01236255

Introduction to Internet of Things

Lecture 2 : IoT Architecture

IoT System and Information Engineering, KMITL

Dr. Thanavit Anuwongpinit

Evolution from Embedded Systems to IoT and Cyber Physical Systems (CPS)

IoT

- Sensing of the physical world
- Internet connectivity
- Used in China and EU

CPS

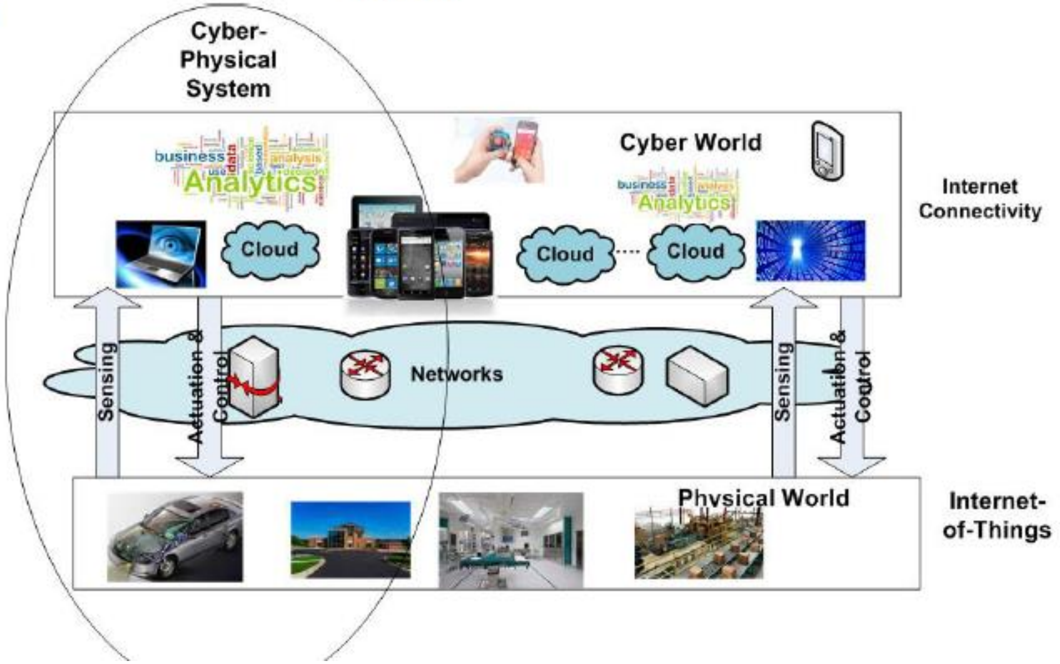
- Control of combined organizational and physical processes
- Tight human machine interaction
- Used in USA and EU

Embedded Systems (e.g., Board)

Networked Embedded Systems

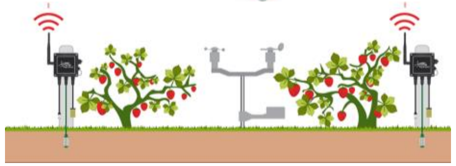
Cyber-Physical Systems (e.g., networked distributed traffic)

Internet of Things (e.g., Smart City Transport)



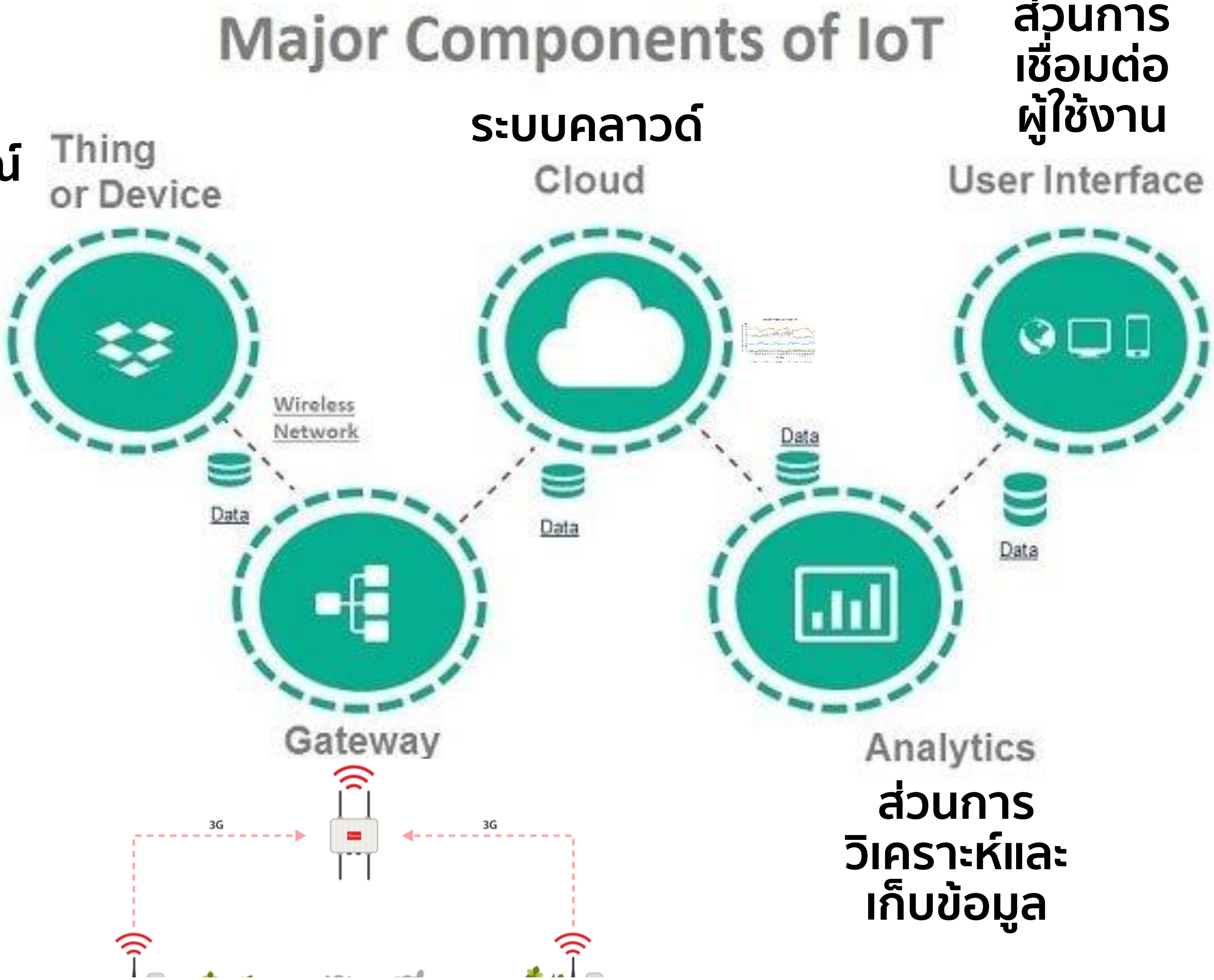
Example : Smart Farming

อุปกรณ์

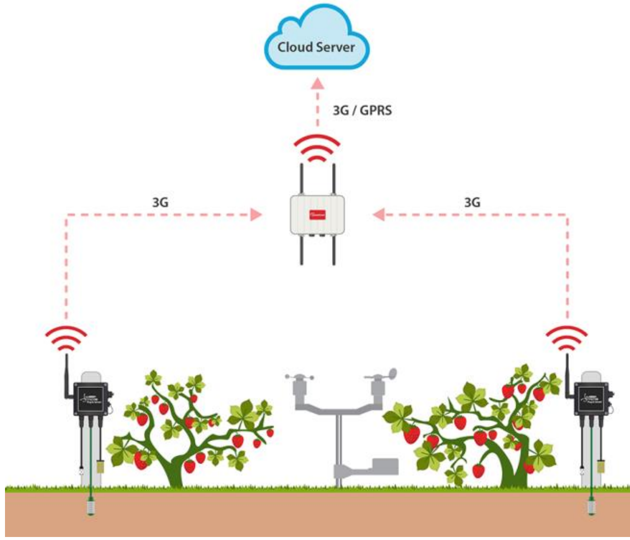


ระบบคลาวด์

ส่วนการ
เชื่อมต่อ
ผู้ใช้งาน



ส่วนการ
วิเคราะห์และ
เก็บข้อมูล



1. อุปกรณ์ (Device, Things)

- ส่วนแรกของระบบไอโอที คือส่วนของอุปกรณ์ (Device) ซึ่งจะเชื่อมต่อกับสิ่งต่าง ๆ
- การได้มาซึ่งข้อมูล จากสภาพแวดล้อมหรือทางกายภาพที่ได้ตรวจรู้ด้วยอุปกรณ์การรับรู้ (Sensor)
- เชื่อมต่อกับอุปกรณ์ควบคุมต่าง ๆ หรืออาจเรียกได้ว่า เป็น Local Processing and storage devices เพื่อส่งข้อมูลผ่านระบบเครือข่ายไปยังระบบคลาวด์หรือเซิฟเวอร์อีกทีหนึ่ง
- หากอุปกรณ์รับคำสั่ง (Command) มาจากเครือข่ายหรือระบบคลาวด์ จะต้องการสั่งการ เพื่อเป็นอุปกรณ์สั่งการ (Actuator)

Things

Sensor

(Input Devices)

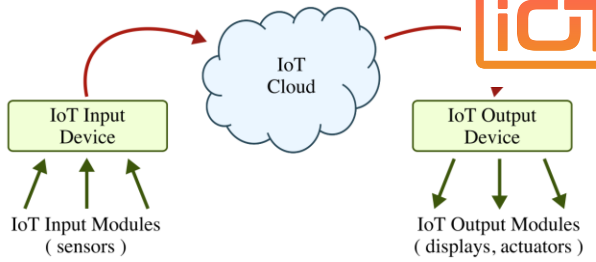
Actuator

(Output Devices)

Controller

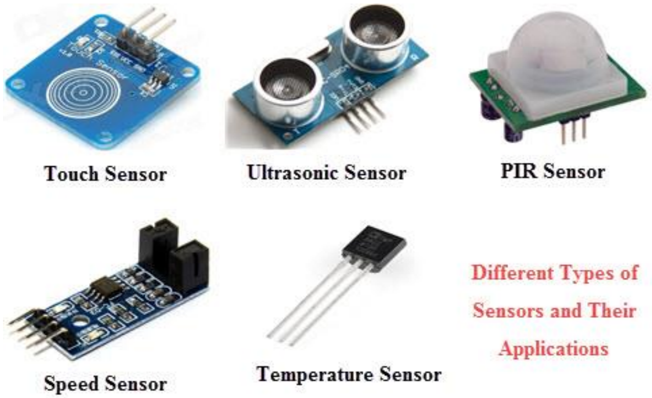
(Local Processing)

1. อุปกรณ์ (Device, Things)



Device Interfacing

- คือการเชื่อมต่อสื่อสาร (Interface) ระหว่างสิ่งต่าง ๆ (Things, Device) หรือระหว่างอุปกรณ์ (Sensor, Actuator) กับอุปกรณ์ควบคุม (Controller)



Digital Input

Serial Communications (UART)

Analog to Digital Convertor (ADC, A2D)

Inter-Integrated Circuit (I2C)

