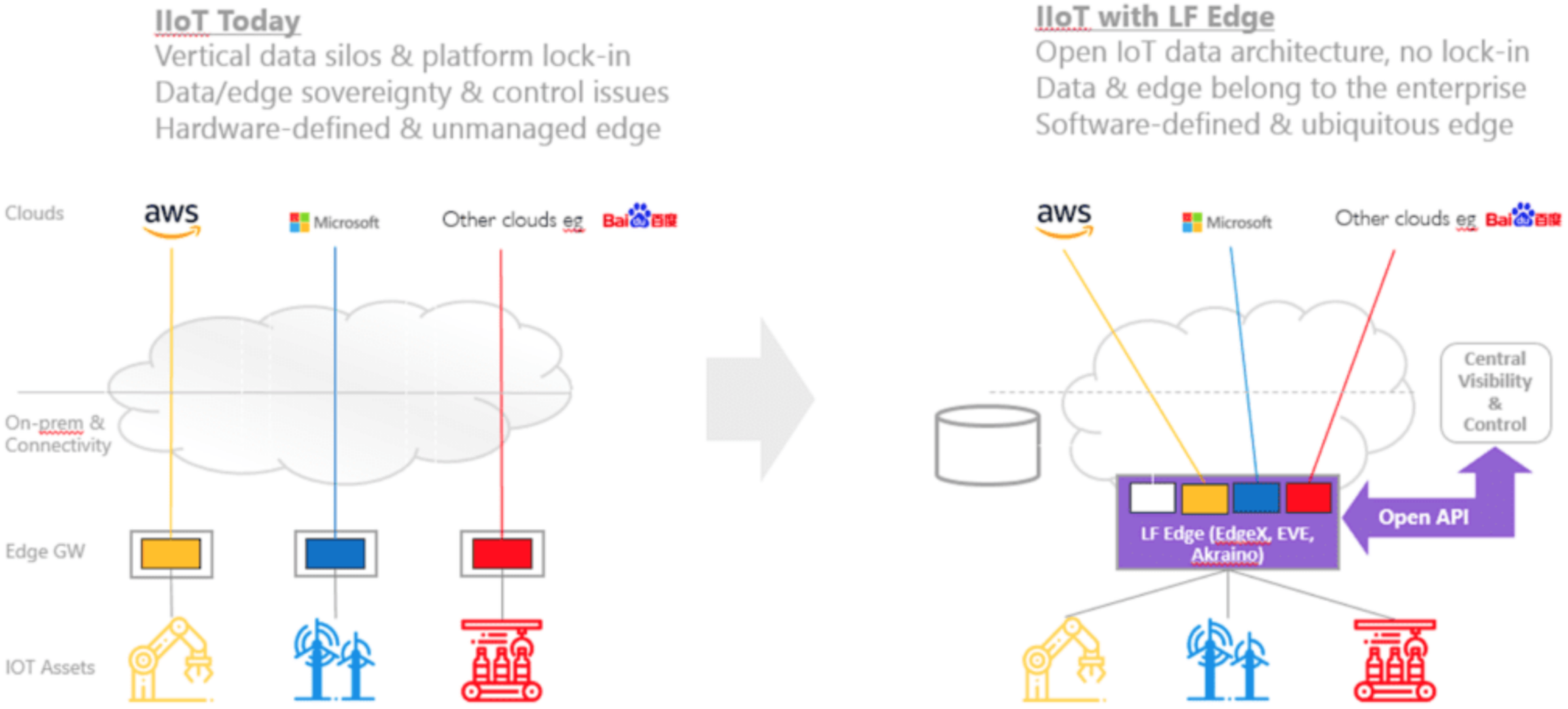
Home Edge Project Scope

- Define use cases, architecture and technical requirements
- Develop and maintain the features and APIs targeting Smart Home use cases and requirements in a manner of open source collaboration
- Upstream the core features back to the existing/upcoming projects under LF Edge
- Connect with Vertical Solutions WG on Smart Home in EdgeX, and Blueprint on Smart Home & Akraino through testbed validation

Technical Requirements

- Dynamic device and service discovery at "Home Edge"
- Quality of Service guarantee in various dynamic conditions (e.g. devices On/Off)
- **Distributed machine learning**
- Multi-vendor interoperability
- User privacy and secure services

