Introduction to IoT

Kevin‘s Explanation

- Kevin‘s Explanation:
- IoT involves the addition of senses to computers.
- In the 20th centaury, computers were brains without senses.
- In the 21st centaury, computers are sensing things for themselves.

Kevin‘s Explanation

Industry 4.0: IoT Integration *(Today)*
Sensors with a new level of interconnectivity are integrated

Industry 3.0: Electronics and Control *(Early 1970's)*
Production is automated further by electronics and IT

Industry 2.0: Mass Production *(Early 20th Century)*
Division of labor and electricity lead to mass production facilities

Industry 1.0: Mechanical Assistance *(Late 18th Century)*
Basic machines powered by water and steam are part of production facilities

The Four Industrial Revolutions

- What is Internet?
  - The Internet is the global system of interconnected computer networks that use the Internet protocol suite (TCP/IP) to link devices worldwide.
  - It is a network of networks that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies.
  - The Internet carries an extensive range of information resources and services, such as the inter-linked hypertext documents and applications of the World Wide Web (WWW), electronic mail, telephony, and peer-to-peer networks for file sharing.

- What is things?
  - Object like Sensor, Computer, Mobile Phone

Evolution of Internet

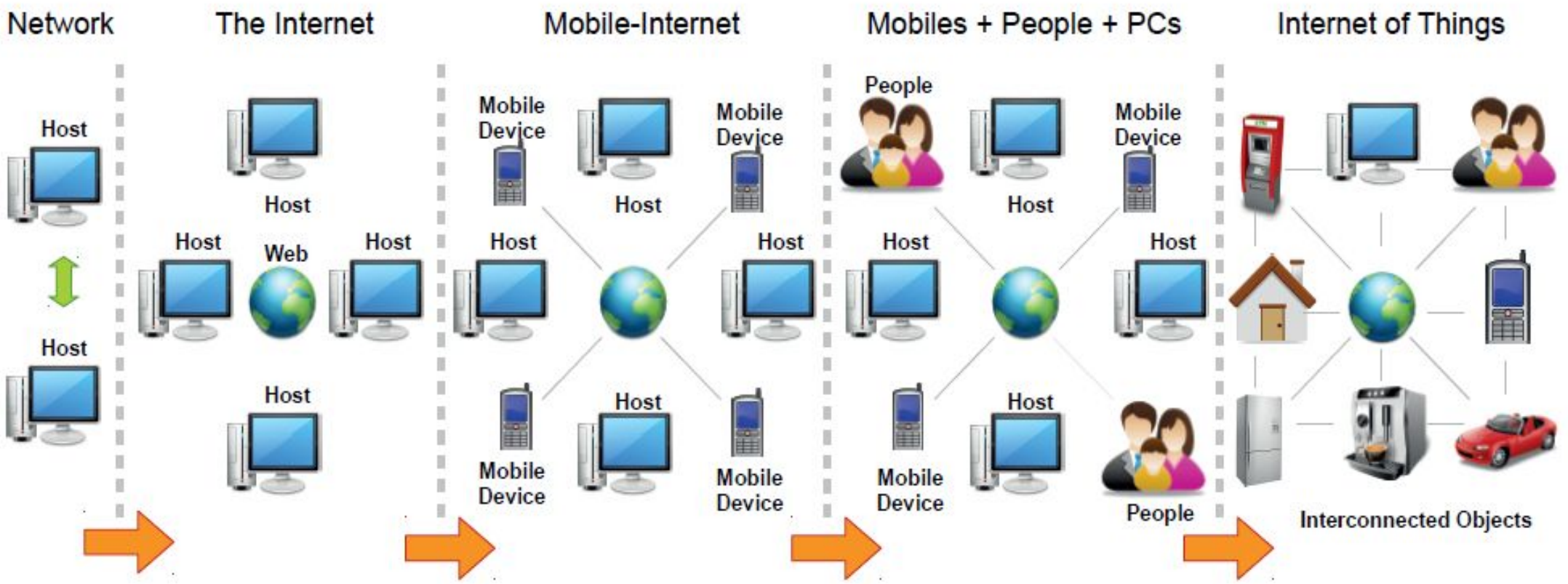


Fig. 1. Evolution of the Internet in five phases. The evolution of Internet begins with connecting two computers together and then moved towards creating World Wide Web by connecting large number of computers together. The mobile-Internet emerged by connecting mobile devices to the Internet. Then, peoples' identities joined the Internet via social networks. Finally, it is moving towards Internet of Things by connecting every day objects to the Internet.

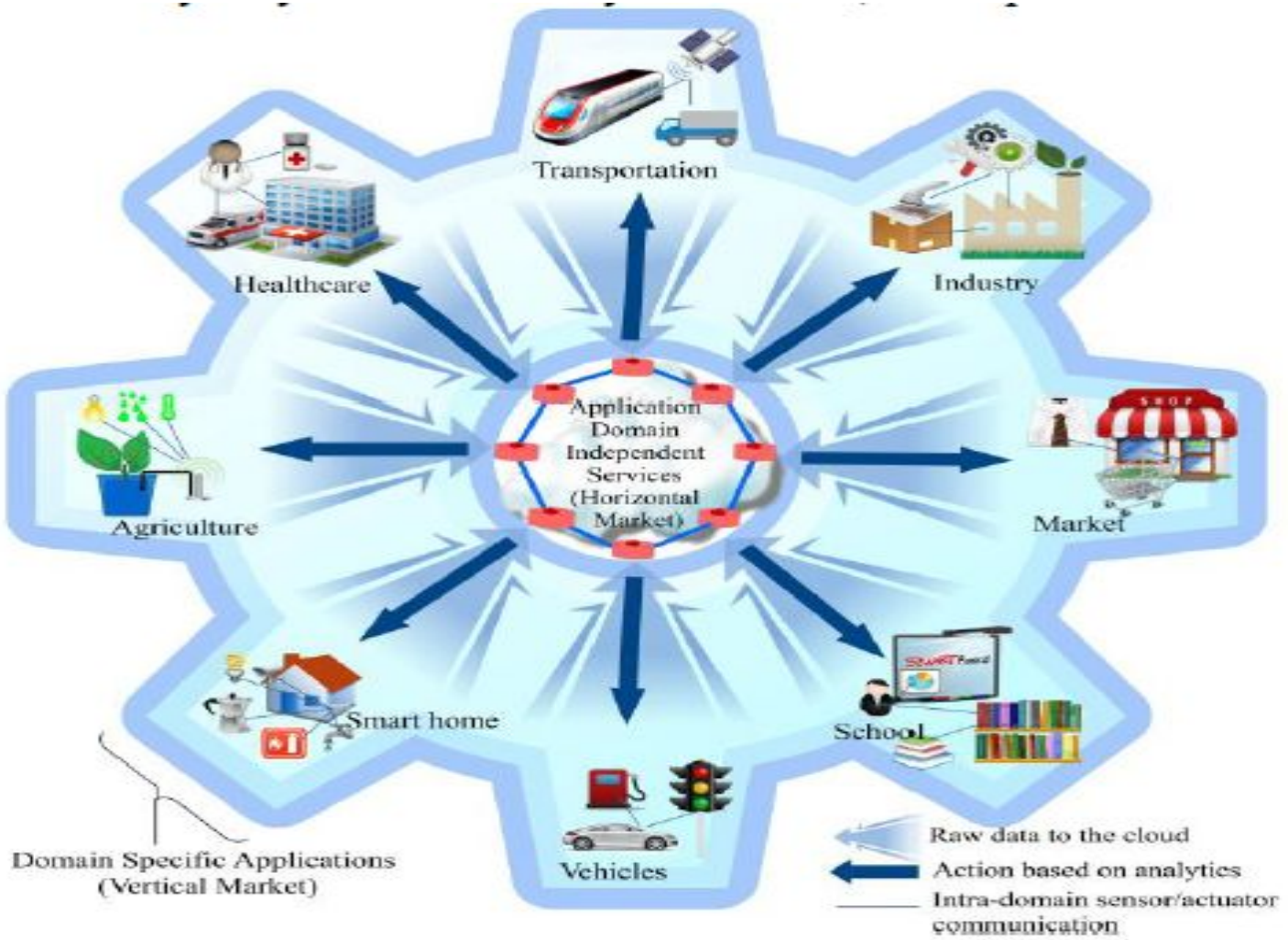


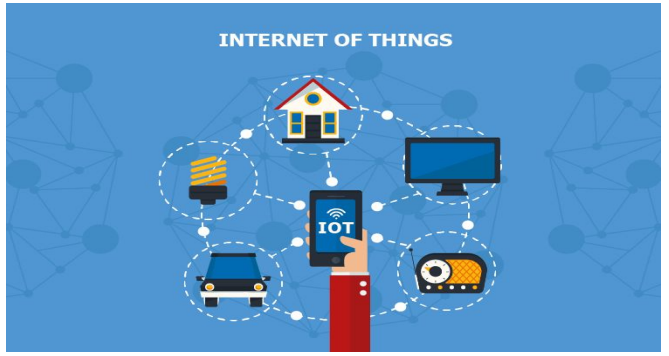
Fig. 1. The overall picture of IoT emphasizing the vertical markets and the horizontal integration between them.

Outline

- IoT definition
- Characteristics of IoT
- Physical Design of IoT
- Logical Design of IoT
- IoT Enabling Technologies Protocols
- IoT Levels & Deployment Templates

What is Internet of Things

- *The Internet of Things, or "IoT" for short, is about extending the power of the internet beyond computers and smartphones to a whole range of other things, processes and environments.*



- The internet of things, or **IoT**, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction

Definition of IoT

A dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual "things" have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network, often communicate data associated with users and their environments.

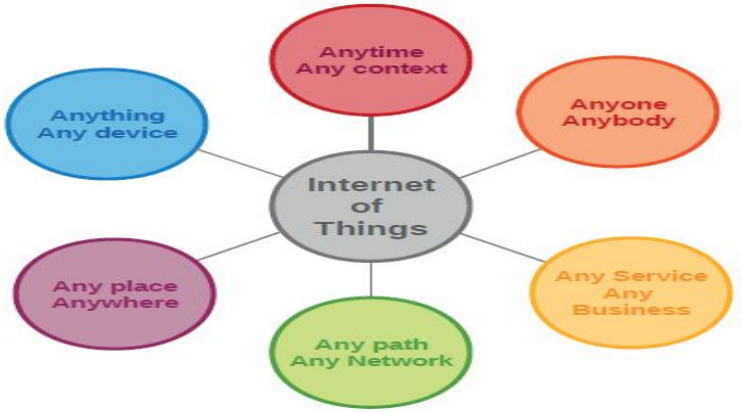


Fig. 2. Definition of the Internet of Things: The Internet of Things allows people and things to be connected anytime, anyplace, with anything and anyone, ideally using any path/network and any service [21].

About Object/Things in IoT

The Internet of things (IoT) is the inter-networking of physical devices, vehicles (also referred to as "connected devices" and "smart devices"), buildings, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to collect and exchange data.

Sensors and Actuators

- **Definition:** *Sensors are active devices that measure some variable of the natural or man-made environment (e.g., a building, an assembly line, an industrial assemblage supporting a process)*.

- **Definition:** *An actuator is a mechanized device of various sizes (from ultra-small to very large) that accomplishes a specified physical action, for example, controlling a mechanism or system, opening or closing a valve, starting some kind or rotary or linear motion, or initiating physical locomotion. An actuator is the mechanism by which an entity acts upon an environment.*

IoT Elements

Fig. 4. The IoT elements.