Industrial Protocols : ModbusRTU, ModbusTCP, 4-20ma

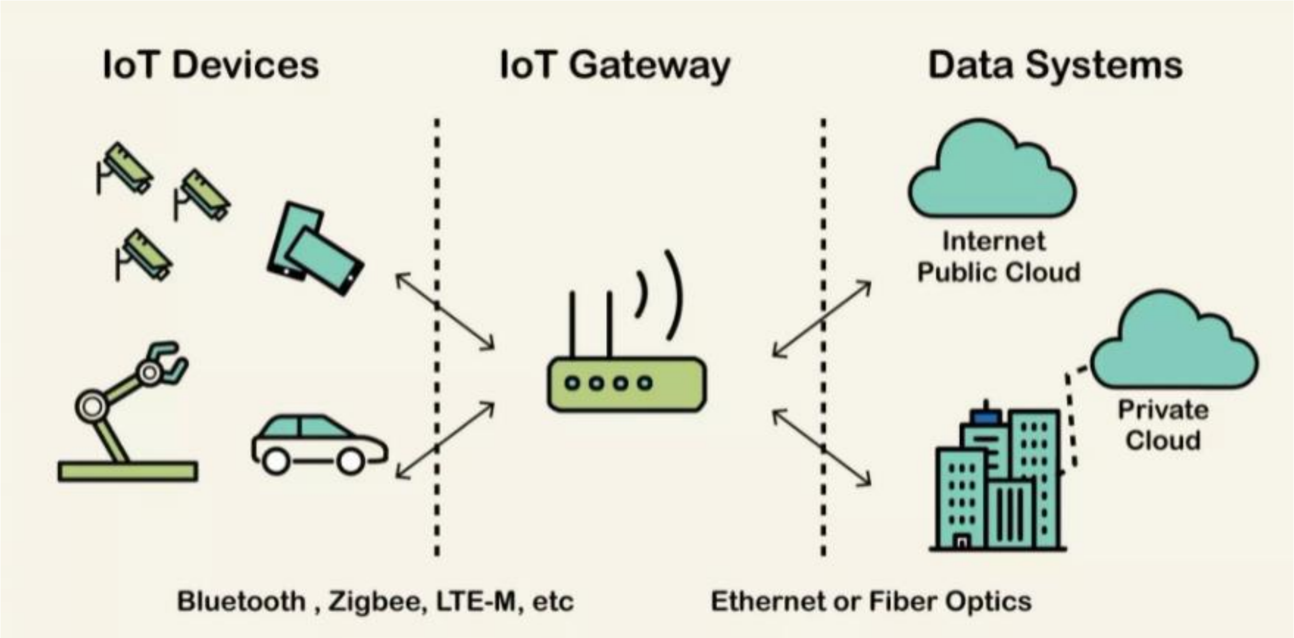
Digital Output

Analog Output (Pulse Width Modulation : PWM)

Driver

2. เกตเวย์ การเชื่อมต่อสื่อสาร (Gateway, Connectivity)

- เกตเวย์ของการเชื่อมต่อสื่อสาร คือ ฮาร์ดแวร์ที่รับข้อมูลมาจากอุปกรณ์ควบคุมและเซนเซอร์แล้วทำการประมวลผลหรือส่งข้อมูลต่อไปยังคลาวด์ หรือรับข้อมูลจากคลาวด์เพื่อส่งต่อไปยังอุปกรณ์
- โดยการเชื่อมต่อสื่อสารของเกตเวย์ ไม่ว่าจะเป็นการสื่อสารแบบมีสาย (Wired) หรือแบบไร้สาย (Wireless) เพื่อส่งข้อมูลไปยังคลาวด์ผ่าน Internet Protocol หรือรับข้อมูลจากคลาวด์เพื่อสั่งการไปยังอุปกรณ์ต่อไป

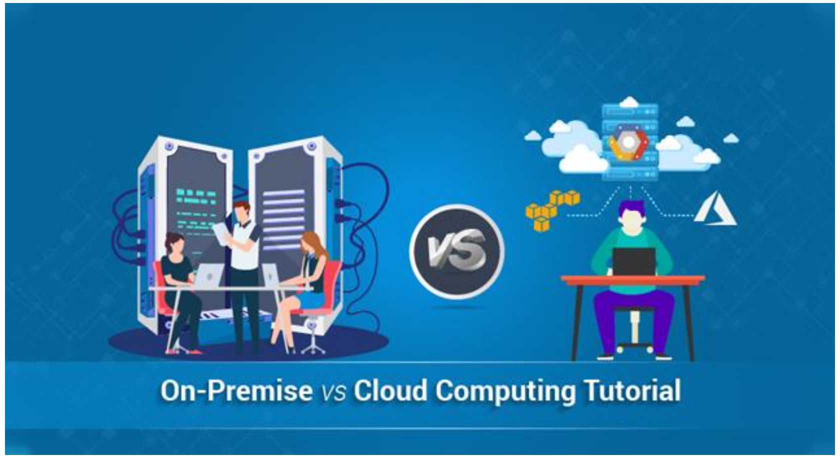


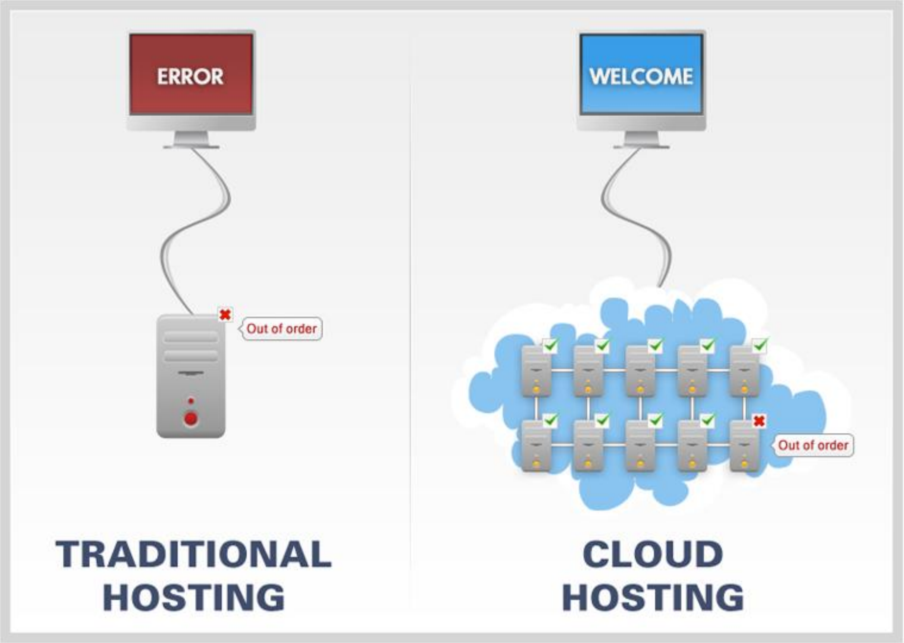
2. เกตเวย์ การเชื่อมต่อสื่อสาร (Gateway, Connectivity)

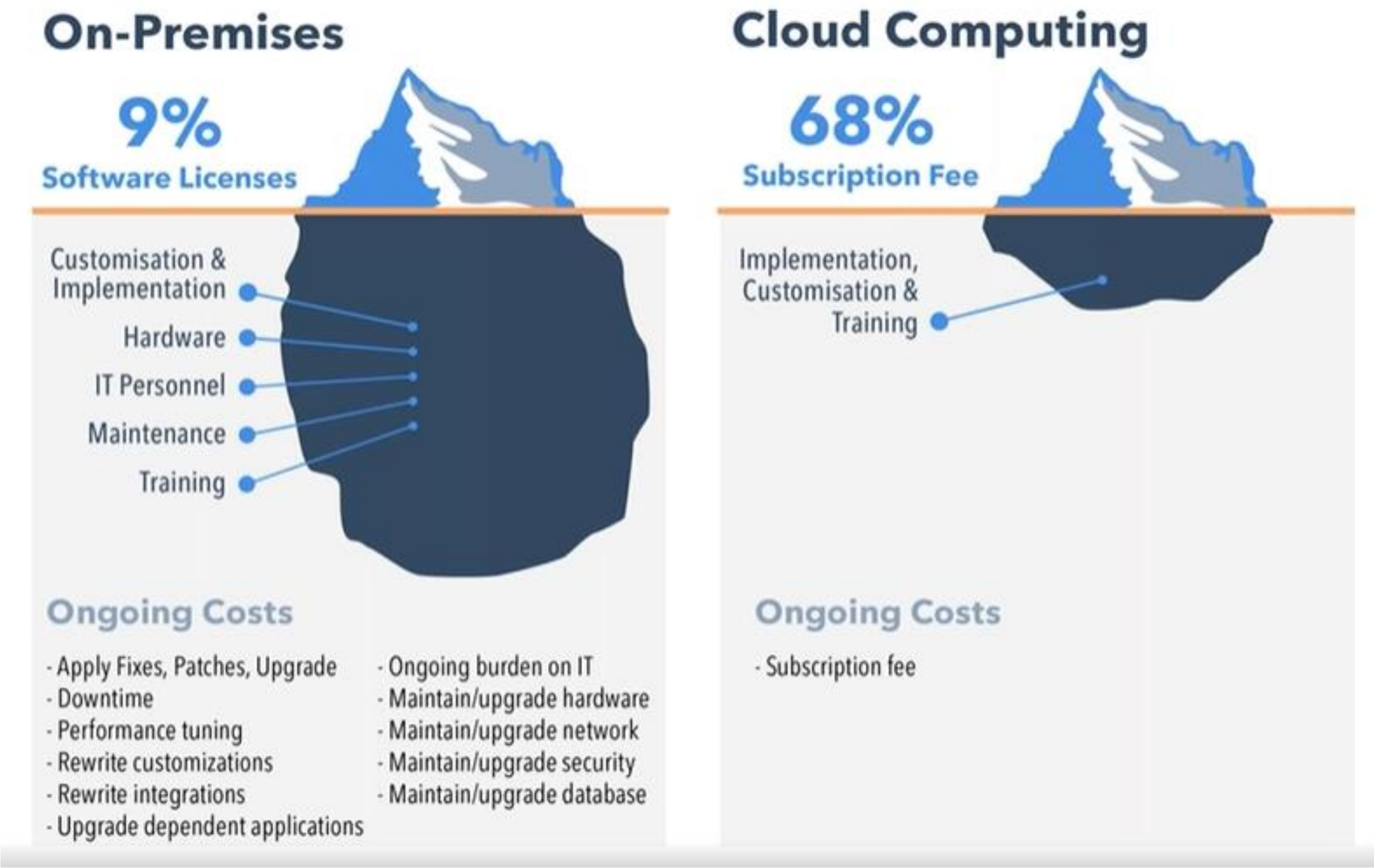
IoT Gateway Features :