Project EVE 필요성



IIoT Today

Vertical data silos & platform lock-in

Data/edge sovereignty & control issues

Hardware-defined & unmanaged edge

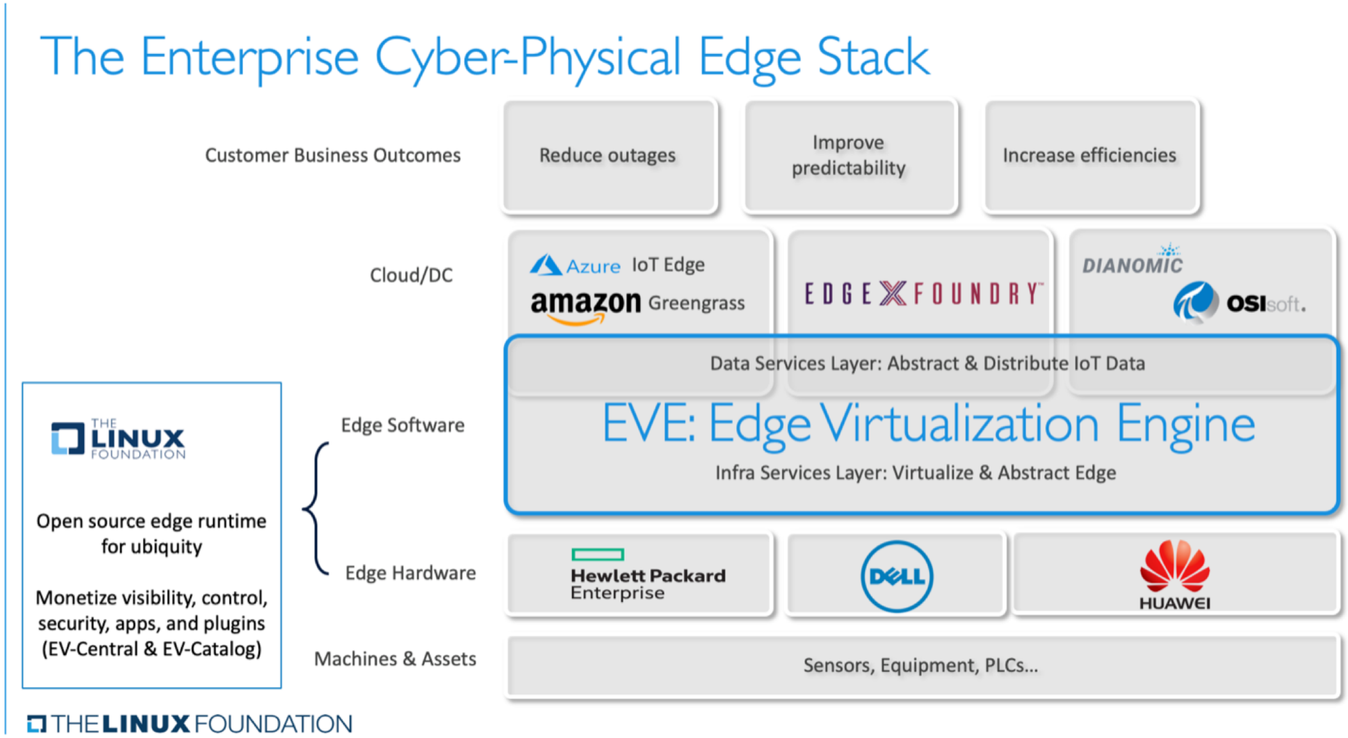
IIoT with LF Edge

Open IoT data architecture, no lock-in

Data & edge belong to the enterprise

Software-defined & ubiquitous edge

CPES



The Enterprise Cyber-Physical Edge Stack

Customer Business Outcomes: Reduce outages | Improve predictability | Increase efficiencies

Cloud/DC: Azure IoT Edge, amazon Greengrass | EDGE X FOUNDRY™ | DIANOMIC, OSIsoft.