Characteristics of IoT

- Dynamic & Self-Adapting
- Self-Configuring
- Interoperable Communication Protocols
- Unique Identity
- Integrated into Information Network

Applications of IoT

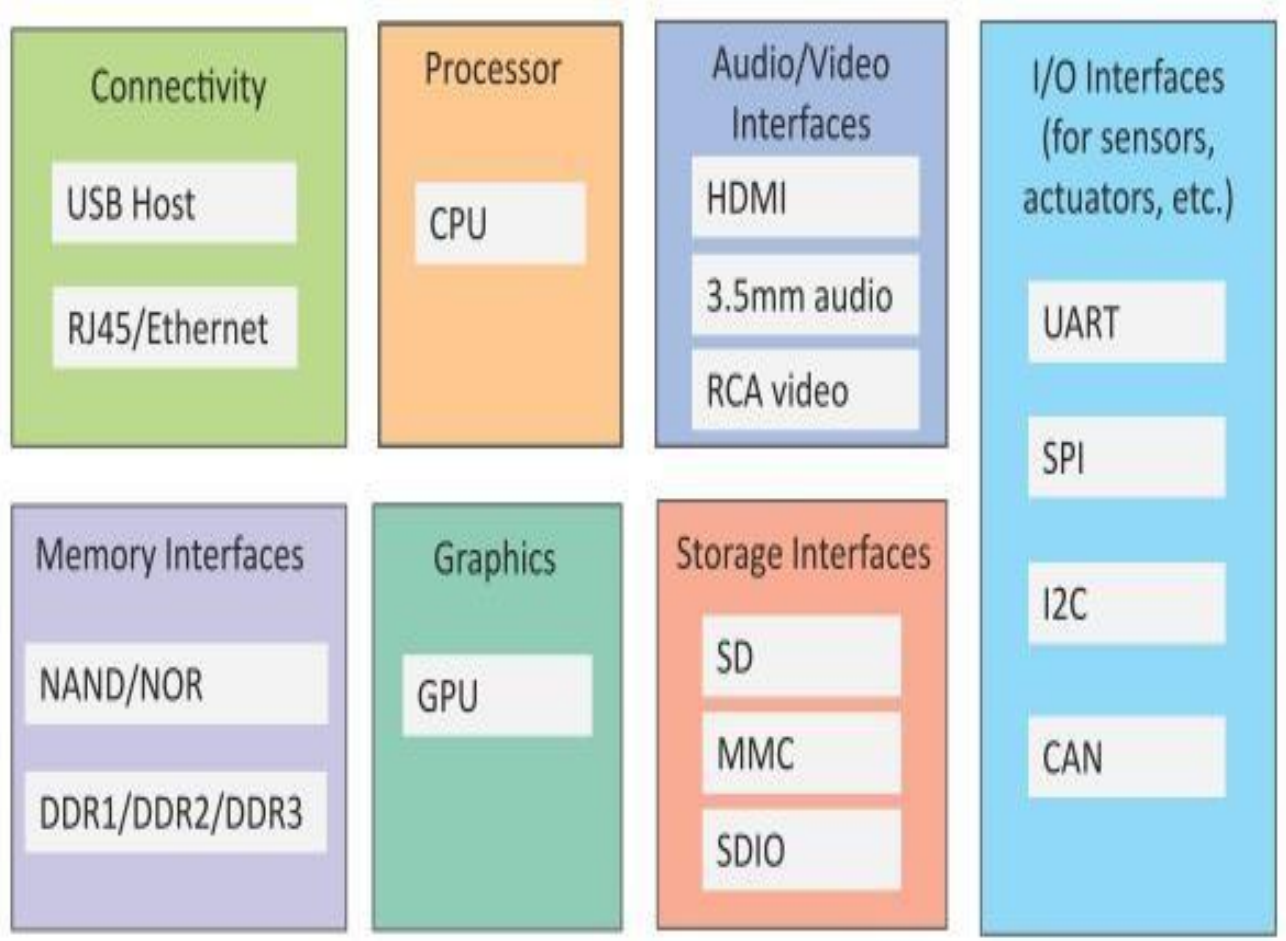
- Home
- Cities
- Environment
- Energy
- Retail
- Logistics
- Agriculture

Physical Design of IoT

- The "Things" in IoT usually refers to IoT devices which have unique identities and can perform remote sensing, actuating and monitoring capabilities.
- IoT devices can:
  - Exchange data with other connected devices and applications (directly or indirectly), or
  - Collect data from other devices and process the data locally or
  - Send the data to centralized servers or cloud-based application back-ends for processing the data,or
  - Perform some tasks locally and other tasks within the IoT infrastructure, based on temporal and space constraints

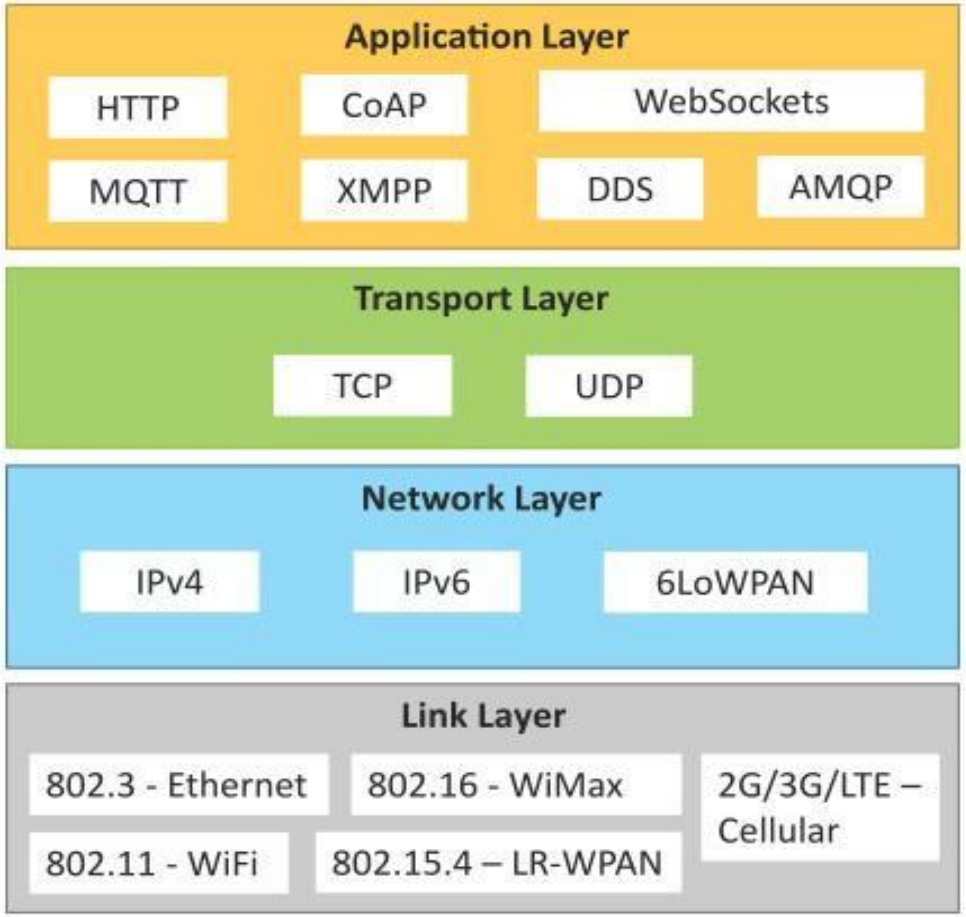
Generic block diagram of an IoT Device

- An IoT device may consist of several interfaces for connections to other devices, both wired and wireless.
  - I/O interfaces for sensors
  - Interfaces for Internet connectivity
  - Memory and storage interfaces
  - Processor & Graphics interfaces
  - Audio/video interfaces.



IoT Protocols

- Link Layer
  - 802.3 – Ethernet
  - 802.11 – WiFi
  - 802.16 – WiMax
  - 802.15.4 – LR-WPAN
  - 2G/3G/4G
- Network/Internet Layer
  - IPv4
  - IPv6
  - 6LoWPAN
- Transport Layer
  - TCP
  - UDP
- Application Layer
  - HTTP
  - CoAP
  - WebSocket
  - MQTT
  - XMPP
  - DDS



Logical Design of IoT

- Logical design of an IoT system refers to an abstract representation of the entities and processes without going into the low-level specifics of the implementation.

- An IoT system comprises of a number of functional blocks that provide the system the capabilities for identification, sensing, actuation, communication, and management.

Logical Design of IoT

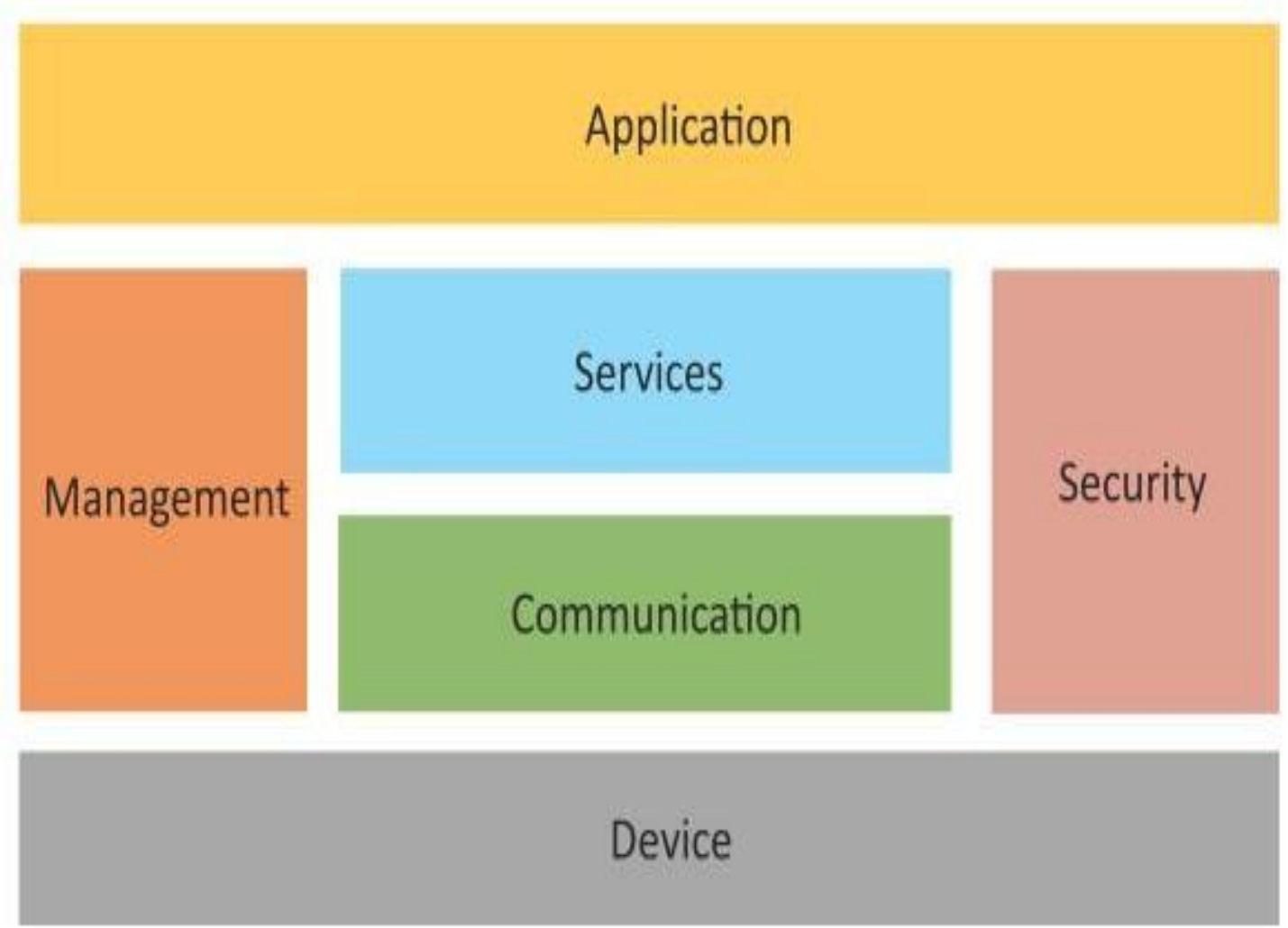
- Terms used to understanding of Logical

1. IoT Functional Blocks
2. IoT Communication Model
3. IoT Communication API’s

Logical Design of IoT: 1. IoT Functional Blocks

- Blocks:

1. Device
2. Communication
3. Services
4. Management
5. Security
6. Application



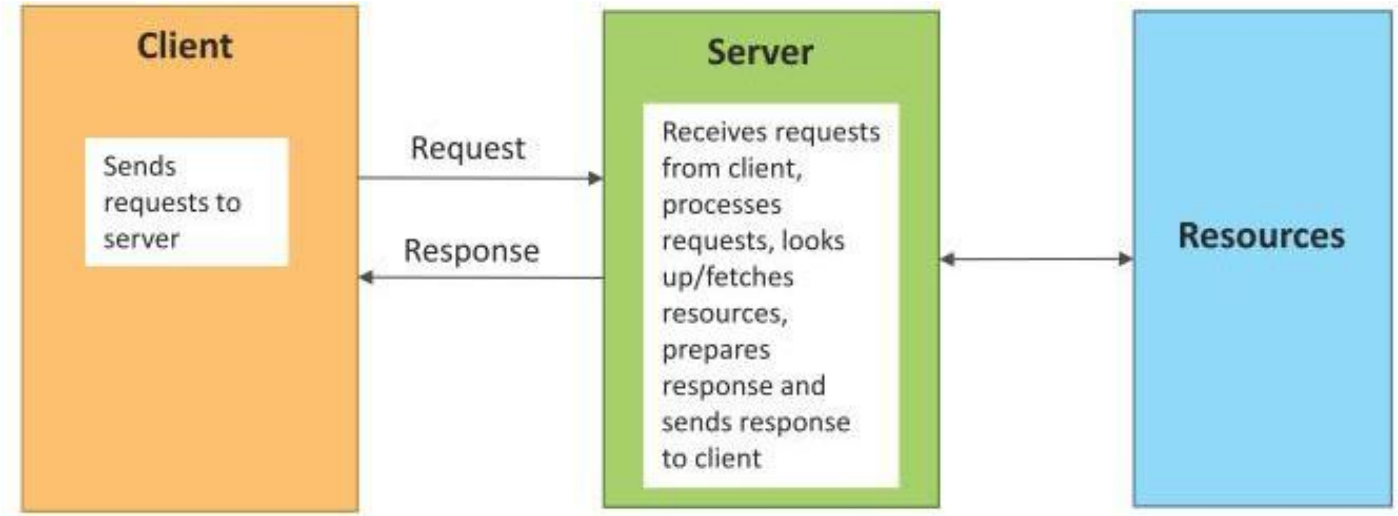
2. IoT Communication Model

- Following are the different communication model

1. Request Response Model
2. Publish Subscribe Model
3. Push-Pull Model
4. Exclusive Pair Model

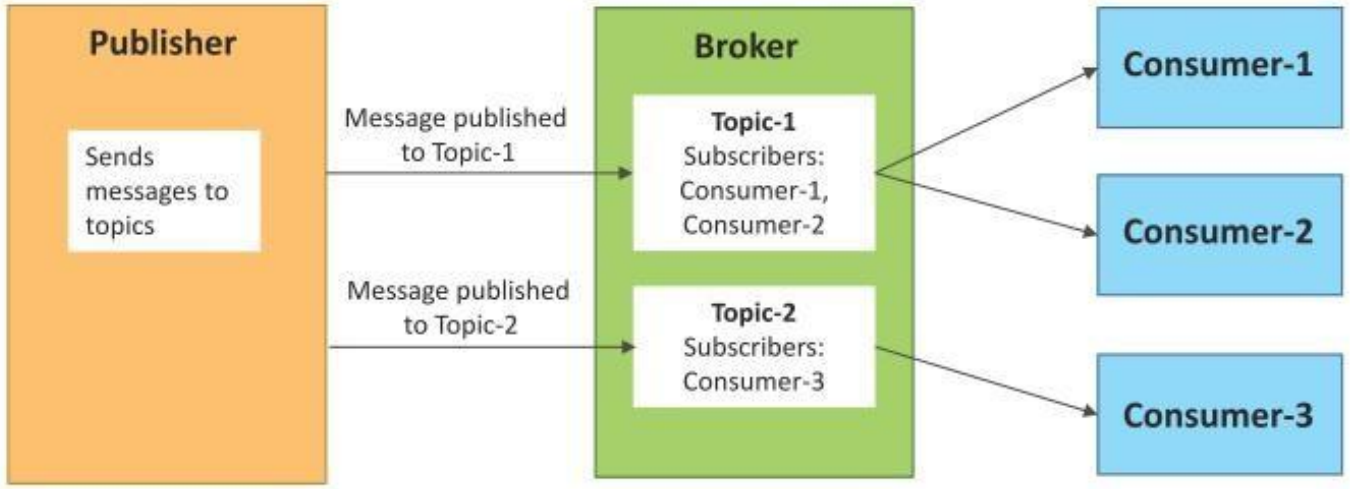
Request-Response communication model

- Request-Response is a communication model in which the client sends requests to the server and the server responds to the requests.
- When the server receives a request, it decides how to respond, fetches the data, retrieves resource representations, prepares the response, and then sends the response to the client.



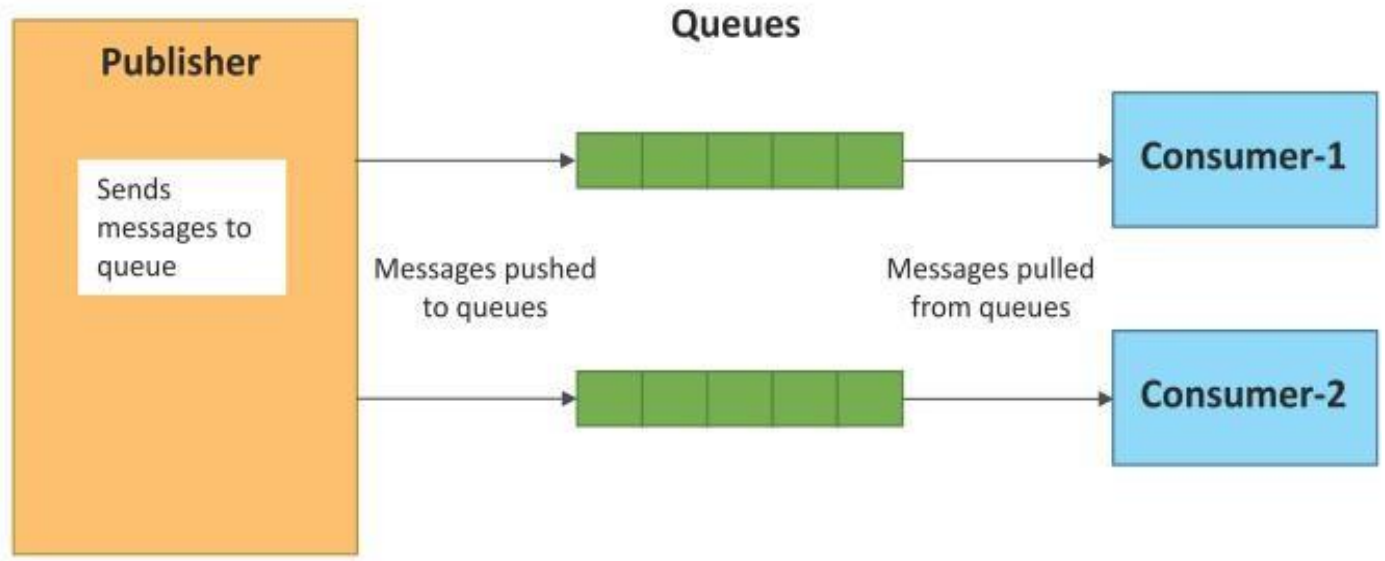
Publish-Subscribe communication model

- Publish-Subscribe is a communication model that involves publishers, brokers and consumers.
- Publishers are the source of data. Publishers send the data to the topics which are managed by the broker. Publishers are not aware of the consumers.
- Consumers subscribe to the topics which are managed by the broker.
- When the broker receives data for a topic from the publisher, it sends the data to all the subscribed consumers.



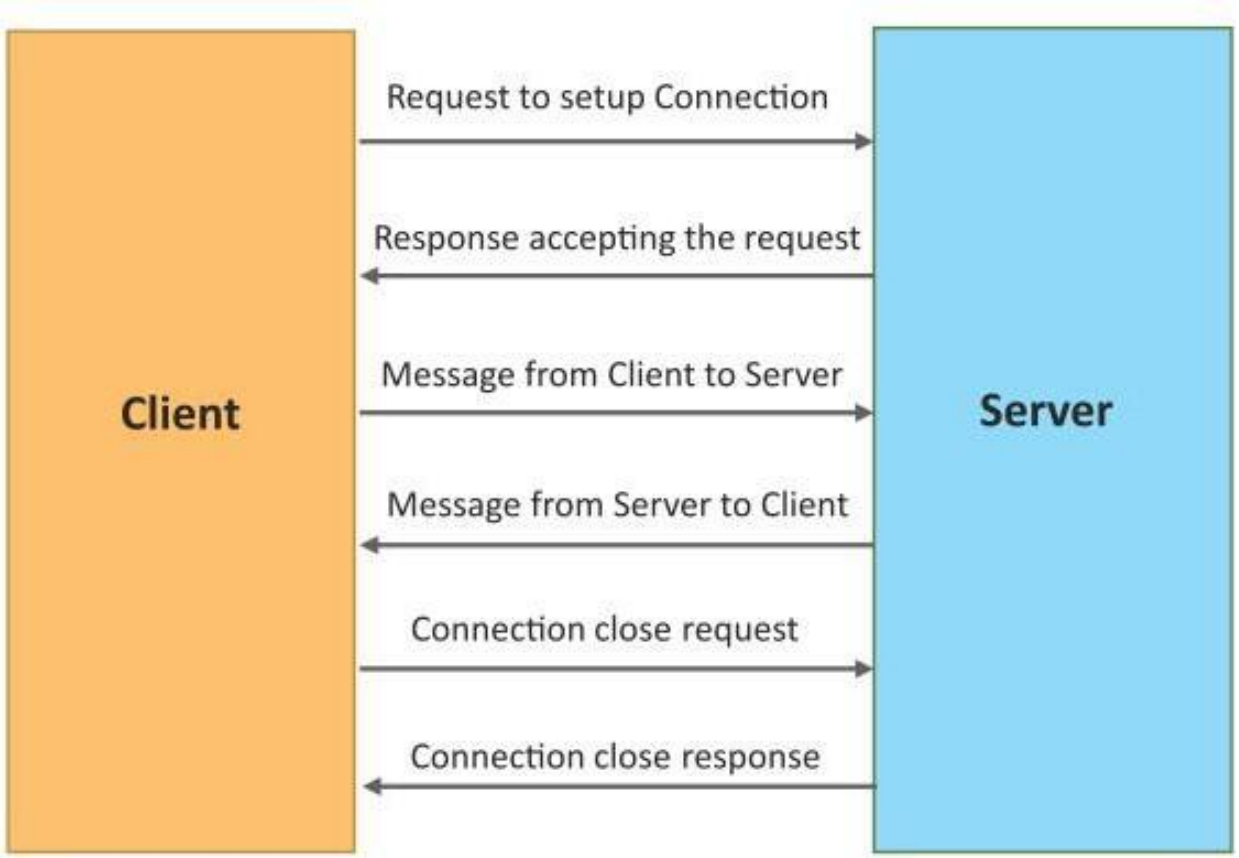
Push-Pull communication model

- Push-Pull is a communication model in which the data producers push the data to queues and the consumers pull the data from the queues. Producers do not need to be aware of the consumers.
- Queues help in decoupling the messaging between the producers and consumers.
- Queues also act as a buffer which helps in situations when there is a mismatch between the rate at which the producers push data and the rate rate at which the consumers pull data.



Exclusive Pair communication model

- Exclusive Pair is a bidirectional, fully duplex communication model that uses a persistent connection between the client and server.
- Once the connection is setup it remains open until the client sends a request to close the connection.
- Client and server can send messages to each other after connection setup.