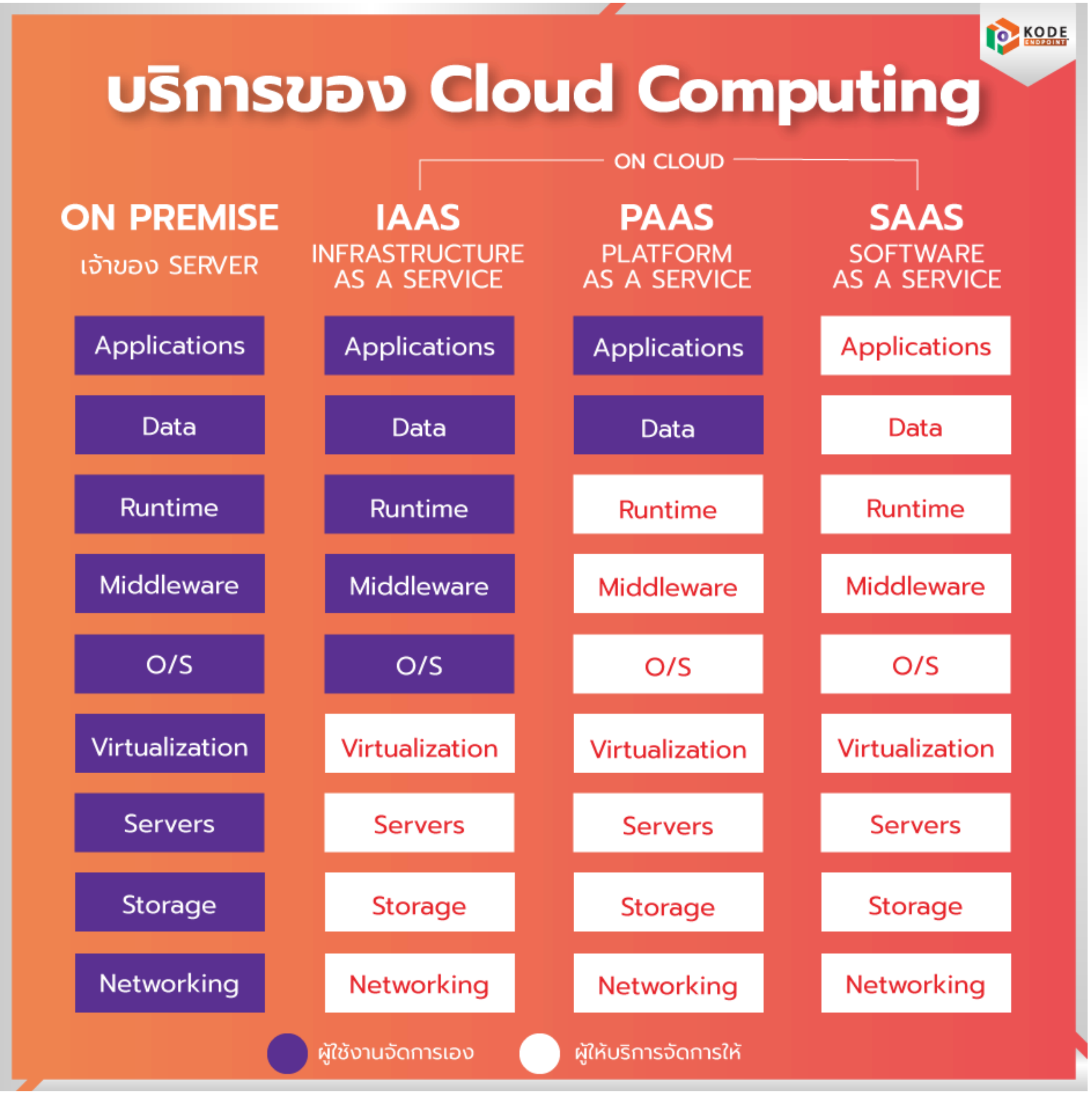
- ติดต่อสื่อสารเชื่อมโยงข้อมูล และการสื่อสารระหว่างอุปกรณ์ (M2M)
- ทำหน้าที่เป็นที่เก็บข้อมูลต่าง ๆ (data cache, buffer, and streaming device)
- การทำงานแบบออฟไลน์และการควบคุมอุปกรณ์แบบเรียลไทม์
- จัดการรวบรวมข้อมูลต่าง ๆ
- ประมวลผลข้อมูลเบื้องต้น (Pre-processes), จัดการข้อมูลที่ไม่จำเป็น (cleans, and filters data) ก่อนไปส่งยังขั้นต่อไป
- เป็นส่วนอัจฉริยะเพิ่มเติมสำหรับอุปกรณ์ไอโอทีบางอย่าง
- เพิ่มความปลอดภัย
- การจัดการกำหนดค่าอุปกรณ์และการเปลี่ยนแปลง

3. ระบบคลาวด์ (Cloud System)








KODE ENDPOINT
บริการของ Cloud Computing
ON CLOUD
ON PREMISE
เจ้าของ SERVER
IAAS
INFRASTRUCTURE AS A SERVICE
PAAS
PLATFORM AS A SERVICE
SAAS
SOFTWARE AS A SERVICE
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
ผู้ใช้งานจัดการเอง
ผู้ให้บริการจัดการให้

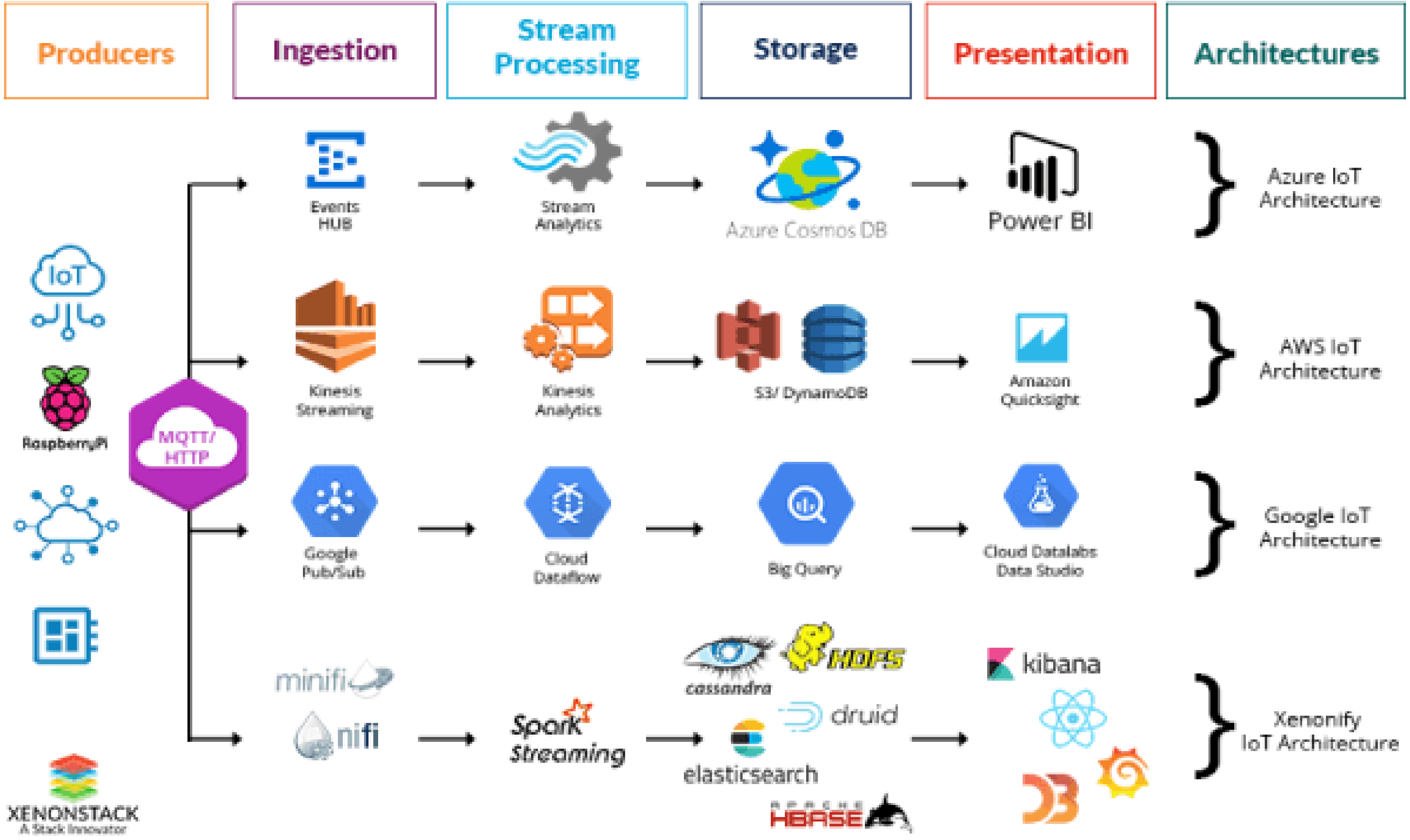
	Service areas	Amazon web services	Microsoft Azure	Google Cloud	Description
Global Cloud / Core	**IoT services**	• Machine learning • Kinesis • Data Pipeline	• Machine learning • Stream analytics • Power BI	• Machine learning • Stream analytics • Data warehouse	• Provides a preconfigured solution for monitoring, maintaining, and deploying common IoT scenario
	Cloud gateway	• AWS IoT	• Azure IoT Hub	• Google Cloud IoT	• A cloud gateway for managing bidirectional communication with billions of IoT devices, securely and at scale
Telco Network Edge	**Bulk data transfer**	• AWS Snowball Edge • AWS Snowmobile	• Azure Databox	• N/A	• PB to EB-scale data transport solution that uses secure data storage devices to transfer large amounts of data in and out of Cloud, at lower cost
Devices / On-premise aggregation & processing	**Edge computing for IoT**	• AWS Greengrass	• Azure IoT Edge	• Cloud IoT Edge • Edge TPU	• Managed service that deploys cloud intelligence directly on IoT devices to run in on-premises scenarios
	Real time data processing	• Kinesis Firehose • Kinesis Stream	• Event Hubs	• Cloud functions available for real-time processing	• Services that allow the mass ingestion of small data inputs, typically from devices and sensors, to process and route the data

Source: Microsoft Azure, Amazon Web Services, Delta Partners analysis

4. ส่วนการวิเคราะห์และเก็บข้อมูล

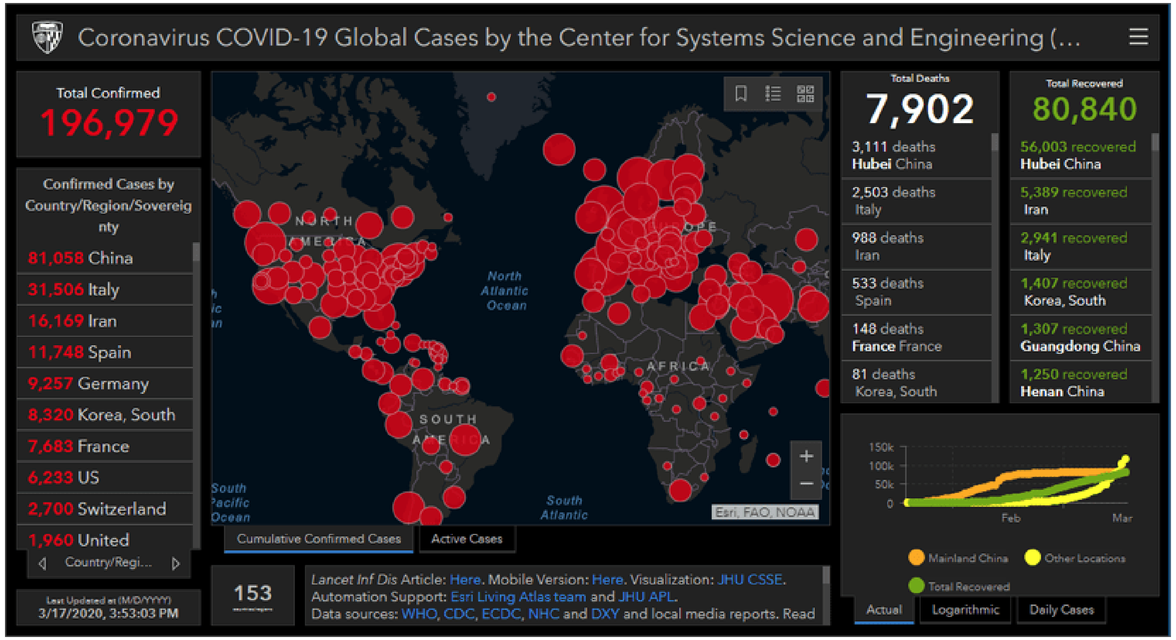
- เป็นส่วนที่นำข้อมูลที่ได้จากอุปกรณ์ส่งผ่านเกตเวย์มายังคลาวด์ เพื่อมาเก็บข้อมูลในระบบฐานข้อมูล (Database)
- ข้อมูลจะถูกเก็บไว้ เพื่อเรียกใช้ในการวิเคราะห์ (Analytics) หรือแสดงผลข้อมูล (Visualization)
- ระบบ AI (Artificial Intelligence) หรือ ML (Machine Learning)