Data Services Layer: Abstract & Distribute IoT Data

EVE: Edge Virtualization Engine

Infra Services Layer: Virtualize & Abstract Edge

Edge Software

Edge Hardware: Hewlett Packard Enterprise | DELL | HUAWEI

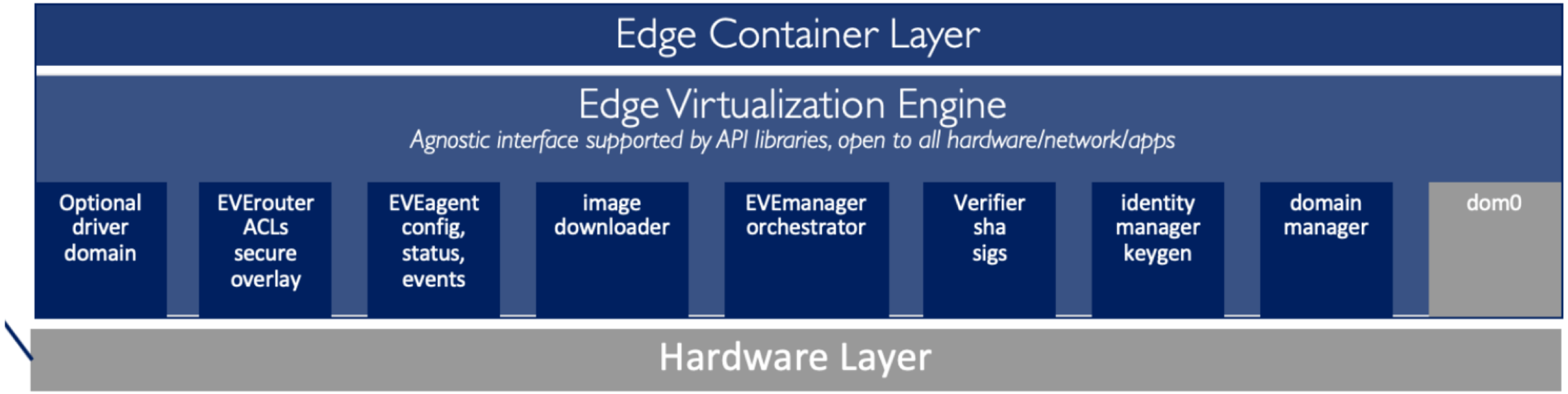
Machines & Assets: Sensors, Equipment, PLCs...

THE LINUX FOUNDATION

Open source edge runtime for ubiquity

Monetize visibility, control, security, apps, and plugins (EV-Central & EV-Catalog)