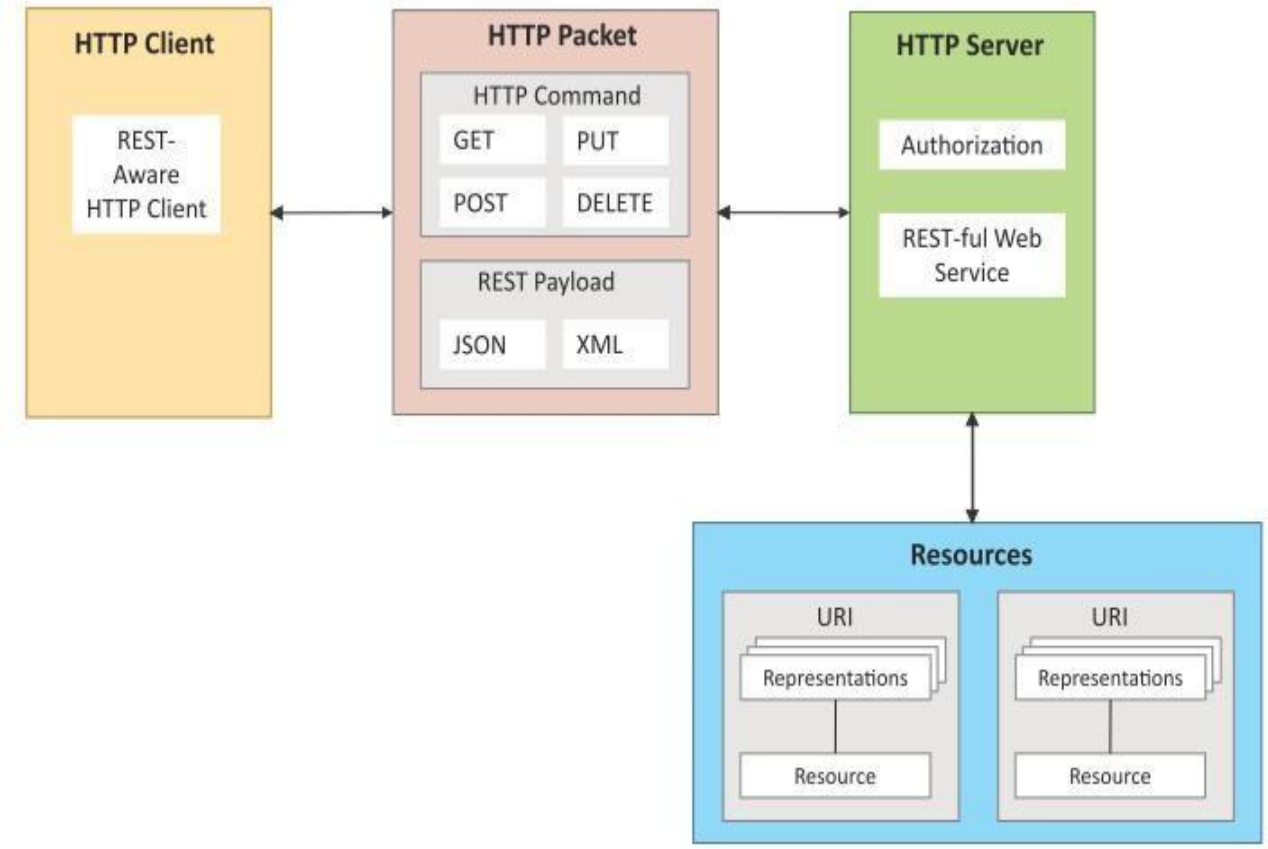
- Following are the different IoT Communication API's

1. REST based communication API's

2. Web-Socket based communication API's

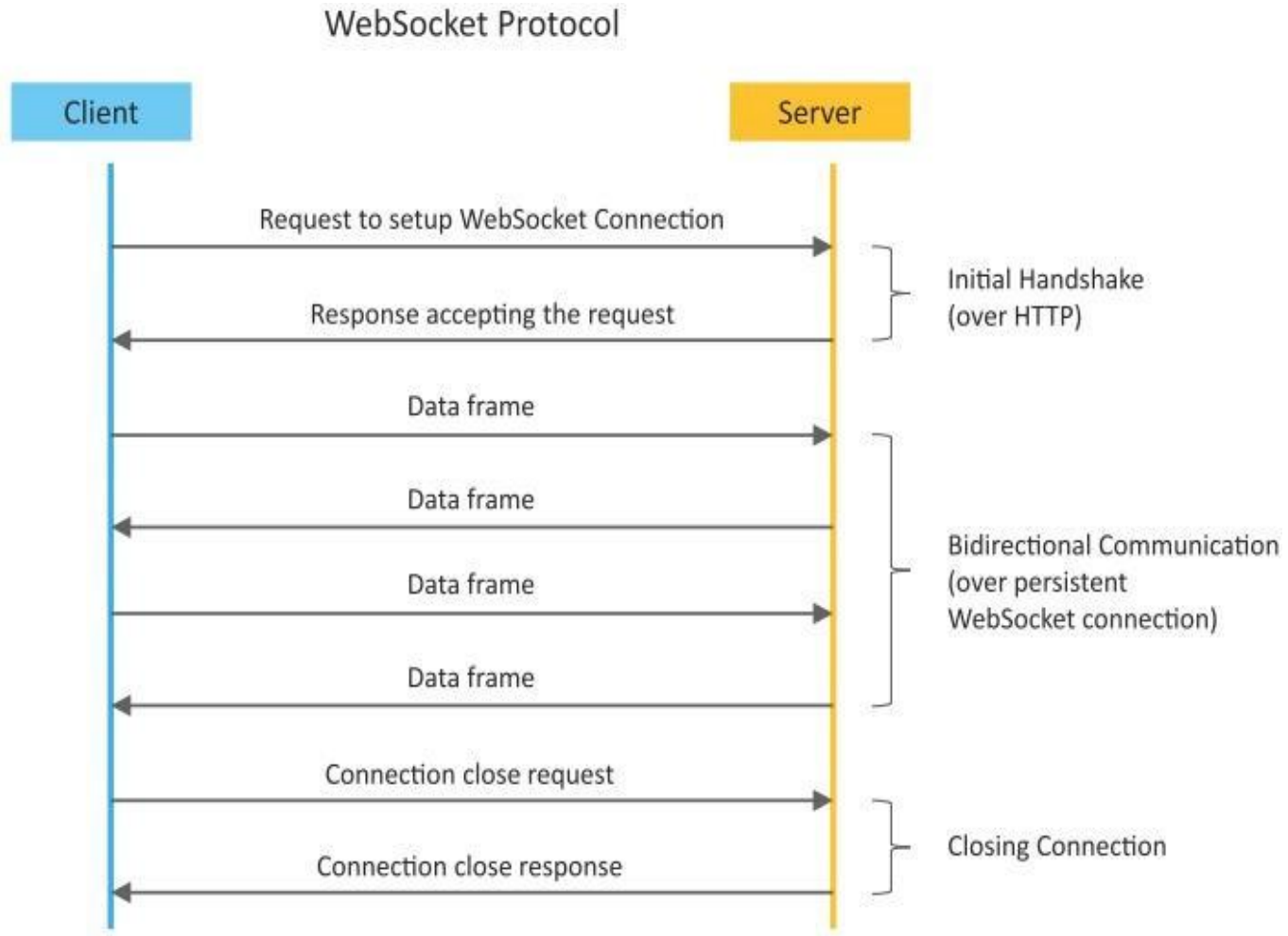
REST-based Communication APIs

- Representational State Transfer (REST) is a set of architectural principles by which you can design web services and web APIs that focus on a system's resources and how resource states are addressed and transferred.
- REST APIs follow the request-response communication model.
- The REST architectural constraints apply to the components, connectors, and data elements, within a distributed hypermedia system.



WebSocket-based Communication APIs

- WebSocket APIs allow bi-directional, full duplex communication between clients and servers.
- WebSocket APIs follow the exclusive pair communication model



IoT Enabling Technologies

IoT is enabled by several technologies including Wireless Sensor Networks, Cloud Computing, Big Data Analytics, Embedded Systems, Security Protocols and architectures, Communication Protocols, Web Services, Mobile internet and semantic search engines.

1. Wireless Sensor Network (WSN)
2. Cloud Computing
3. Big Data Analytics
4. Communication Protocols
5. Embedded Systems

IoT Enabling Technologies

- **Wireless Sensor Network (WSN)**

1. Weather Monitoring System
2. Indoor air quality monitoring system
3. Soil Moisture Monitoring System
4. Smart Grids
5. Structural Health Monitoring System

IoT Enabling Technologies

- **Cloud Computing**

1. Infrastructure as a service

2. Platform as a service

3. Software as a service

IoT Enabling Technologies

- **Big Data Analytics**

1. Sensor data generator
2. Machine sensor data collector
3. Health and fitness data
4. Location and Tracking data
5. Inventory Monitoring data

IoT Enabling Technologies

- **Communication Protocols**

1. Allow devices to exchange data

2. Define the exchange format

3. Sequence control , flow control and retransmission

IoT Enabling Technologies

- **Embedded Systems:**

Embedded System is a computer system that has computer hardware and software embedded to perform specific tasks. Embedded System range from low cost miniaturized devices such as digital watches to devices such as digital cameras, POS terminals, vending machines, appliances etc.

IoT Levels & Deployment Templates

An IoT system comprises of the following components:

- **Device**
- **Resource**
- **Controller Service**
- **Data Base**
- **Web Service**
- **Analysis component**
- **Application**

IoT Levels & Deployment Templates

An IoT system comprises of the following components:

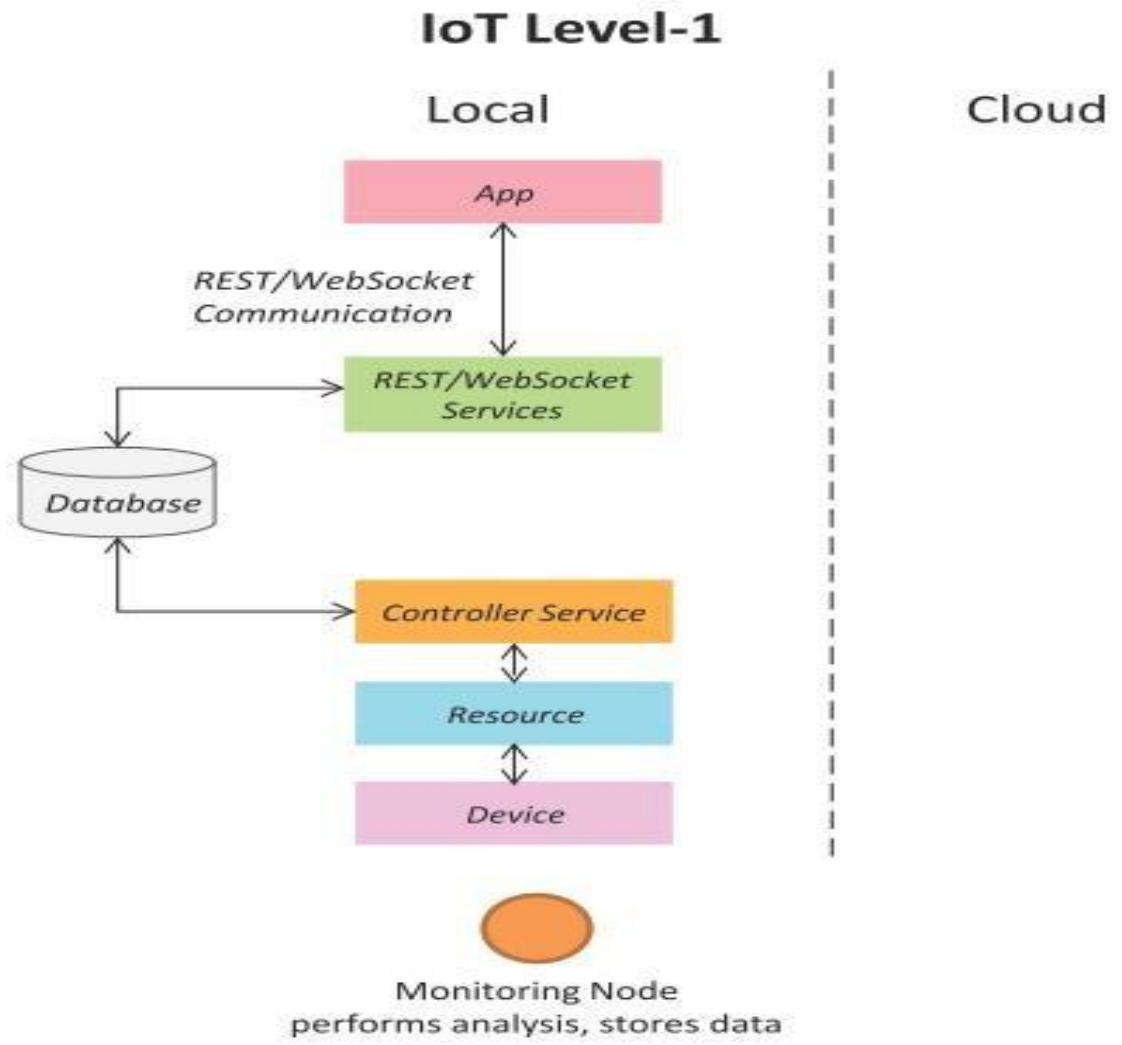
- **Device**: An IoT device allows identification, remote sensing, actuating and remote monitoring capabilities. You learned about various examples of IoT devices in section
- **Resource**: Resources are software components on the IoT device for accessing, processing, and storing sensor information, or controlling actuators connected to the device. Resources also include the software components that enable network access for the device.
- **Controller Service**: Controller service is a native service that runs on the device and interacts with the web services. Controller service sends data from the device to the web service and receives commands from the application (via web services) for controlling the device.