4. ส่วนการวิเคราะห์และเก็บข้อมูล



5. ส่วนการปฏิสัมพันธ์กับผู้ใช้งาน (Human Interfacing)

IoT Dashboard คือ หน้าจอ หรือกระดานที่สรุปข้อมูลในรูปแบบต่าง ๆ เพื่อให้สามารถดูได้ง่าย ใช้เวลาตีความสั้น ๆ เพื่อให้เห็นการเปลี่ยนแปลงของข้อมูลตลอดเวลา



IoT Architecture & Model

IoT System &
Information
Engineering
KMITL

The Importance of IoT Architecture

- One of the main challenges to deal with the technological field to promote the deployment of IoT systems is to **define a reference architecture that supports current features and future extensions.**

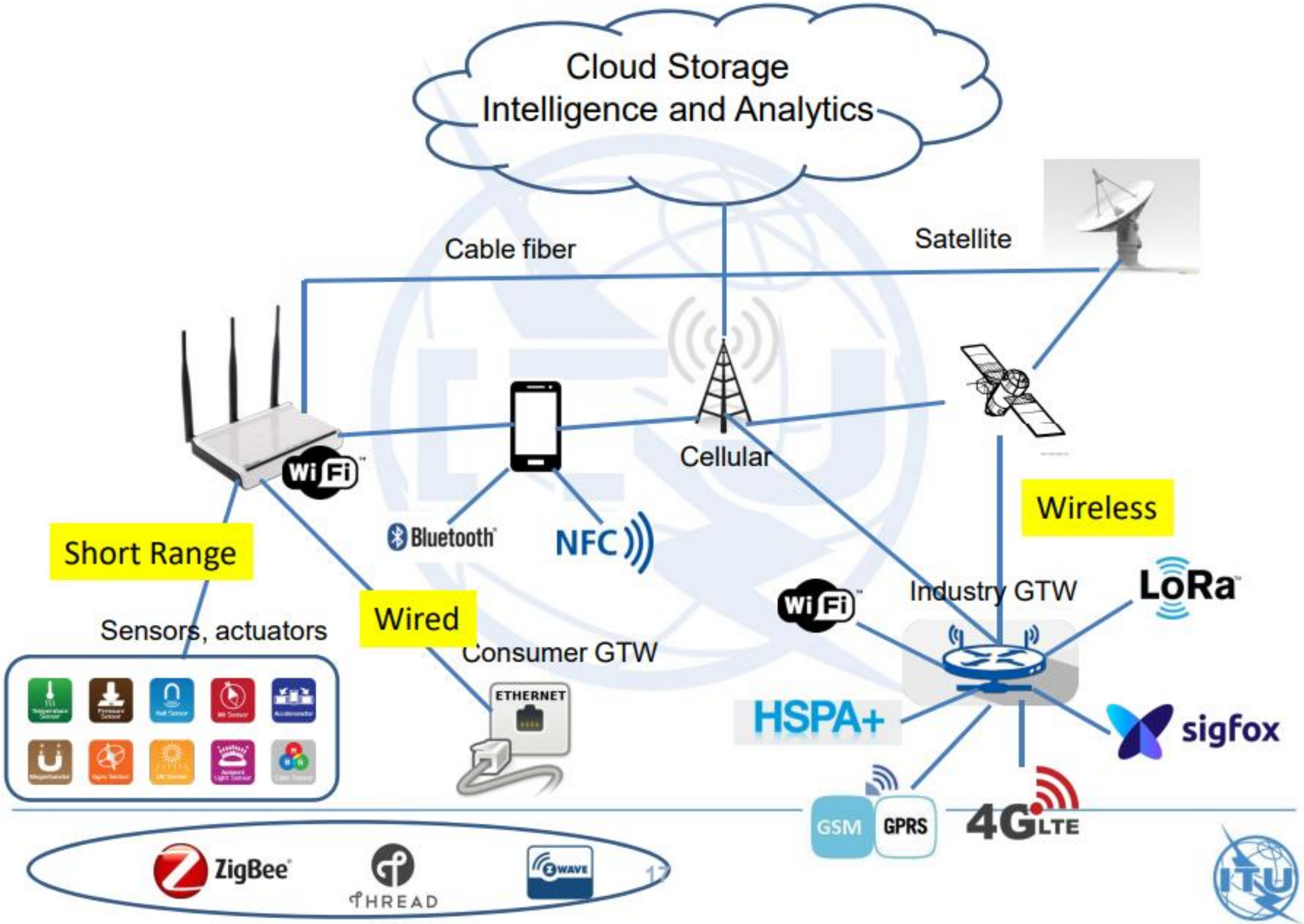
  - **Scalable**, in order to manage the increasing number of devices and services without degrading their performance;
  - **interoperable**, so that devices from different vendors can cooperate to achieve common goals;
  - **distributive**, to allow to create of a distributed environment in which, after being collected from different sources, data are processed by different entities in a distributed way;
  - **Able to operate with few resources**, since objects generally have little computing power;
  - **Secure** so as not to allow unauthorized access

Currently, there is no single reference architecture, and creating one is proving very complicated despite many standardization efforts.

The Importance of IoT Architecture

- Administrators use IoT architecture to manage and support IoT devices.
- IoT devices can be anything from an internet-connected light bulb to pressure safety sensors in a chemical plant.
- IoT architecture makes this all possible by ensuring data gets where it needs to and is processed correctly.
- Without proper IoT architecture, networks would become unreliable, defeating the entire purpose of investing in IoT in the first place.

IoT network general architecture

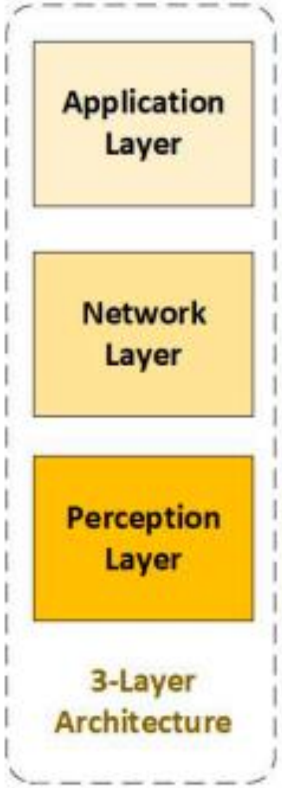


IoT reference models

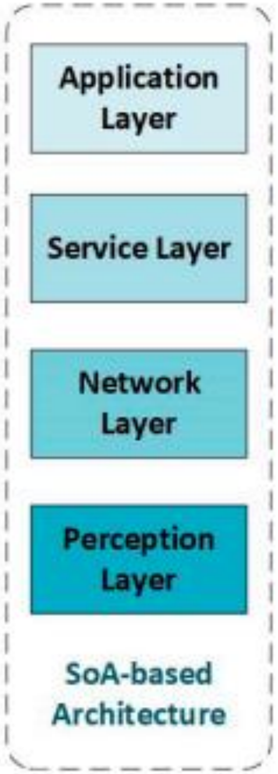
In general, the OSI and TCP/IP reference models can be applied to IoT networks, however, IoT does have specific requirements that have seen the development of specialised models.

IoT Architecture

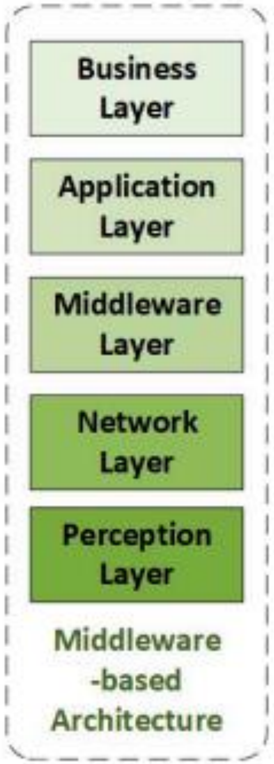
Three Layer



Four Layer

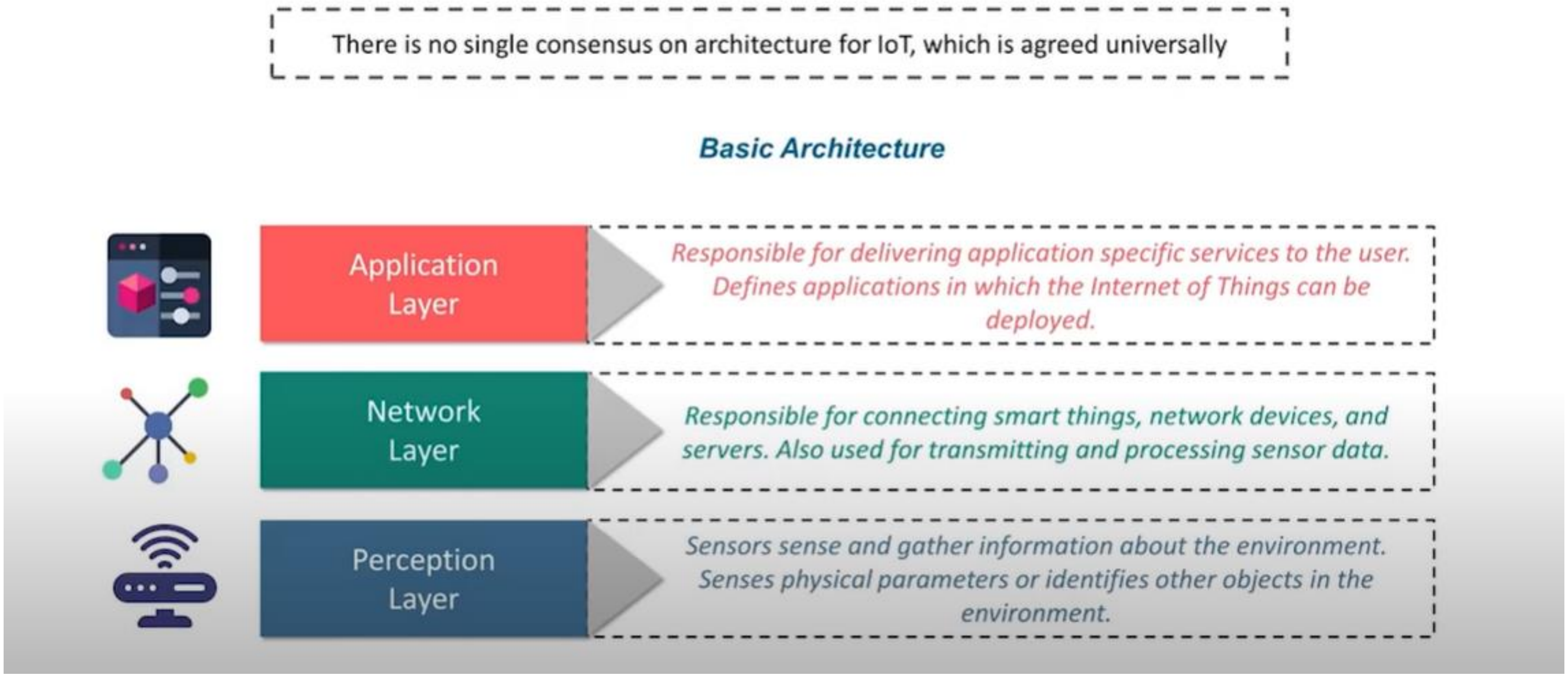


Five Layer



Seven Layer

IoT Architecture : Three Layer



oneM2M IoT Standardized Architecture

- In effort to standardize the rapidly growing field of machine to machine (M2M) communications, the European telecommunications Standards Institute (ETSI) created the M2M Technical committee in 2008.
- Goal : create a common architecture that would help accelerate the adoption of M2M applications and devices.