THE LINUX FOUNDATION



Edge Container Layer

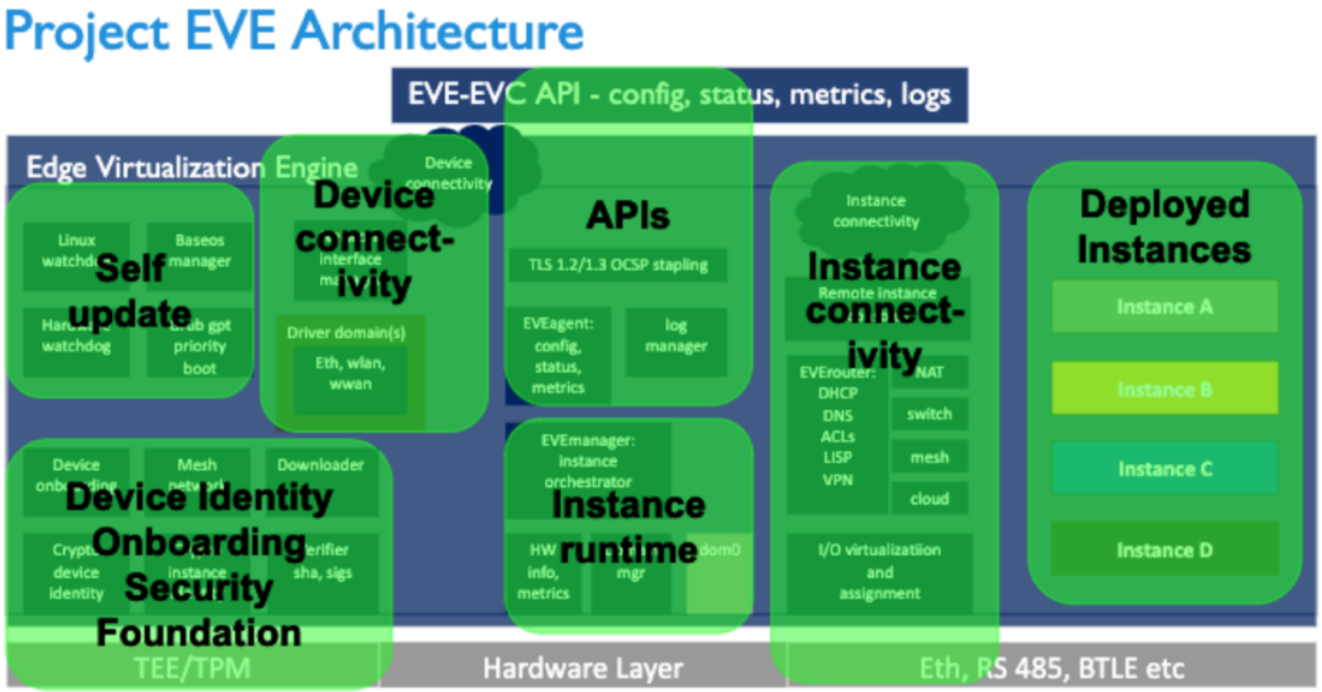
Edge Virtualization Engine

Agnostic interface supported by API libraries, open to all hardware/network/apps

Optional driver domain | EVErouter ACLs secure overlay | EVEagent config, status, events | image downloader | EVEmanager orchestrator | Verifier sha sigs | identity manager keygen | domain manager | dom0

Hardware Layer

Proposed EVE Architecture



Open Edge Computing (Cloudlet)

- Small data center at the edge of the Internet (many sizes & forms)
  - One wireless hop (+fiber or LAN) to mobile devices (Wi-Fi or 4G LTE or 5G)
  - Multi-tenant, as in cloud
  - Good isolation and safety (VM-based guests)
  - Lighter-weight containers (e.g. Docker within VMs) also possible

- Non-constraints (relative to mobile devices)
  - Energy
  - Weight/size/heat

http://openedgecomputing.org

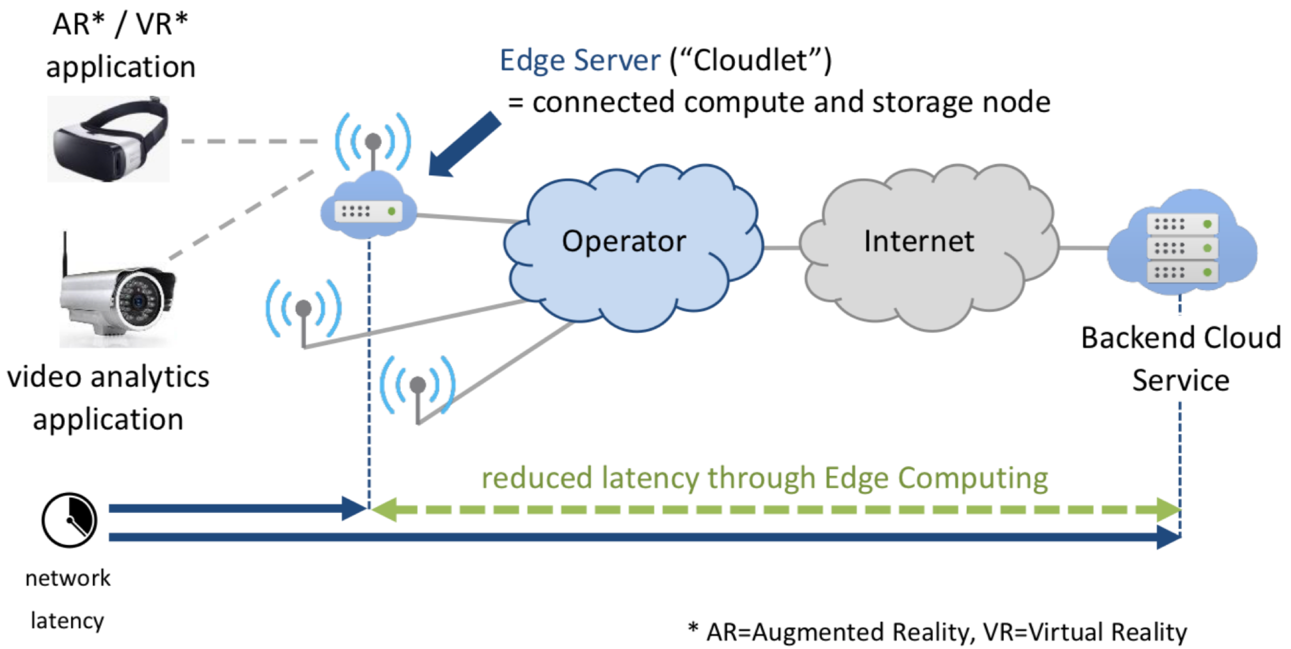
Cloudlet

Edge Computing

A new network feature that offers connected compute and storage resources (a.k.a. Cloudlets) right next to you – wherever you are!