IoT Levels & Deployment Templates

- **Database**: Database can be either local or in the cloud and stores the data generated by the IoT device.
- **Web Service**: Web services serve as a link between the IoT device, application, database and analysis components. Web service can be either implemented using HTTP and REST principles (REST service) or using WebSocket protocol (WebSocket service).
- **Analysis Component**: The Analysis Component is responsible for analyzing the IoT data and generate results in a form which are easy for the user to understand.
- **Application**: IoT applications provide an interface that the users can use to control and monitor various aspects of the IoT system. Applications also allow users to view the system status and view the processed data.

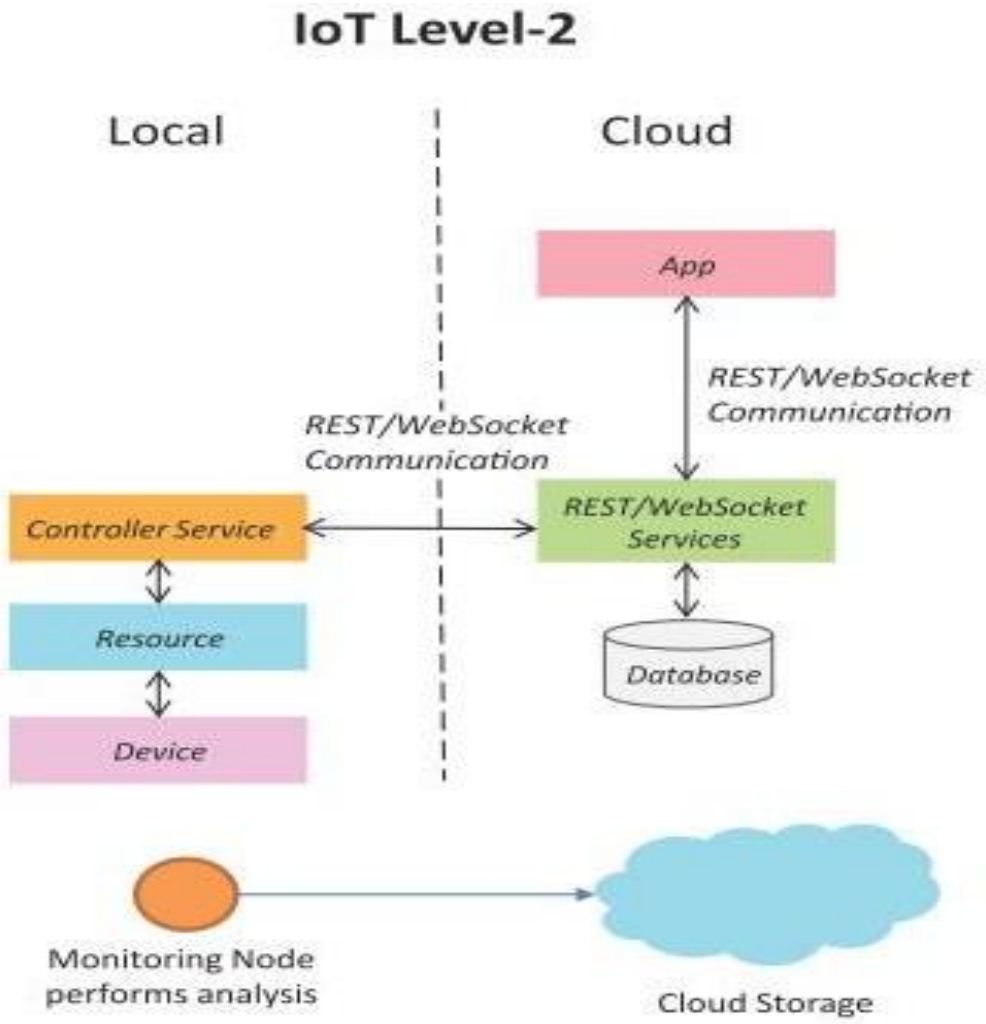
IoT Level-1

- A level-1 IoT system has a single node/device that performs sensing and/or actuation, stores data, performs analysis and hosts the application
- Level-1 IoT systems are suitable for modeling low-cost and low-complexity solutions where the data involved is not big and the analysis requirements are not computationally intensive.



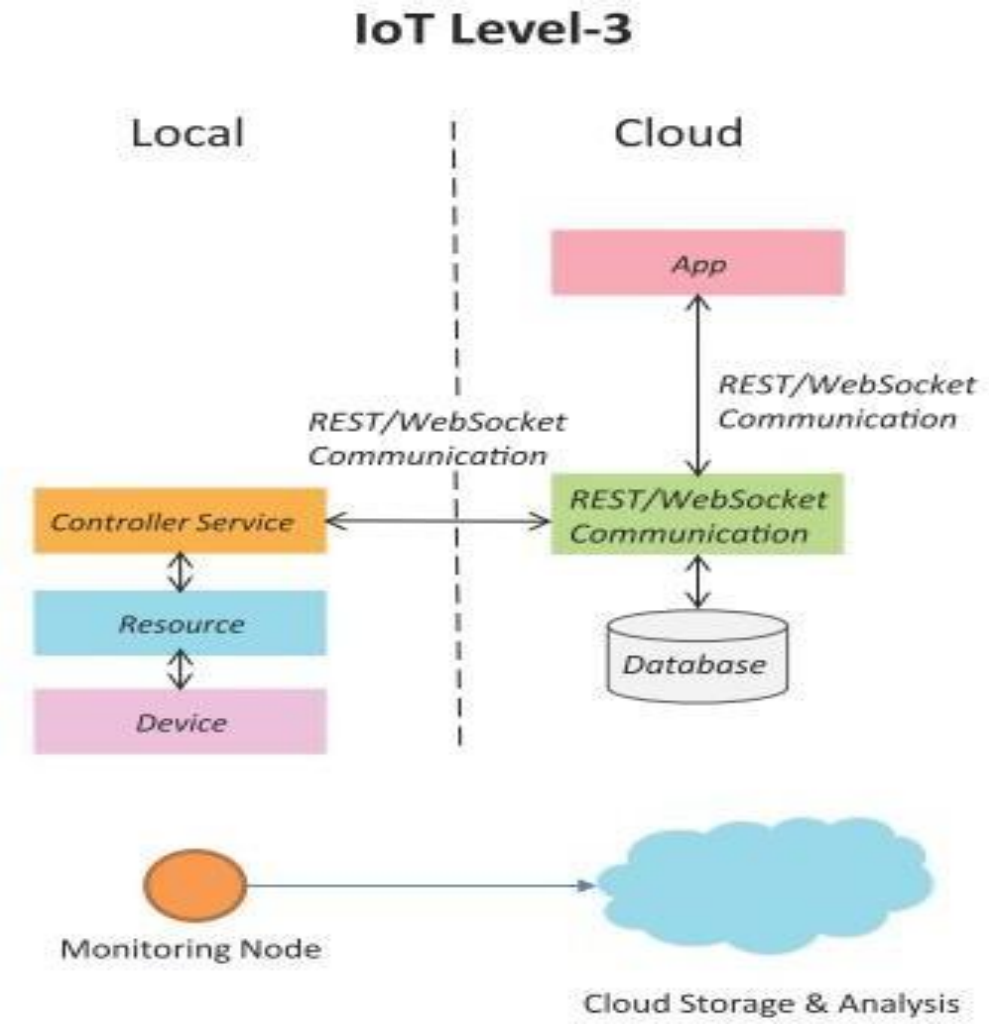
IoT Level-2

- A level-2 IoT system has a single node that performs sensing and/or actuation and local analysis.
- Data is stored in the cloud and application is usually cloud-based.
- Level-2 IoT systems are suitable for solutions where the data involved is big, however, the primary analysis requirement is not computationally intensive and can be done locally itself.



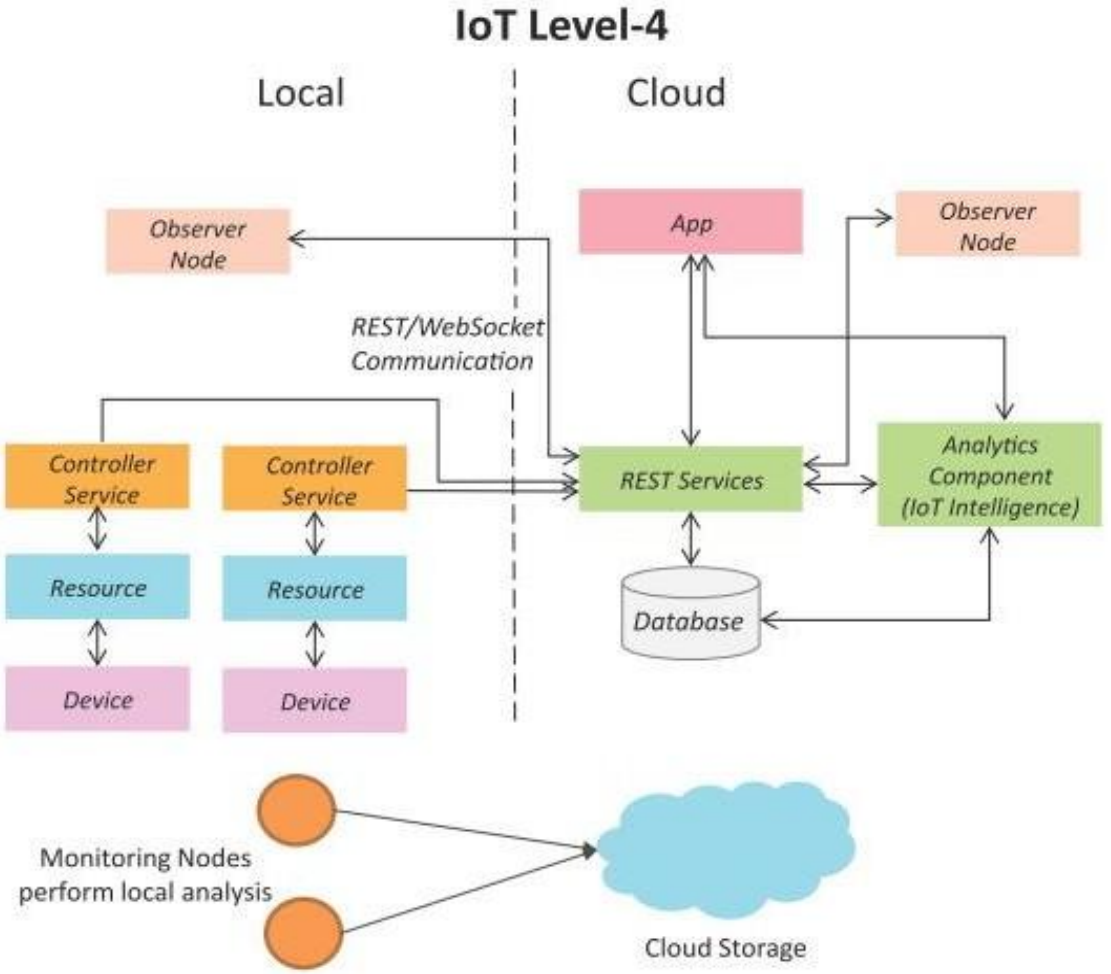
IoT Level-3

- A level-3 IoT system has a single node. Data is stored and analyzed in the cloud and application is cloud- based.
- Level-3 IoT systems are suitable for solutions where the data involved is big and the analysis requirements are computationally intensive.