oneM2M IoT Standardized Architecture

- **Applications Layer**: connectivity between devices and their applications. This domain includes the application layer protocols and attempts to standardize northbound API definitions for interaction with Business Intelligence (BI) Systems.

- **Services Layer**: a horizontal framework across the vertical industry applications. horizontal modules include the physical network that the IoT applications run on, the underlying management protocols, and the hardware. Examples include backhaul communications via cellular, MPLS (Multiprotocol label switching) networks, VPNs and so on.

- **Network Layer**: This is the communication domain for the IoT devices and endpoints. It includes the devices themselves and the communication network that links them. Embodiments of this communication infrastructure includes wireless mesh technologies and wireless point to multipoint systems.

oneM2M IoT Standardized Architecture

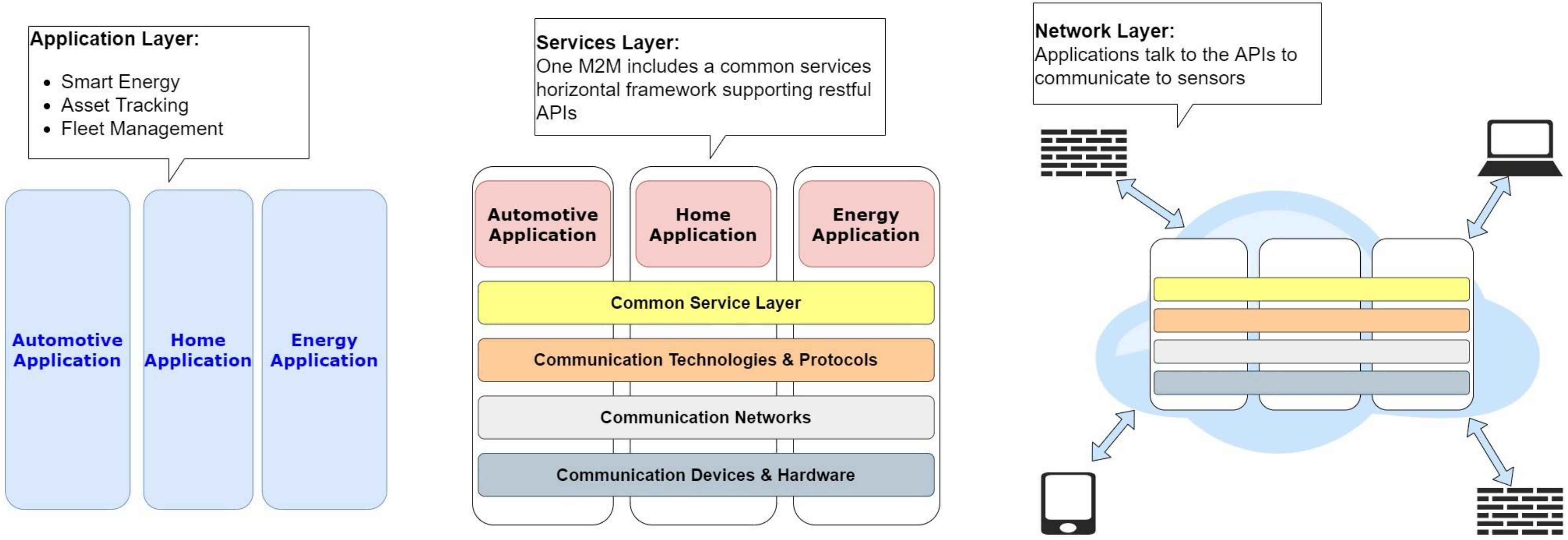
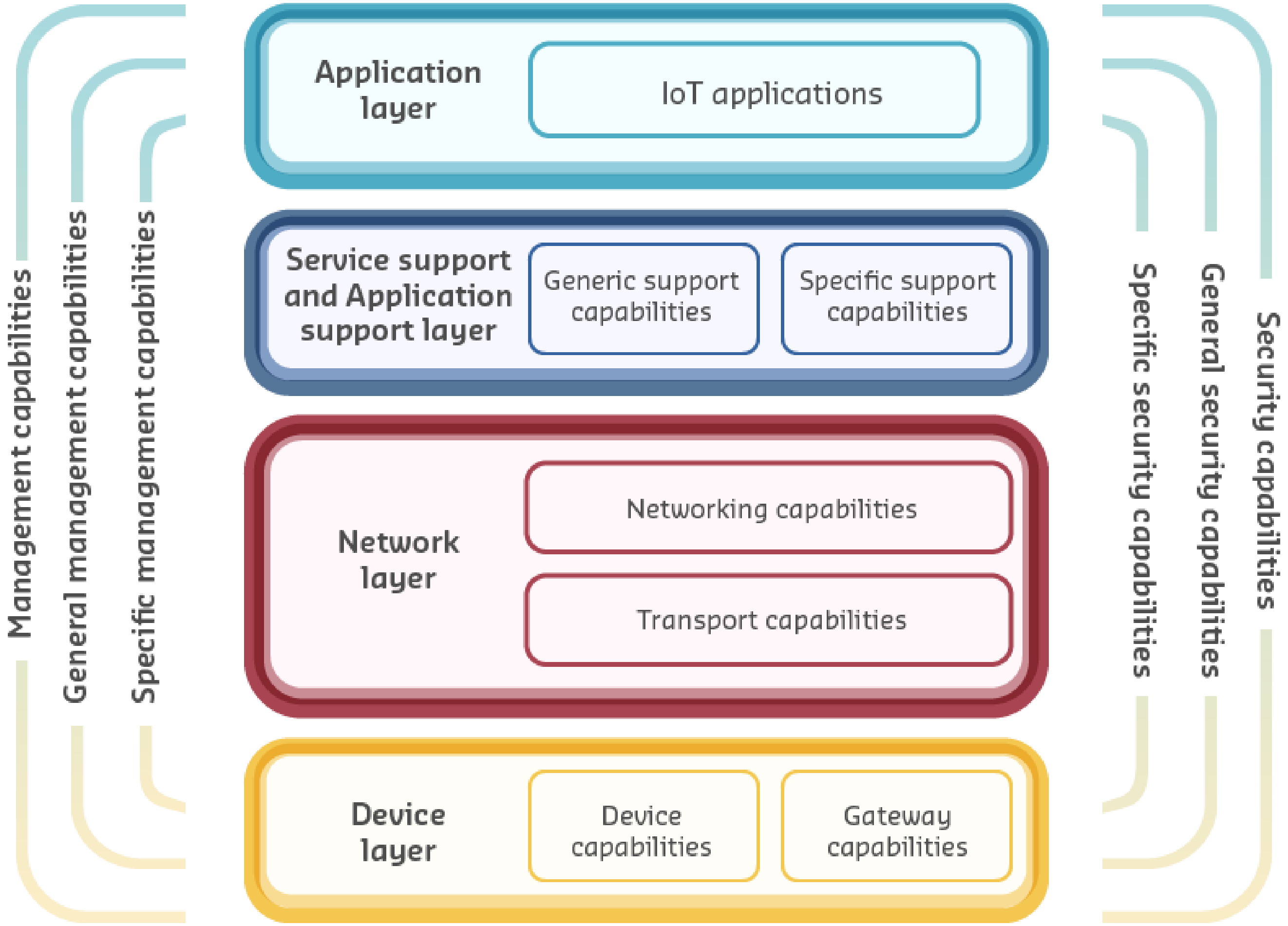


Fig: The Main Elements of the one M2M IoT Architecture

IoT reference models

- An **IoT reference model** is a *first step* towards standardising the concept and terminology surrounding the IoT.
- **ITU IoT reference model**
- The ITU (International Telecommunications Union) has developed a general model to represent the four functional layers to provide interconnectivity of **things using IoT applications**, with all layers supported by both specific and general management and security capabilities.