Benefits

- ✓ **Ultra-low latency**: disruptive improvement of customer experience
- ✓ **Reduction of data traffic**: cloud services (e.g., big data analytics) right at the edge



* AR=Augmented Reality, VR=Virtual Reality

LIVING EDGE LAB

- BUILDING AN OPEN AND FLEXIBLE LAB FOR EDGE COMPUTING

Mission Statement

"We are building a **real-world testbed** for Edge Computing with leading **edge applications** and user **acceptance testing**."

LEL

Our Way Forward in 2017

Infrastructure, telco and research team up and **build testbeds**

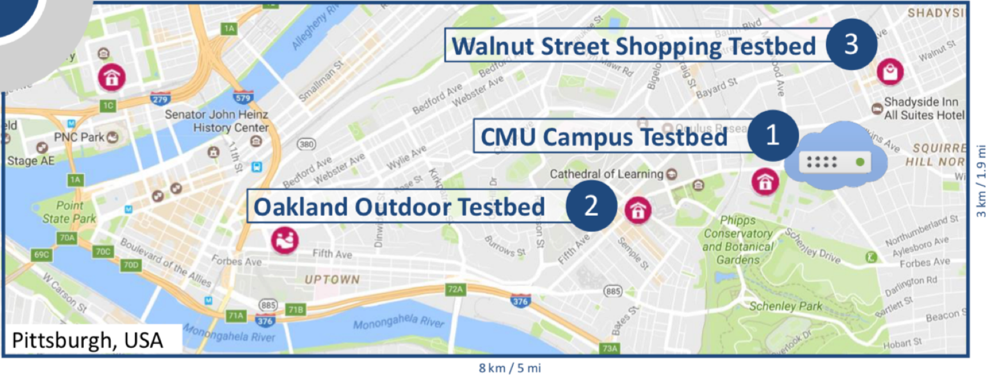
Application partners join the lab for dedicated test projects

Integration and testing of latest edge computing applications

Joint **evaluation and promotion** of results among partners

Key Elements

- **Partnership:** developers for apps, services and devices join forces with telco, infrastructure and research
- **Test Diversity:** various testbeds and latest technology available for a variety of use-case scenarios
- **Open Platform:** edge computing based on OpenStack, vendor & operator independent



EDGE COMPUTING OFFERS A WIDE RANGE OF APPLICATIONS

EDGE COMPUTING brings cloud computing closer to mobile users. This enables applications with a demand for both **LOW LATENCY** and **HIGH BANDWIDTH**.

Almost all industries could benefit from this, e.g., by building **AUGMENTED** or **VIRTUAL REALITY** apps, services and devices that become much more feasible with the edge.

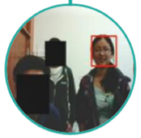
The Living Edge Lab community has implemented a broad spectrum of **DEMO USE CASES**. Examples can be found below. **MANY MORE SCENARIOS** will be supported due to the urban environment of our testbeds in downtown Pittsburgh:

EDUCATION | **HEALTHCARE** | **TRAFFIC** | **LIVE SPORTS** | **DRIVING** | **ROBOTICS**

SHOPPING

Window Shopping App ⏱
Live video analysis and enriched augmented reality with [handset] or [glasses].