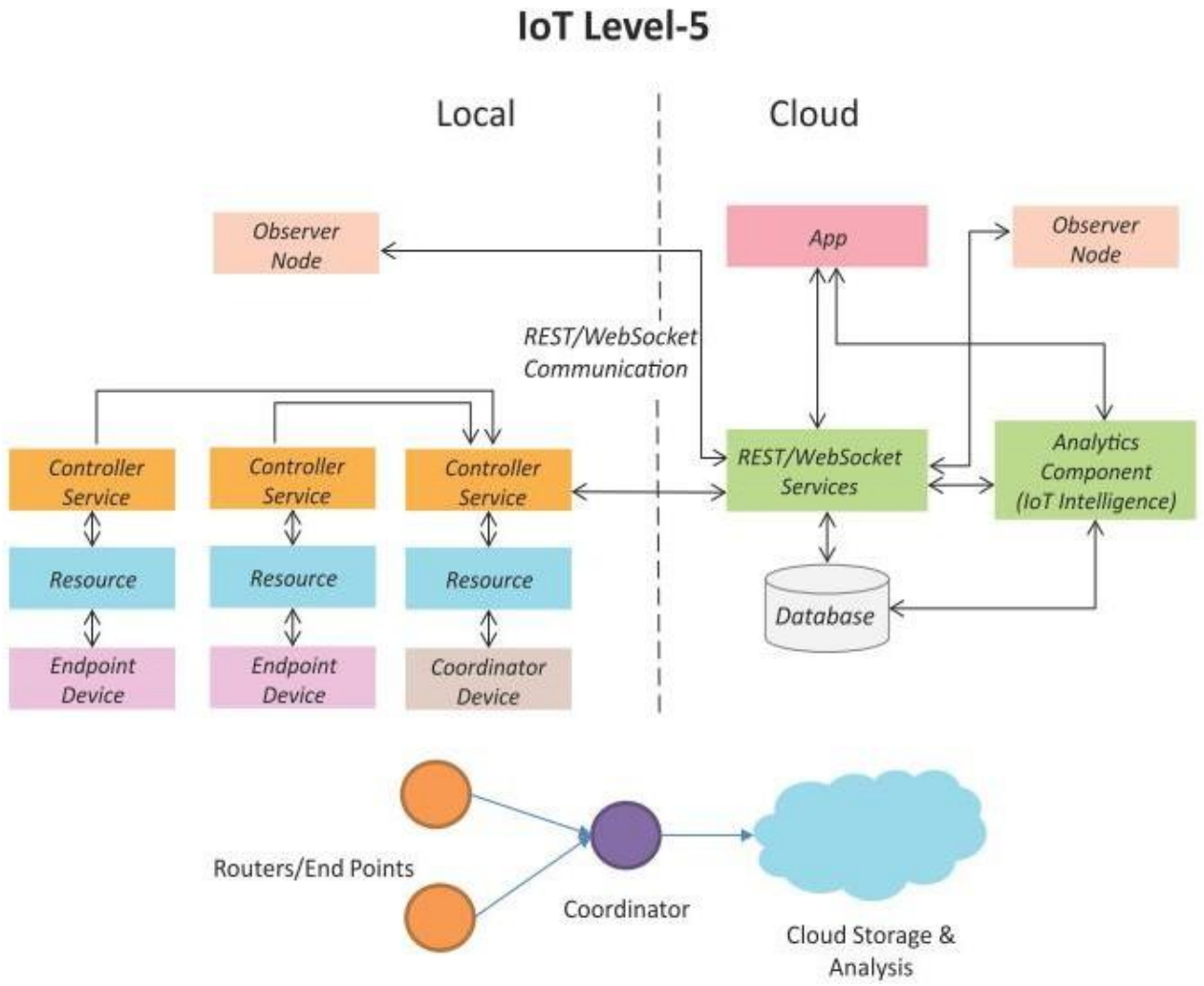
IoT Level-4

- A level-4 IoT system has multiple nodes that perform local analysis. Data is stored in the cloud and application is cloud-based.
- Level-4 contains local and cloud-based observer nodes which can subscribe to and receive information collected in the cloud from IoT devices.
- Level-4 IoT systems are suitable for solutions where multiple nodes are required, the data involved is big and the analysis requirements are computationally intensive.



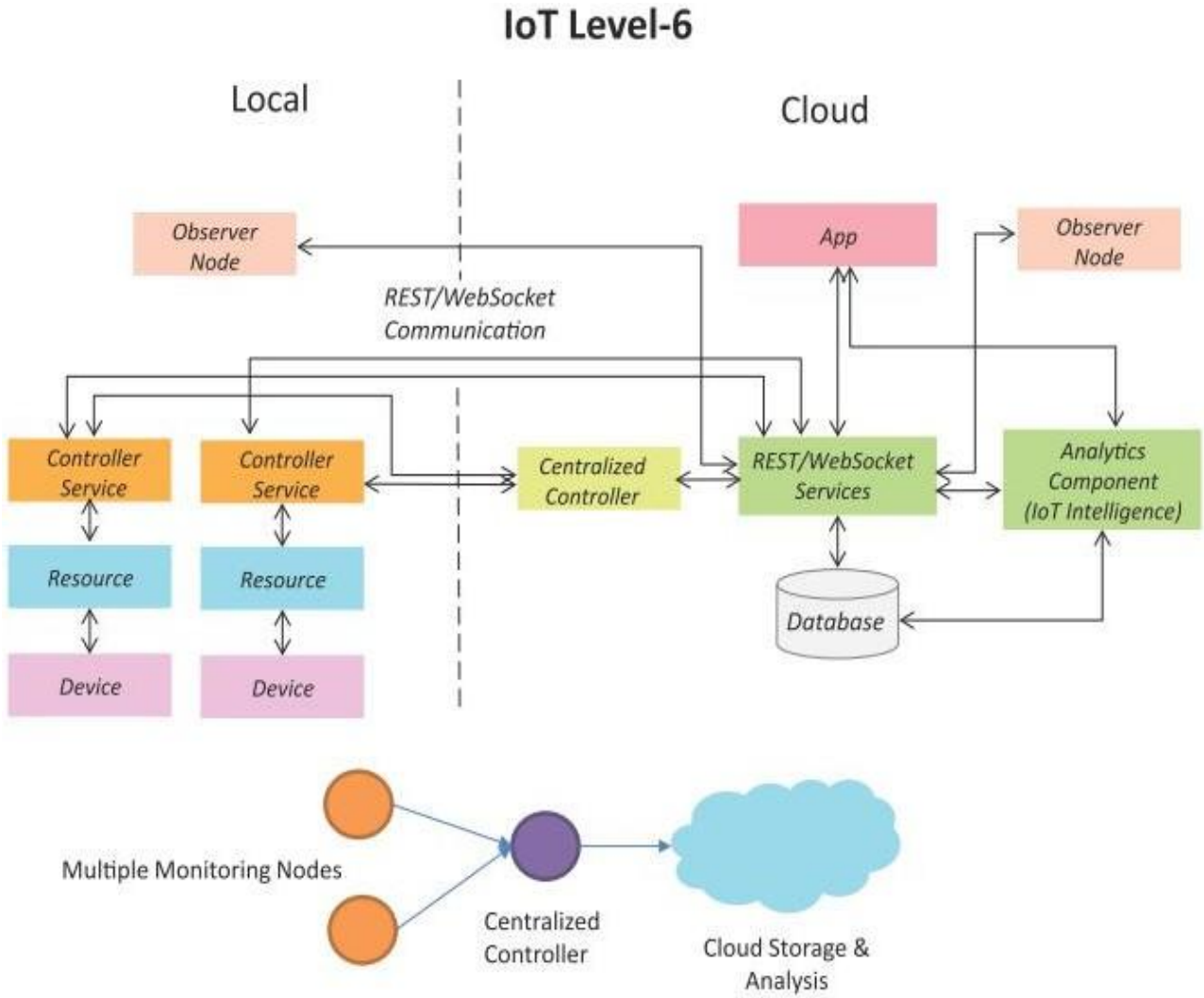
IoT Level-5

- A level-5 IoT system has multiple end nodes and one coordinator node.
- The end nodes that perform sensing and/or actuation.
- Coordinator node collects data from the end nodes and sends to the cloud.
- Data is stored and analyzed in the cloud and application is cloud-based.
- Level-5 IoT systems are suitable for solutions based on wireless sensor networks, in which the data involved is big and the analysis requirements are computationally intensive.



IoT Level-6

- A level-6 IoT system has multiple independent end nodes that perform sensing and/or actuation and send data to the cloud.
- Data is stored in the cloud and application is cloud-based.
- The analytics component analyzes the data and stores the results in the cloud database.
- The results are visualized with the cloud-based application.
- The centralized controller is aware of the status of all the end nodes and sends control commands to the nodes.



CONVERGENCE OF IT AND IOT

- IT supports connections to the Internet along with related data and technology systems and is focused on the secure flow of data across an organization.
- OT monitors and controls devices and processes on physical operational systems
- These systems include assembly lines, utility distribution networks, production facilities, roadway systems, and many more

- The IT organization is responsible for the information systems of a business, such as email, file and print services, databases, and so on.
- OT is responsible for the devices and processes acting on industrial equipment, such as factory machines, meters, actuators, electrical distribution automation devices and so on
- IT vs OT

Criterion	Industrial OT Network	Enterprise IT Network
Operational focus	Keep the business operating 24x7	Manage the computers, data, and employee communication system in a secure way
Priorities	1. Availability 2. Integrity 3. Security	1. Security 2. Integrity 3. Availability
Types of data	Monitoring, control, and supervisory data	Voice, video, transactional, and bulk data
Security	Controlled physical access to devices	Devices and users authenticated to the network
Implication of failure	OT network disruption directly impacts business	Can be business impacting, depending on industry, but workarounds may be possible
Network upgrades (software or hardware)	Only during operational maintenance windows	Often requires an outage window when workers are not onsite; impact can be mitigated
Security vulnerability	Low: OT networks are isolated and often use proprietary protocols	High: continual patching of hosts is required, and the network is connected to Internet and requires vigilant protection

Source: Maciej Kranz, *IT Is from Venus, OT Is from Mars*, blogs.cisco.com/digital/it-is-from-venus-ot-is-from-mars, July 14, 2015.

- With the rise of IoT and standards-based protocols, such as IPv6, the IT and OT worlds are converging
- OT is beginning to adopt the network protocols, technology, transport, and methods of the IT organization, and
- The IT organization is beginning to support the operational requirements used by OT

- The convergence of IT and OT to a single consolidated network poses several challenges
- The OT organization is baffled when IT schedules a weekend shutdown to update software without regard to production requirements
- When the IT team deploys QoS, voice and video traffic are almost universally treated with the highest level of service.
- However, when the OT system shares the same network, a very strong argument can be made that the real-time OT traffic should be given a higher priority

IOT CHALLENGES

- Scale
- Security
- Privacy
- Big data and data analytics
- Interoperability

Challenge	Description
Scale	While the scale of IT networks can be large, the scale of OT can be several orders of magnitude larger. For example, one large electrical utility in Asia recently began deploying IPv6-based smart meters on its electrical grid. While this utility company has tens of thousands of employees (which can be considered IP nodes in the network), the number of meters in the service area is tens of millions. This means the scale of the network the utility is managing has increased by more than 1,000-fold! Chapter 5, "IP as the IoT Network Layer," explores how new design approaches are being developed to scale IPv6 networks into the millions of devices.
Security	With more "things" becoming connected with other "things" and people, security is an increasingly complex issue for IoT. Your threat surface is now greatly expanded, and if a device gets hacked, its connectivity is a major concern. A compromised device can serve as a launching point to attack other devices and systems. IoT security is also pervasive across just about every facet of IoT. For more information on IoT security, see Chapter 8, "Securing IoT."