IoT Architecture : Four Layer

IoT Applications

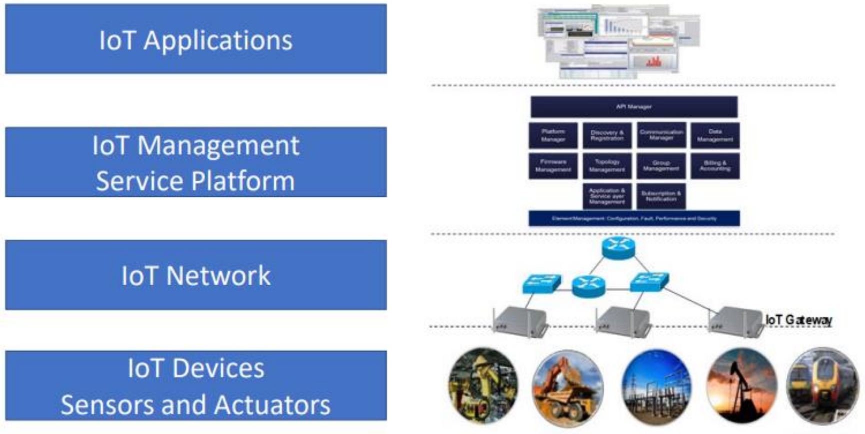
IoT Management
Service Platform

IoT Network

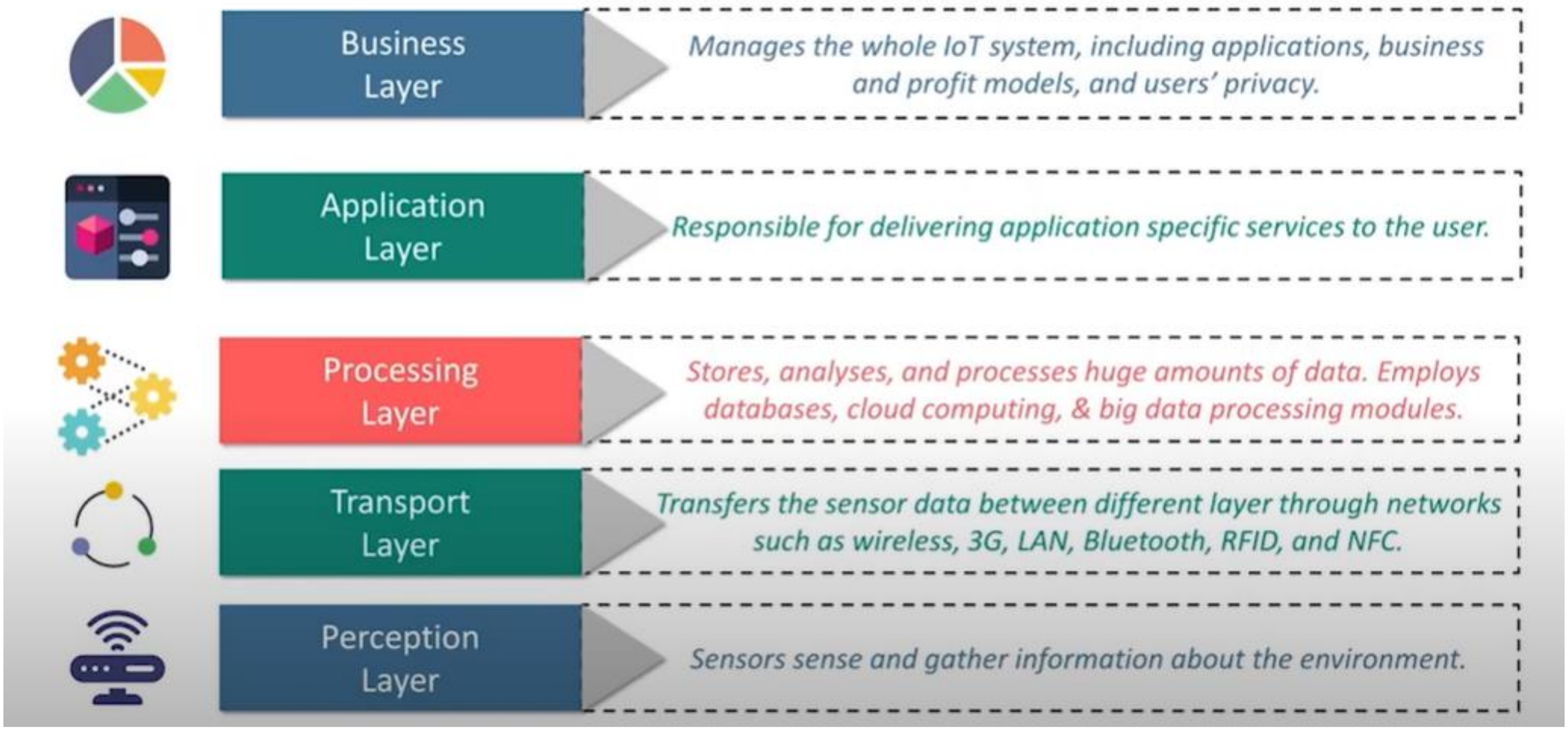
IoT Devices
Sensors and Actuators

IoT Architecture : Four Layer

- **Level 1:** IoT Device Level includes all IoT sensors and actuators.
- **Level 2:** IoT Network Level includes all IoT network components including IoT Gateways, Routers, Switches, etc.
- **Level 3:** IoT Application Services Platform Level includes the key management software functions to enables the overall management of IoT devices and network. It also includes main functions connecting the device and network levels with the application layer.
- **Level 4:** IoT Application Level includes all applications operating in the IoT network.

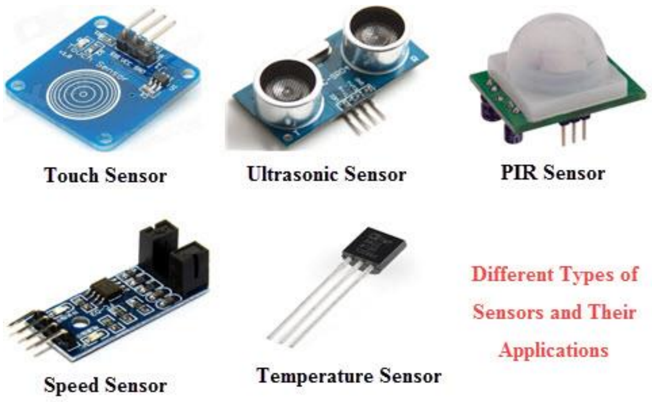


IoT Architecture : Five Layer



Perception Layer

- This is the first layer of IoT architecture. Number of sensors and actuators are used to gather useful information like temperature, moisture content, intruder detection, sounds, etc.
- The main function of this layer is to get information from surroundings and to pass data to another layer so that some actions can be done based on that information.
- **Sensors** such as probes, gauges, meters, and others. They collect physical parameters like temperature or humidity, turn them into electrical signals, and send them to the IoT system. IoT sensors are typically small and consume little power.
- **Actuators**, translating electrical signals from the IoT system into physical actions. Actuators are used in motor controllers, lasers, and robotic arms.
- **Machines** and devices connected to sensors and actuators or having them as integral parts.



Network/Transport Layer

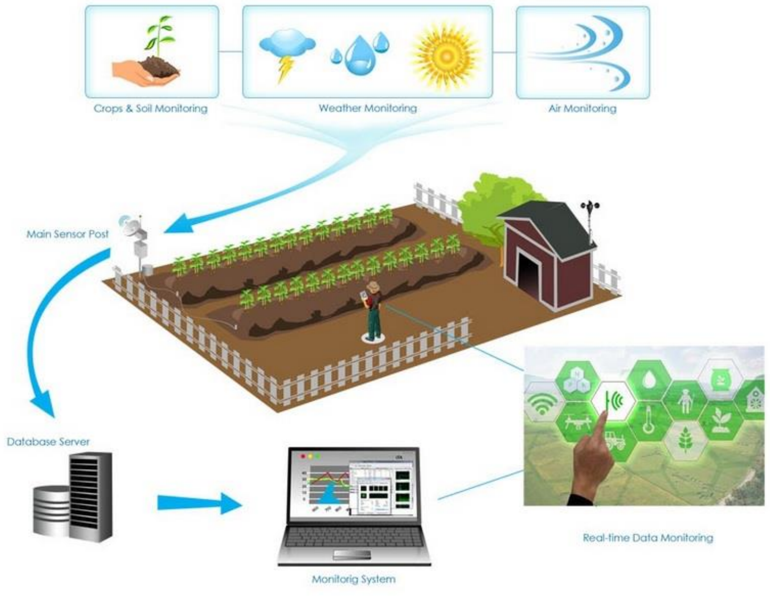
- It gets data from perception layer and passes data to middleware layer using networking technologies like 3G, 4G, Bluetooth, WiFI, infrared, Low-Power wide area network etc.
- This is also called communication layer because it is responsible for communication between perception and middleware layer.
- All the transfer of data done securely keeping the obtained data confidential.

Middleware Layer/Processing Layer

- Middleware Layer has some advanced features like **storage, computation, processing**, action taking capabilities.
- It stores all data-set and based on the device address and name it gives appropriate data to that device.
- It can also take decisions based on calculations done on data-set obtained from sensors.
- Modern IoT architectures leverage machine learning and artificial intelligence that create value by analyzing this data.

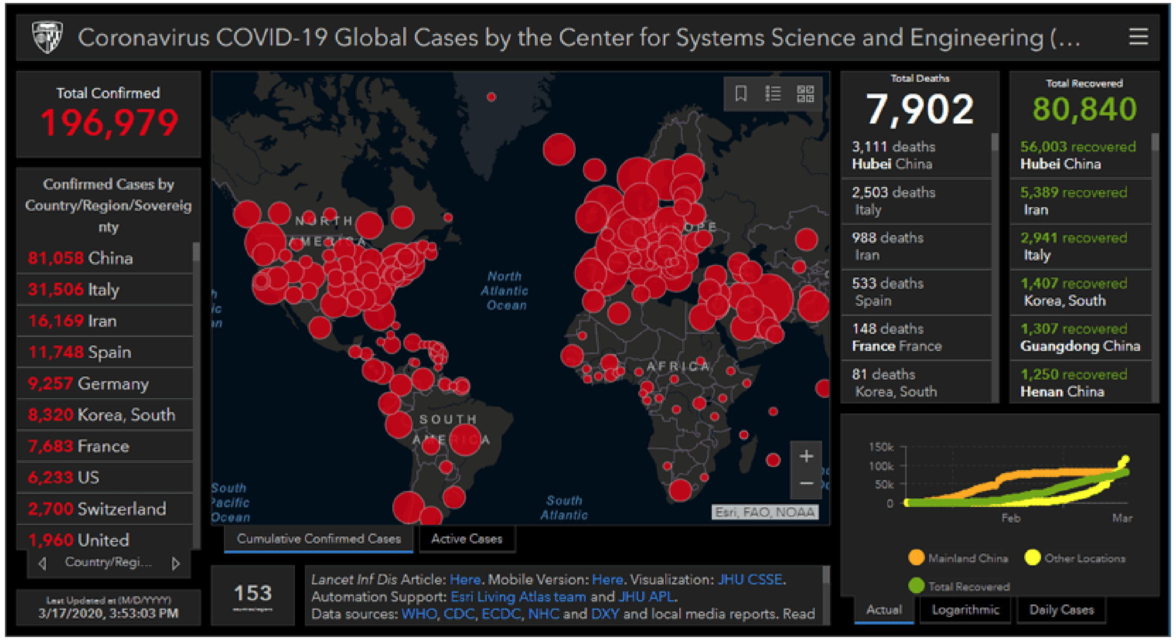
Application Layer

- The application layer manages all application process based on information obtained from middleware layer.
- This application involves sending emails, activating alarm, security system, turn on or off a device, smartwatch, smart agriculture, etc.