PRIVACY



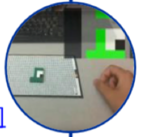
Video Privacy Filter ⏱
Face denaturing is applied to video streams using face re-cognition on a Cloudlet. [Video]

SPORTS & CREATIVITY

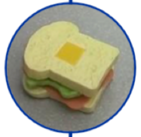
Table Tennis Assistant ⏱
Combines smart glasses, object recognition, motion prediction and real-time instructions. [Video]

Drawing Assistant
Combines object recognition and corrective real-time feedback to the user. [Video]

PRODUCTIVITY



Lego® Assembly
Combines object recognition with giving instructions in real-time. [Video]



Sandwich Assistant
Pre-learned sandwich slice recognition offers instructions to users in real-time. [Video]

IKEA Assembly Assistant
Holograms assist during furniture assembly using smart glasses and object recognition. [Video]

MEDICINE

AR Surgery Assistant ⏱
Surgery of a broken rib is assisted by image recognition and holographic guidance. [Video]

RENDERING

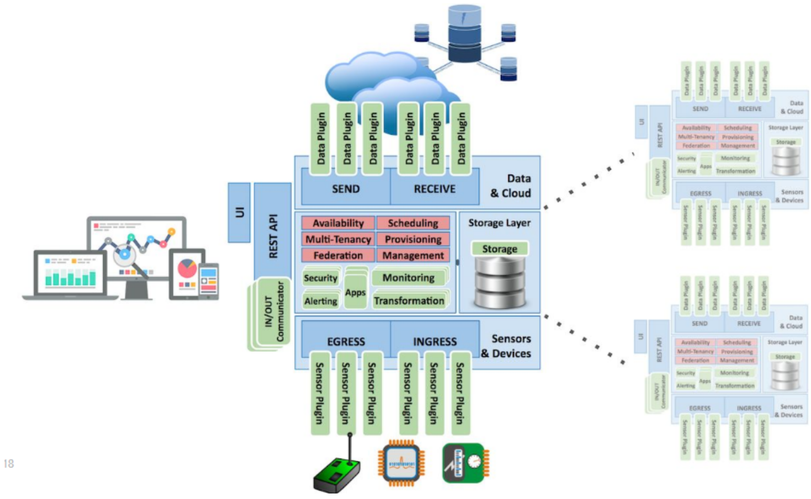
User Experience Demo ⏱
A compute-intensive simulation of particle movement shows how a [Cloudlet] performs vs [Cloud].

⏱ latency-sensitive application

http://openedgecomputing.org

FogLAMP

- OSIsoft Open Edge Module
  - An open source platform for IoT and an essential component in Fog Computing.
- Micro-service architecture
  - Provide the FogLAMP
    - Distribution
    - Scale-out/-up
    - Resilience
    - Best language/deployment



http://dianomic.com/platform/foglamp/

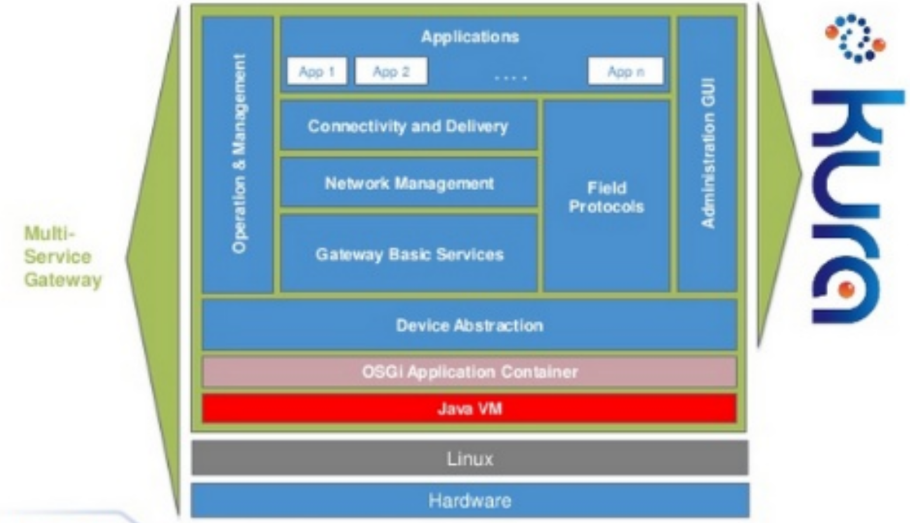
Flogo

- Ultralight Edge Micro-services Framework
  - Go-based open source ecosystem for building event-driven apps
  - Build **IoT applications that run on edge devices** and quickly integrate them with **IoT gateways** and **cloud services** like AWS™ IoT or Microsoft Azure® IoT Hub.
  - The lightest way to connect IoT devices. Up to 20 to 50 times lighter than Node.js and Java® Dropwizard.
  - Serverless compute
    - Infinitely scale Flogo's ultralight functions
  - **Edge Machine Learning**
    - Run Machine learning models including **Google TensorFlow** as Flogo activities on tiny edge devices