Privacy	As sensors become more prolific in our everyday lives, much of the data they gather will be specific to individuals and their activities. This data can range from health information to shopping patterns and transactions at a retail establishment. For businesses, this data has monetary value. Organizations are now discussing who owns this data and how individuals can control whether it is shared and with whom.
Big data and data analytics	IoT and its large number of sensors is going to trigger a deluge of data that must be handled. This data will provide critical information and insights if it can be processed in an efficient manner. The challenge, however, is evaluating massive amounts of data arriving from different sources in various forms and doing so in a timely manner. See Chapter 7 for more information on IoT and the challenges it faces from a big data perspective.
Interoperability	As with any other nascent technology, various protocols and architectures are jockeying for market share and standardization within IoT. Some of these protocols and architectures are based on proprietary elements, and others are open. Recent IoT standards are helping minimize this problem, but there are often various protocols and implementations available for IoT networks. The prominent protocols and architectures—especially open, standards-based implementations—are the subject of this book. For more information on IoT architectures, see Chapter 2, "IoT Network Architecture and Design." Chapter 4, "Connecting Smart Objects," Chapter 5, "IP as the IoT Network Layer," and Chapter 6, "Application Protocols for IoT," take a more in-depth look at the protocols that make up IoT.

IoT Network Architecture and Design

- Imagine that one day you decide to build a house
- To successfully complete a construction project, time and effort are required to design each phase, from the foundation to the roof.
- Your plans must include detailed designs for the electrical, plumbing, heating, and security systems

- A computer network should be built with-- careful planning, security policies, and adherence to well-understood design practices.
- Failure to meet these will likely result in something that is difficult to scale, manage, adapt to organizational changes, and, worst of all, troubleshoot when things go wrong
- If the network fails, company operations can be seriously impaired

- Just as a house must be designed with the strength to withstand potential natural disasters, such as seismic events and hurricanes,
- information technology (IT) systems need to be designed to withstand “network earthquakes,” such as
- distributed denial of service (DDoS) attacks,
- future growth requirements,
- network outages, and
- even human error

DRIVERS BEHIND NEW NETWORK ARCHITECTURES

- Building residential houses vs building a massive stadium..
- The difference between IT and IoT networks is much like the difference between residential architecture and stadium architecture
- The key difference between IT and IoT is the data.
- IT systems are mostly concerned with reliable and continuous support of business applications such as email, web, databases, CRM systems, and so on.
- IoT is all about the data generated by sensors and how that data is used.

- The essence of IoT architectures thus involves how the data is transported, collected, analyzed, and ultimately acted upon

Challenge	Description	IoT Architectural Change Required
Scale	The massive scale of IoT endpoints (sensors) is far beyond that of typical IT networks.	The IPv4 address space has reached exhaustion and is unable to meet IoT's scalability requirements. Scale can be met only by using IPv6. IT networks continue to use IPv4 through features like Network Address Translation (NAT).
Security	IoT devices, especially those on wireless sensor networks (WSNs), are often physically exposed to the world.	Security is required at every level of the IoT network. Every IoT endpoint node on the network must be part of the overall security strategy and must support device-level authentication and link encryption. It must also be easy to deploy with some type of a zero-touch deployment model.
Devices and networks constrained by power, CPU, memory, and link speed	Due to the massive scale and longer distances, the networks are often constrained, lossy, and capable of supporting only minimal data rates (tens of bps to hundreds of Kbps).	New last-mile wireless technologies are needed to support constrained IoT devices over long distances. The network is also constrained, meaning modifications need to be made to traditional network-layer transport mechanisms.

The massive volume of data generated	The sensors generate a massive amount of data on a daily basis, causing network bottlenecks and slow analytics in the cloud.	Data analytics capabilities need to be distributed throughout the IoT network, from the edge to the cloud. In traditional IT networks, analytics and applications typically run only in the cloud.
Support for legacy devices	An IoT network often comprises a collection of modern, IP-capable endpoints as well as legacy, non-IP devices that rely on serial or proprietary protocols.	Digital transformation is a long process that may take many years, and IoT networks need to support protocol translation and/or tunneling mechanisms to support legacy protocols over standards-based protocols, such as Ethernet and IP.
The need for data to be analyzed in real time	Whereas traditional IT networks perform scheduled batch processing of data, IoT data needs to be analyzed and responded to in real-time.	Analytics software needs to be positioned closer to the edge and should support real-time streaming analytics. Traditional IT analytics software (such as relational databases or even Hadoop), are better suited to batch-level analytics that occur after the fact.

Security

- IT networks use firewall, IT endpoints are behind firewall
- IoT endpoints are often located in wireless sensor networks that use unlicensed spectrum and are not only visible to the world through a spectrum analyzer but often physically
  accessible
- IoT systems require consistent mechanisms of authentication, encryption, and intrusion prevention techniques

- IoT systems must:
- Be able to identify and authenticate all entities involved in the IoT service (that is, gateways, endpoint devices, home networks, roaming networks, service platforms)
- Ensure that all user data shared between the endpoint device and back-end applications is encrypted
- Comply with local data protection legislation so that all data is protected and stored correctly
- Take network-level approach to security in addition to device level approach

Constrained Devices and Networks

- Most IoT sensors are designed for a single job, and they are typically small and inexpensive
- They often have limited power, CPU, and memory, and they transmit only when there is something important
- The networks that provide connectivity also tend to be very lossy and support very low data rates.
- This is a completely different situation from IT networks
- IoT requires a new breed of connectivity technologies that meet both the scale and constraint limitations

Legacy Device Support

- Supporting legacy devices in an IT organization is not usually a big problem
- If someone's computer or operating system is outdated, she simply upgrades
- If someone is using a mobile device with an outdated Wi-Fi standard, such as 802.11b or 802.11g, you can simply deny him access to the wireless network, and he will be forced to upgrade

- In OT systems, end devices are likely to be on the network for a very long time—sometimes decades.
- As IoT networks are deployed, they need to support the older devices already present on the network, as well as devices with new capabilities

COMPARING IOT ARCHITECTURES

- oneM2M architecture
- The IoT World Forum(IoTWF)

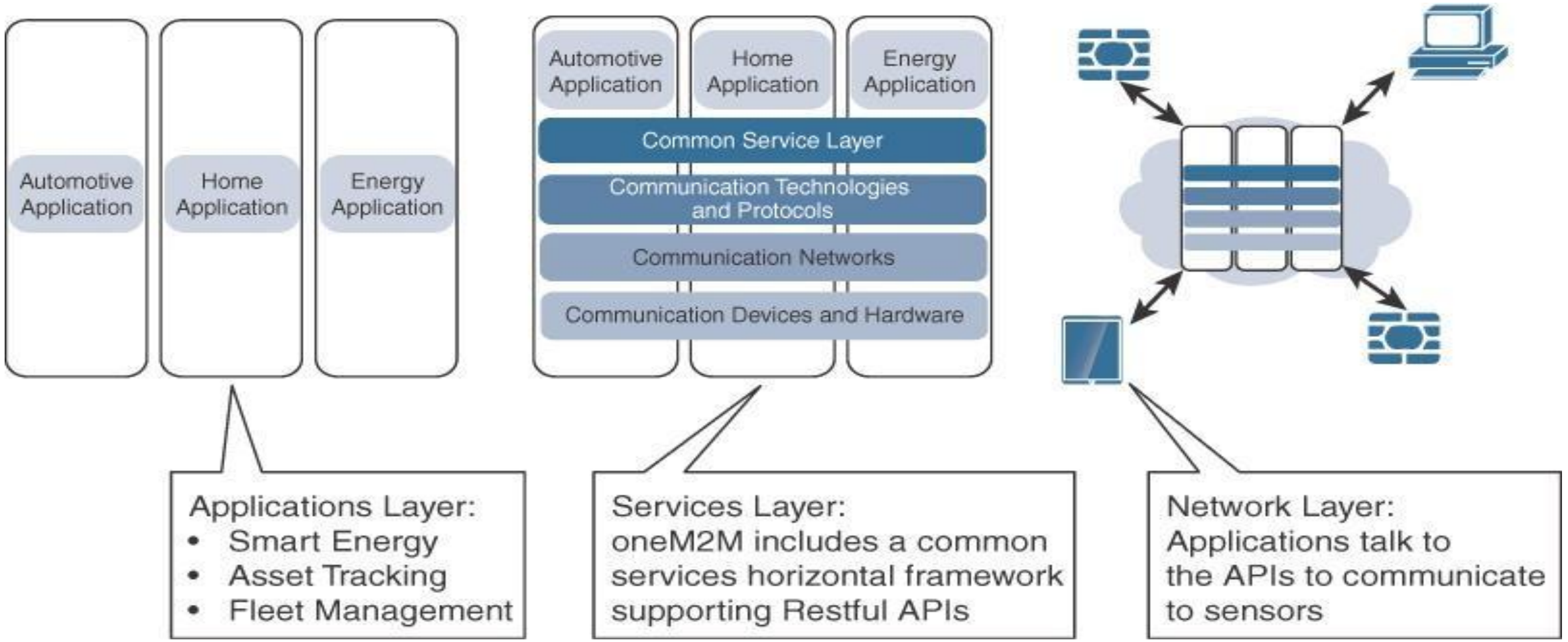
M2M IoT Standardized Architecture

- 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
- The goal of oneM2M is to create a common services layer, which can be readily embedded in field devices to allow communication with application servers
- oneM2M's framework focuses on IoT services, applications, and platforms

- One of the greatest challenges in designing an IoT architecture is dealing with the heterogeneity of devices, software, and access methods
- By developing a horizontal platform architecture, oneM2M is developing standards that allow interoperability at all levels of the IoT stack
- For example, you might want to automate your HVAC system by connecting it with wireless temperature sensors spread throughout your office

- The problem is that the LoRaWAN network and the BACnet system that your HVAC and BMS run on are completely
  different systems and have no natural connection point
- This is where the oneM2M common services architecture comes in.
- oneM2M’s horizontal framework and RESTful APIs allow the LoRaWAN system to interface with the building management system over an IoT network, thus promoting end-to-end IoT communications in a consistent way, no
  matter how heterogeneous the networks

The Main Elements of the oneM2M IoT Architecture



- Three major domains:
- The application layer,
- The services layer, and
- The network layer

Application layer

- The oneM2M architecture gives major attention to 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

- This layer is shown as a horizontal framework across the vertical industry applications
- At this layer, 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 networks, VPNs, and so on.
- Riding on top is the common services layer.
- This conceptual layer adds APIs and middleware supporting third-party services and applications

Network layer

- This is the communication domain for the IoT devices and endpoints.
- It includes the devices themselves and the communications network that links them
- Communications infrastructure include wireless mesh technologies, such as IEEE 802.15.4, and wireless point-to- multipoint systems, such as IEEE 801.11ah

The IoT World Forum (IoTWF) Standardized Architecture

- A seven-layer IoT architectural reference model published by IoTWF architectural committee (Cisco, IBM, Rockwell Automation)
- Edge computing
- Data storage
- Access

IoT Reference Model Published by the IoT World Forum

Levels

7 **Collaboration & Processes**
(Involving People & Business Processes)

6 **Application**
(Reporting, Analytics, Control)