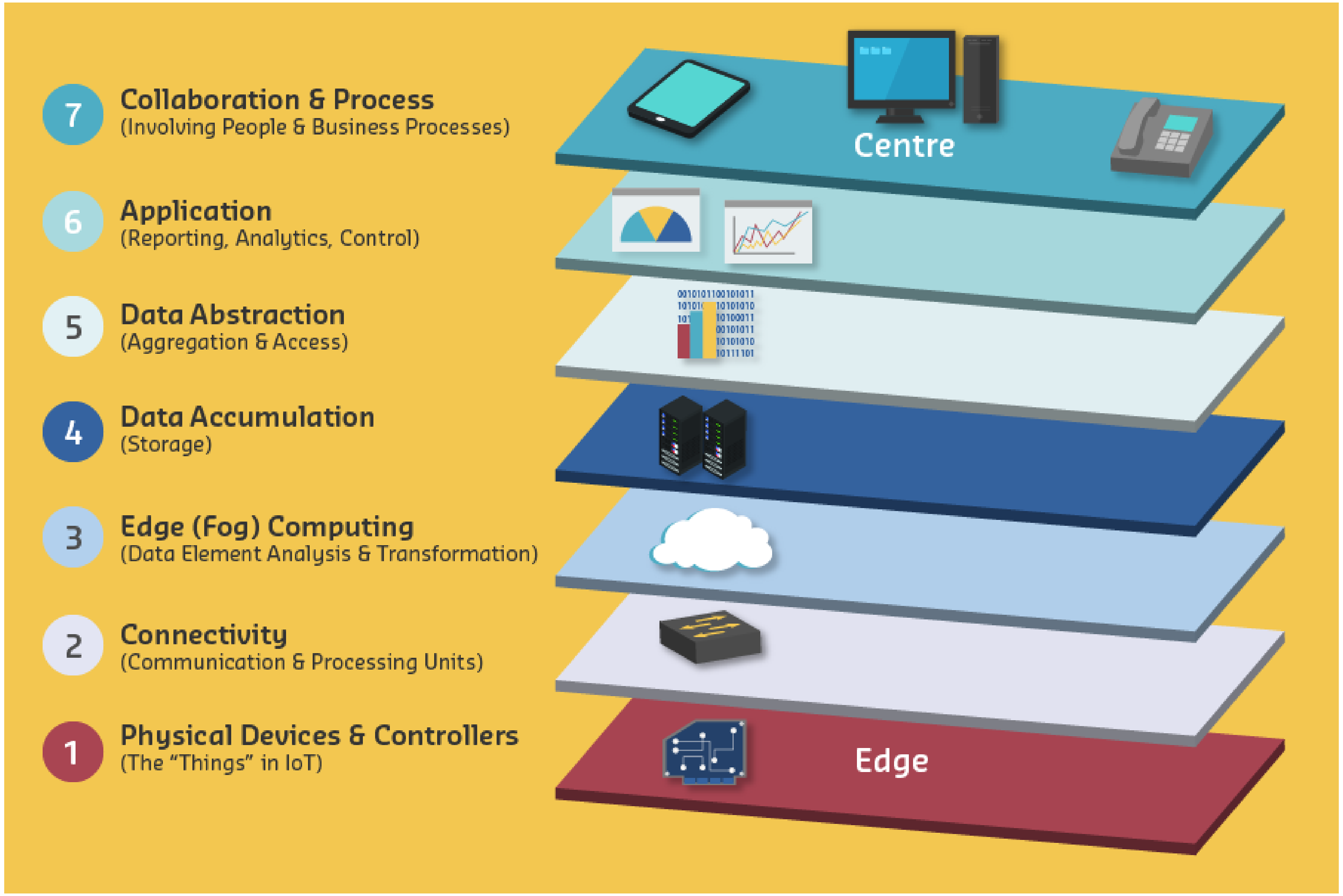
Business Layer

- The success of any device does not depend only on technologies used in it but also how it is being delivered to its consumers.
- Business layer does these tasks for the device. It involves making flowcharts, graphs, analysis of results, and how device can be improved, etc.
- For example, it takes raw data and forms insights in graphs and charts. This layer is related to decision-making because it interacts with stakeholders.



IoT reference models : The IoT World Forum (IoTWF)

- More applied approach can be seen in models such as the one developed by Cisco Systems, Inc.
- It relates the layers to features and processes found in IoT networks.
- This example IoT reference model sets out the functionalities required, and challenges that must be addressed, from physical devices and controllers at Level 1, to the collaboration and processes at Level 7, so the value of the IoT can be effectively and securely realised.



IoT reference model

Levels

7 Collaboration & Processes
(Involving People & Business Processes)

6 Application
(Reporting, Analytics, Control)

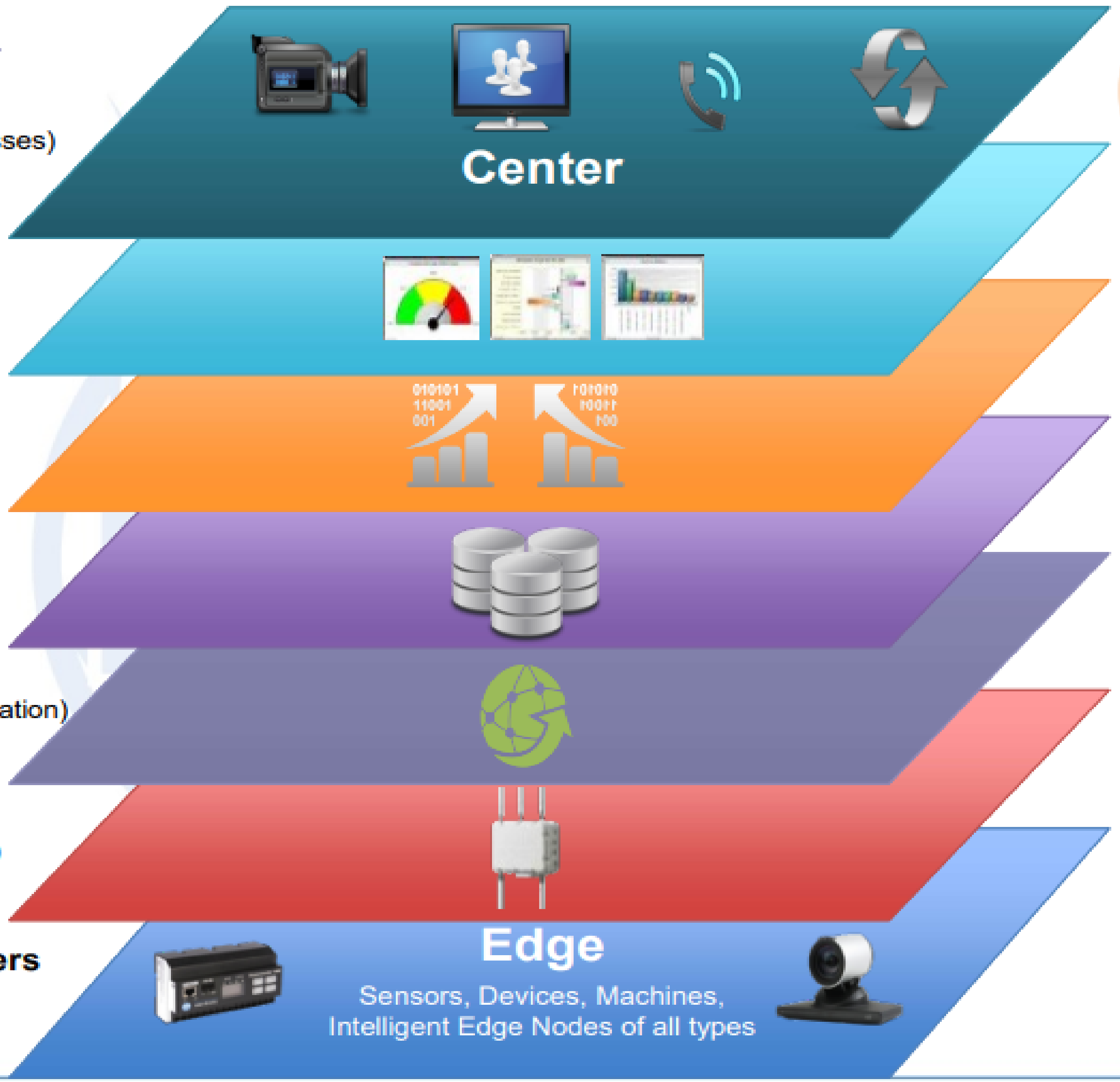
5 Data Abstraction
(Aggregation & Access)

4 Data Accumulation
(Storage)

3 Edge Computing
(Data Element Analysis & Transformation)

2 Connectivity
(Communication & Processing Units)

1 Physical Devices & Controllers
(The "Things" in IoT)



The model is based on **"Integrated Security & Management"**

Data at Rest

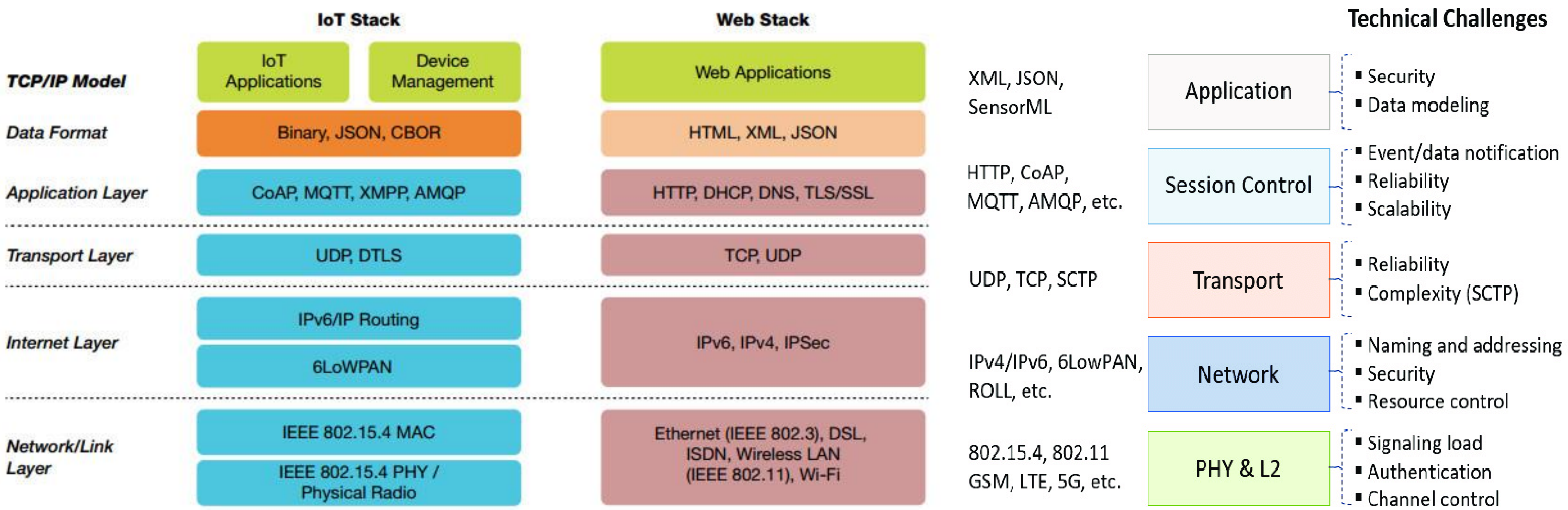
IT

OT

Data in Motion

The model is based on **"Information Flow"**

IoT Stack vs Web Stack



Other used architecture

- The high-level architecture just described provides a simple and general overview of an IoT project's structure and has been described in more detail to provide an overview of the leading technologies and protocols used in IoT.
- Depending on the application, however, it could be necessary to add additional layers or adapt the architecture to the specific application to realize.
- In particular, there are two different architectures: the architectures belonging to
- the first class are obtained by adapting to the context of IoT existing architectures; the architectures belonging to the second class are built from scratch.

Architectures Adapted to the IoT Context	**Architectures from Scratch**
IETF Protocol stack	Cloud-based architectures
Server-based architecture	Edge computing-based architectures
	Social Internet of Things (SIoT) architectures

Internet Engineering Task Force (IETF) Protocol Stack

- Due to the scarcity of computational resources and the heterogeneity of devices and traffic, new protocols have been introduced that have replaced or flanked the TCP/IP stack protocols.
- In particular, instead of the application-level protocols of the TCP/IP stack, the Constrained Application Protocol (CoAP) is used, which represents a lightweight version of the Hypertext Transfer Protocol (HTTP), suitable to working with devices and sensors with limited resources. CoAP also uses the User Datagram Protocol (UDP) at the transport level,
- Finally, a layer is added (adaptation layer) in which theIPv6 packet headers, using the IPv6 over Low Power Wireless Personal Area Network (6LoWPAN) protocol, are encapsulated and compressed so that devices can manage them with little computing power.