http://www.flogo.io/

Kura

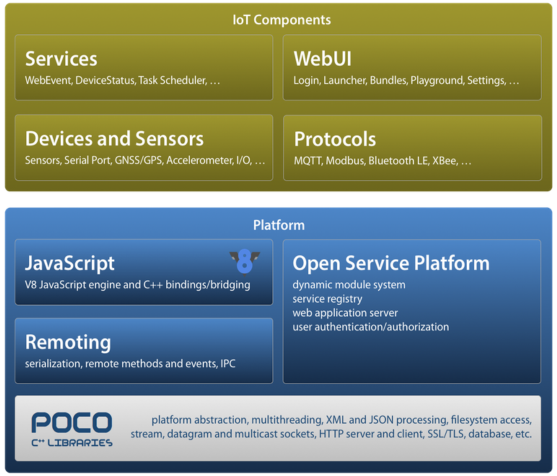
- **Focus on IoT Gateway**
- Set of Java and OSGi services
  - Including I/O services, Data Services, Cloud Services, Networking, etc.
  - Web UI for Configuration (Devices, Network, Protocols, etc.)
- Eclipse Public License 1.0
- Eclipse IoT Projects
  - IoT Gateway --> Eclipse Kura
  - Services like Eclipse SmartHome
  - Standard implementations like Mosquitto (**MQTT Server**)
  - Connectivity via Eclipse Paho (MQTT Client)
- Apache Camel Connector
  - Integration Framework (**connectivity, enterprise integration patterns**)



http://www.eclipse.org/kura

Macchina.io

- A toolkit(Software framework) for **embedded IoT edge and fog computing applications** that connect sensors, devices and cloud services.
  - Quickly building device applications for the Internet of Things running on **Linux-based devices like the Raspberry Pi, Beaglebone or MangOH**.