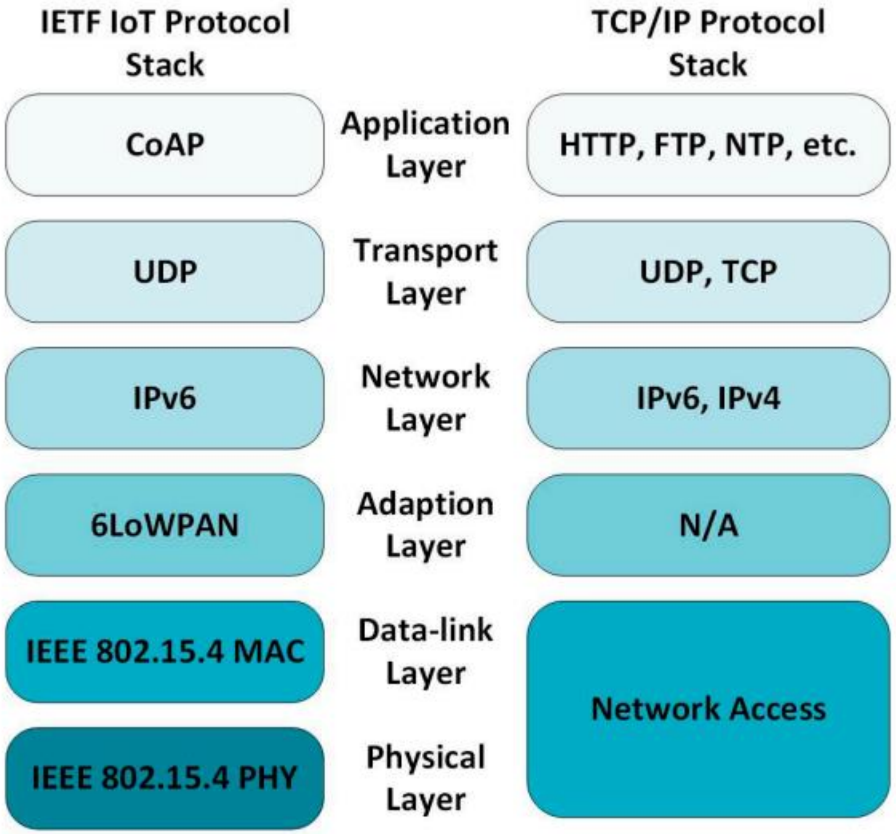


Figure 2. IETF protocol stack for IoT (**left**) and TCP/IP stack (**right**).

**All of protocols will be discussed in detail in the future.

Server Based Architecture

- Various devices are connected to a gateway server that manages the devices' connection to the Internet,
- The main difference from the architecture described above is that the devices are not directly accessible and only look at the Internet through the gateway.
- In this way, the constraints given by the scarcity of computational resources and heterogeneity must be managed only at the lowest level, where the connection between the devices and the gateway itself is implemented.
- At the higher level, the standard protocols of the TCP/IP stack can be used

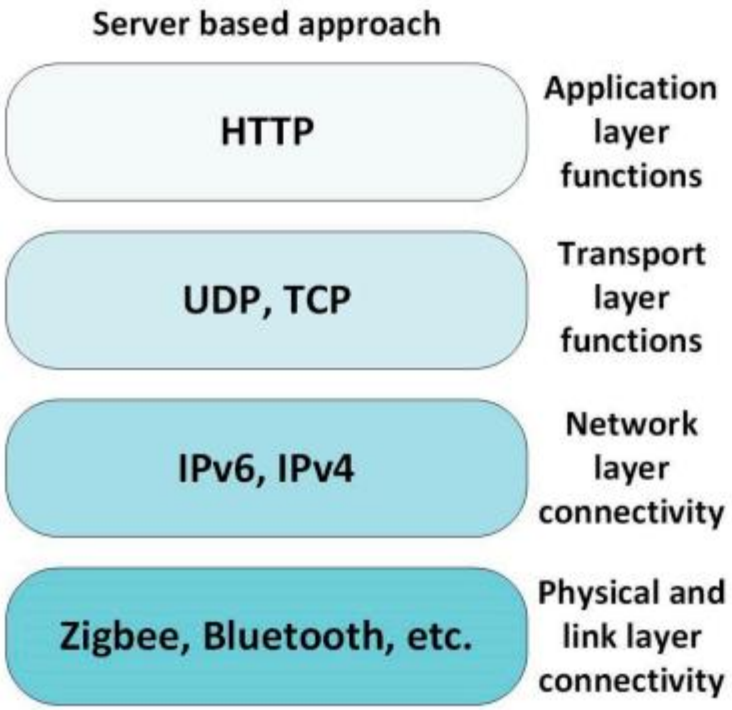


Figure 3. Server-based architecture.

Cloud-Based Architecture

- The large amount of data generated by objects is stored, processed, and presented to the user through services made available from the cloud.
- There are many architectures of this type realized by various companies such as AWS by Amazon, Azure by Microsoft, Google, etc.
- AWS consists of four main elements: The Device Gateway, the Registry, the Rules Engine, and the Device Shadow. The Device Gateway acts as an intermediary between the connected devices and the cloud services through the Message Queue Telemetry Transport (MQTT) protocol.
- The Rules Engine processes incoming messages and distributes them to other devices or cloud services. The Registry is responsible for assigning a unique ID to each connected device, regardless of device type, vendor, or connection mode. Additionally, it is responsible for storing metadata about connected devices in order to track them.
- Device Shadows are virtual representations of physical objects. These representations are persistent and stored in the cloud to be accessed at any time (even when the device is offline) by cloud services or other devices.

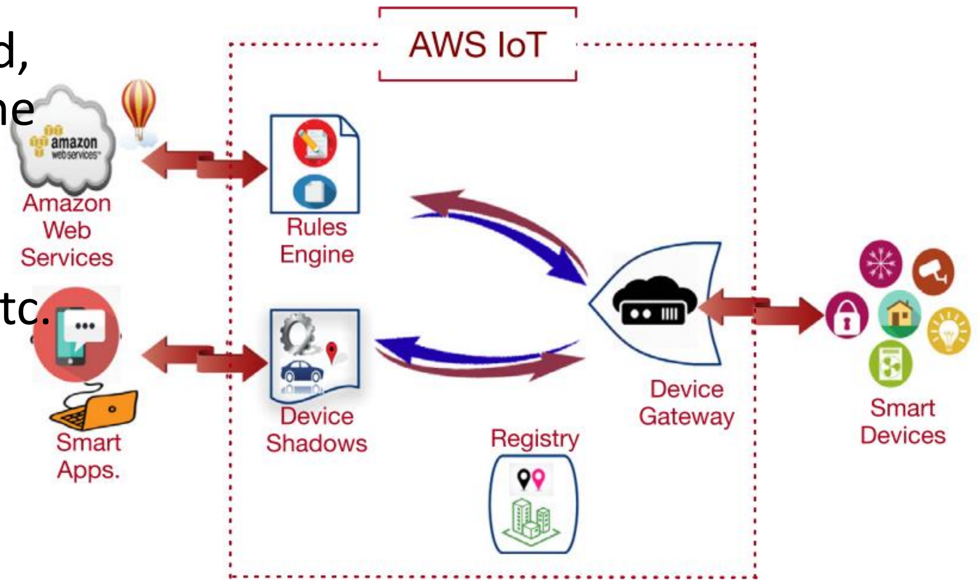
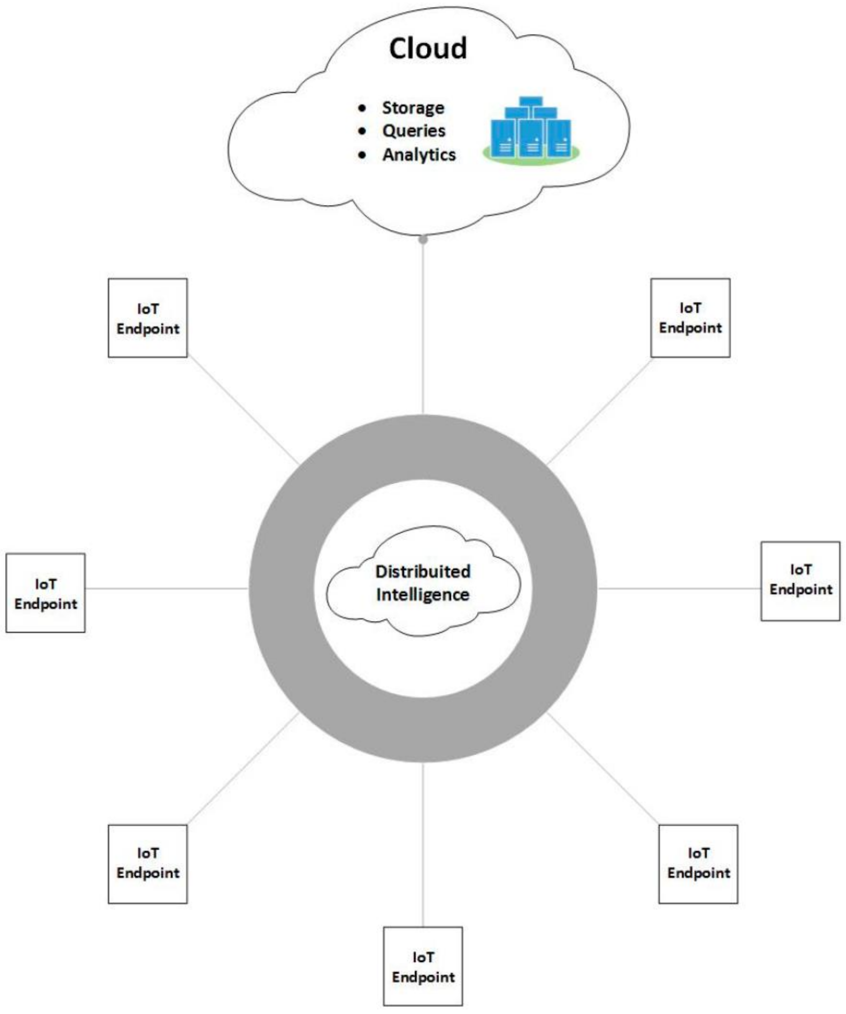


Fig. 2. AWS IoT architecture.

Edge Computing-Based Architecture

- Edge Computing is a horizontal architecture that provides distributed computing, data storage, and control capabilities. It interferes between cloud structures and end users to make available files and resources that are usually accessible only through a network connection.
- Edge Computing can provide faster response and greater quality of service.
- Edge Computing may have more benefits:
  - Minimizes latency;
  - many actions are taken very close to where the action is;
  - Allows for bandwidth saving, avoiding over dimensioning the band to the Cloud;
  - Solves some security issues, because many decisions are taken in a subnet and are not exposed to the risks arising from the external Internet.



References

Lombardi, M.; Pascale, F.; Santaniello, D. Internet of Things: A General Overview between Architectures, Protocols and Applications. Information 2021, 12, 87. https://doi.org/10.3390/info12020087

อินเตอร์เน็ตในทุกสรรพสิ่ง

Internet of things