IoT Components

Services
WebEvent, DeviceStatus, Task Scheduler, …

WebUI
Login, Launcher, Bundles, Playground, Settings, …

Devices and Sensors
Sensors, Serial Port, GNSS/GPS, Accelerometer, I/O, …

Protocols
MQTT, Modbus, Bluetooth LE, XBee, …

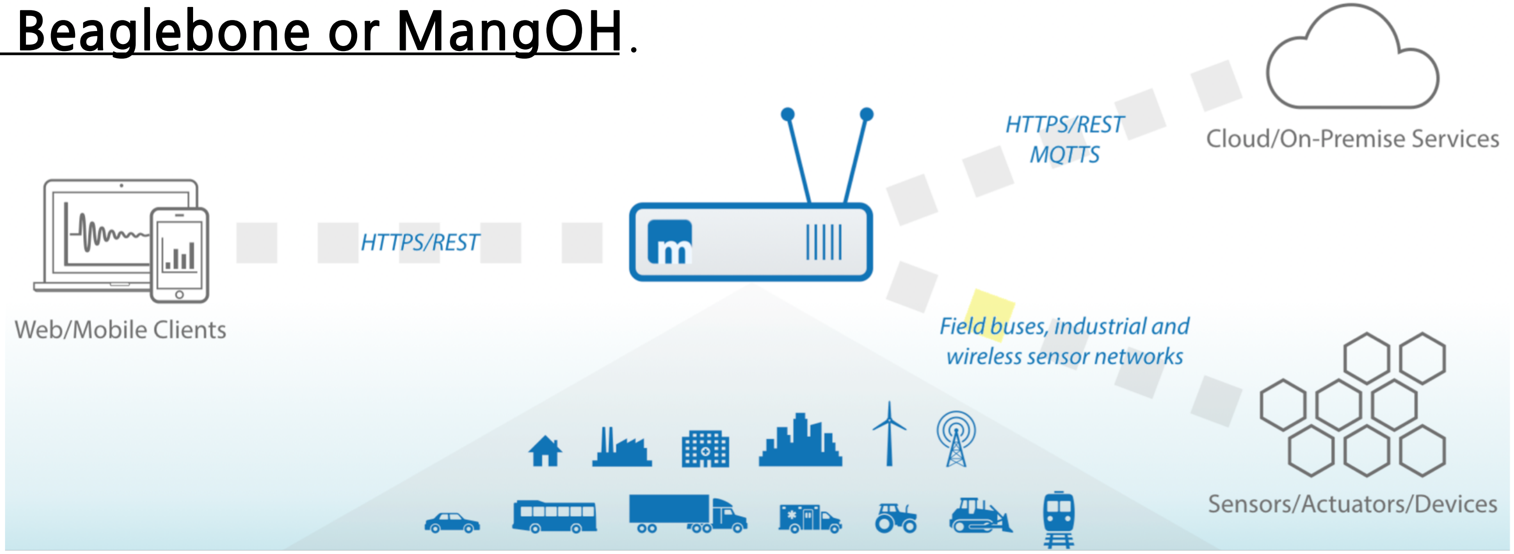
Platform

JavaScript
V8 JavaScript engine and C++ bindings/bridging

Open Service Platform
dynamic module system
service registry
web application server
user authentication/authorization

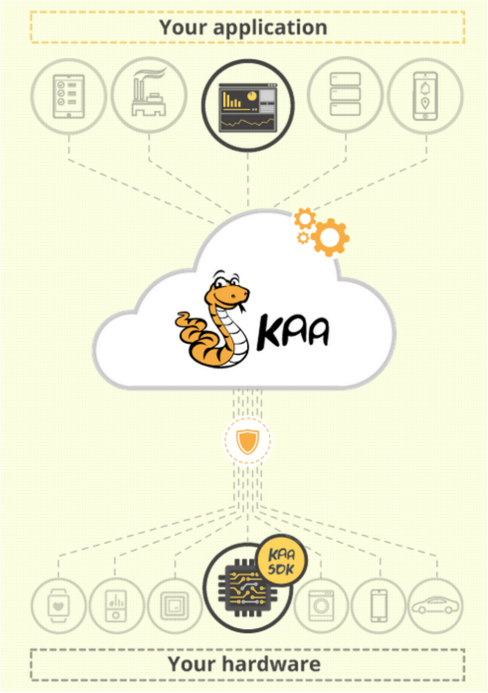
Remoting
serialization, remote methods and events, IPC

POCO C++ LIBRARIES
platform abstraction, multithreading, XML and JSON processing, filesystem access, stream, datagram and multicast sockets, HTTP server and client, SSL/TLS, database, etc.

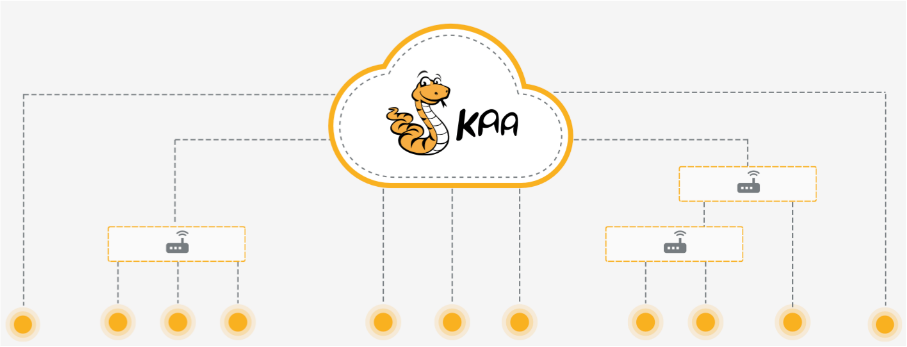

HTTPS/REST
MQTTS
Cloud/On-Premise Services
HTTPS/REST
Web/Mobile Clients
Field buses, industrial and
wireless sensor networks
Sensors/Actuators/Devices

Kaa as IoT Edge **Analytic** platform

- Kaa Platform
  - Kaa server is the back-end part of the platform
  - Kaa extensions are independent software modules that improve the platform functionality
  - Endpoint SDK is a library that provides client-side APIs
  - Gateway support



https://www.kaaproject.org/overview/



Distributed Analytics in Fog Platform

- Implementation of Fog computing platform using TensorFlow and Kubernetes
  - Enhance open source projects
  - Docker -> Dynamic deployment
  - Kubernetes -> Management
  - TensorFlow -> Real-time analytics

docker kubernetes TensorFlow