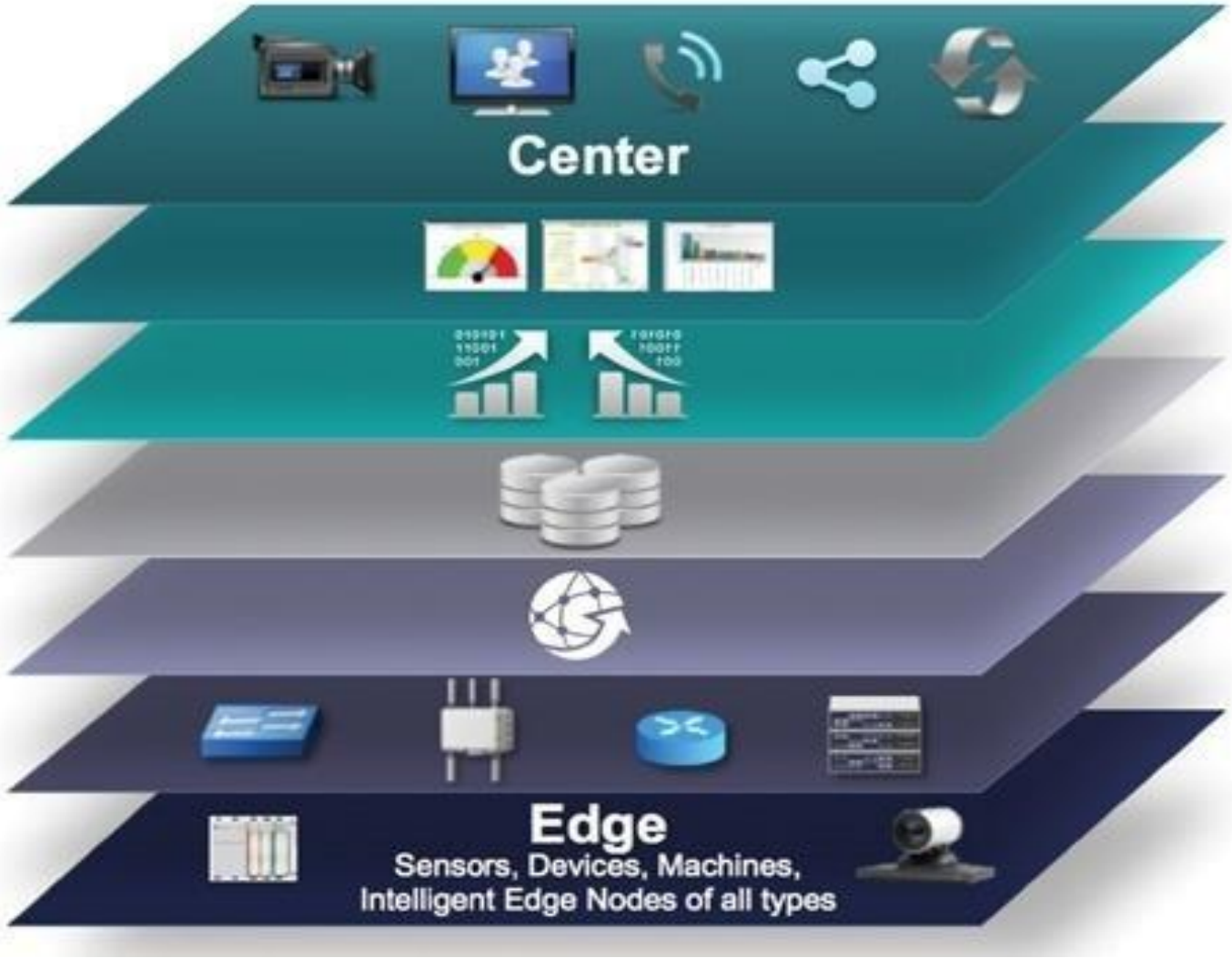
5 **Data Abstraction**
(Aggregation & Access)

4 **Data Accumulation**
(Storage)

3 **Edge Computing**
(Data Element Analysis & Transformation)

2 **Connectivity**
(Communication & Processing Units)

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



- Using this reference model, we are able to achieve the following:
- Decompose the IoT problem into smaller parts
- Identify different technologies at each layer and how they relate to one another
- Define a system in which different parts can be provided by different vendors
- Have a process of defining interfaces that leads to interoperability
- Define a tiered security model that is enforced at the transition points between levels

Layer 1: Physical Devices and Controllers Layer

- The various endpoint devices and sensors that send and receive information
- The size of these “things” can range from almost microscopic sensors to giant machines in a factory.
- Their primary function is generating data and being capable of being queried and/or controlled over a network.

- The most important function of this IoT layer is the reliable and timely transmission of data.
- More specifically, this includes transmissions between Layer 1 devices and the network and between the network and information processing that occurs at Layer 3 (the edge computing layer).

② **Connectivity**
(Communication and Processing Units)

Layer 2 Functions:

- Communications Between Layer 1 Devices
- Reliable Delivery of Information Across the Network
- Switching and Routing
- Translation Between Protocols
- Network Level Security

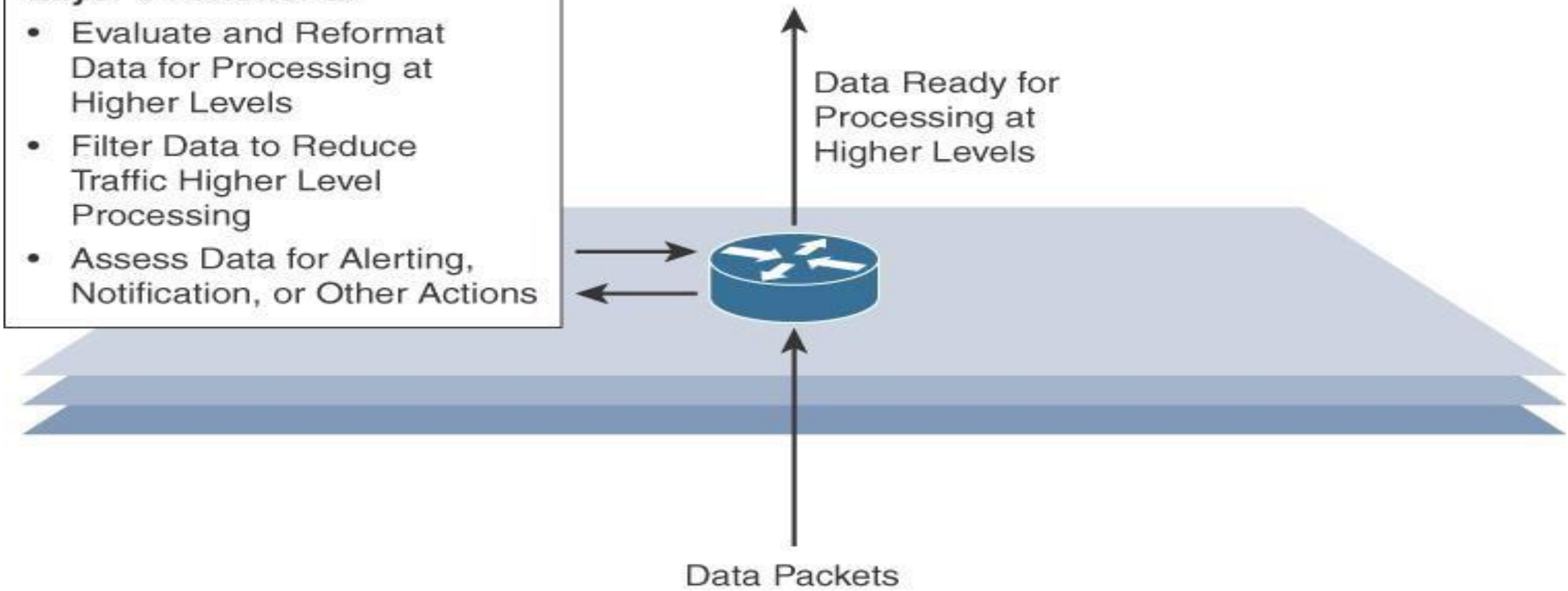
Layer 3: Edge Computing Layer

- Fog layer
- At this layer, the emphasis is on data reduction and converting network data flows into information that is ready for storage and processing by higher layers
- One of the basic principles of this reference model is that information processing is initiated as early and as close to the edge of the network as possible
- Another important function that occurs at Layer 3 is the evaluation of data to see if it can be filtered or aggregated before being sent to a higher layer
- This also allows for data to be reformatted or decoded, making additional processing by other systems easier

③ **Edge (Fog) Computing**
(Data Element Analysis and Transformation)

Layer 3 Functions:

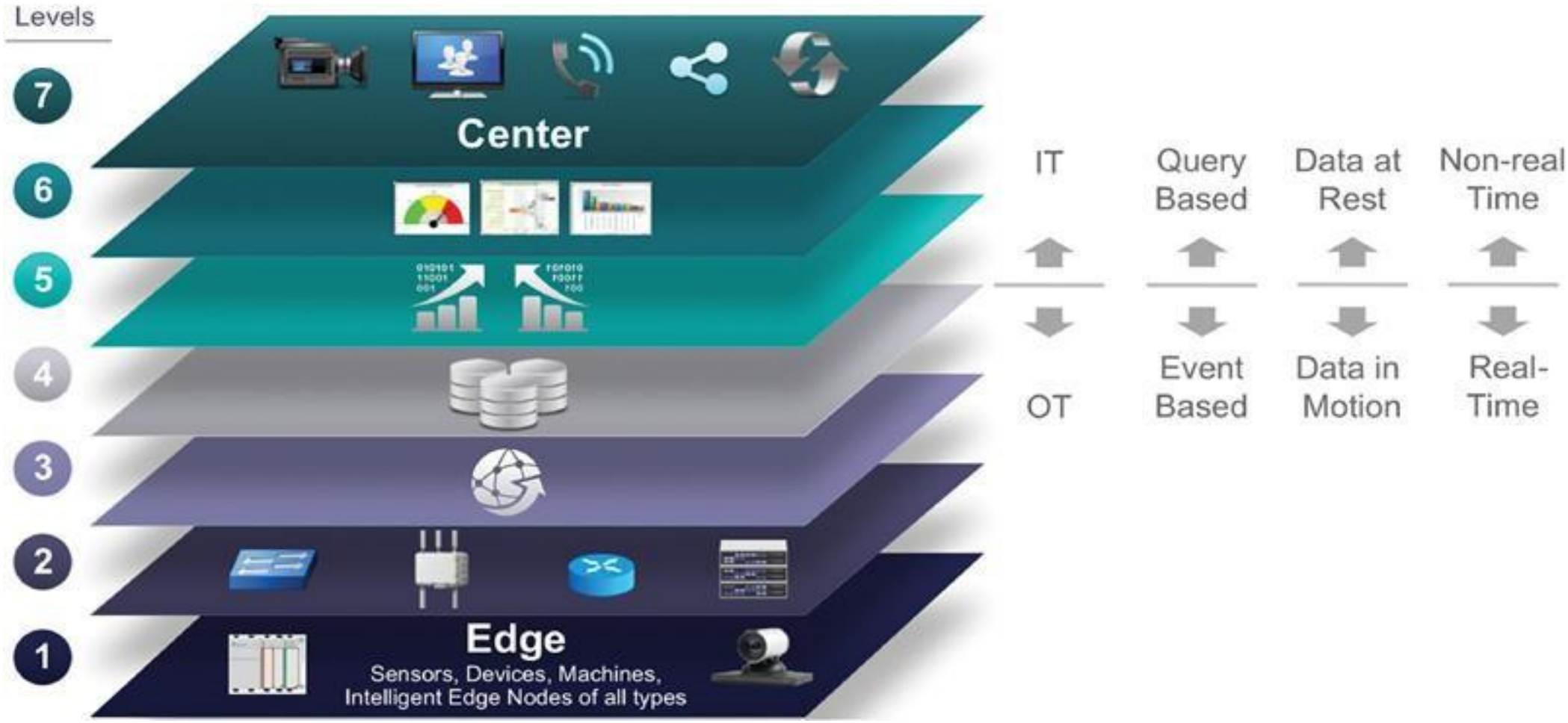
- Evaluate and Reformat Data for Processing at Higher Levels
- Filter Data to Reduce Traffic Higher Level Processing
- Assess Data for Alerting, Notification, or Other Actions



- The upper layers deal with handling and processing the IoT data generated by the bottom layer

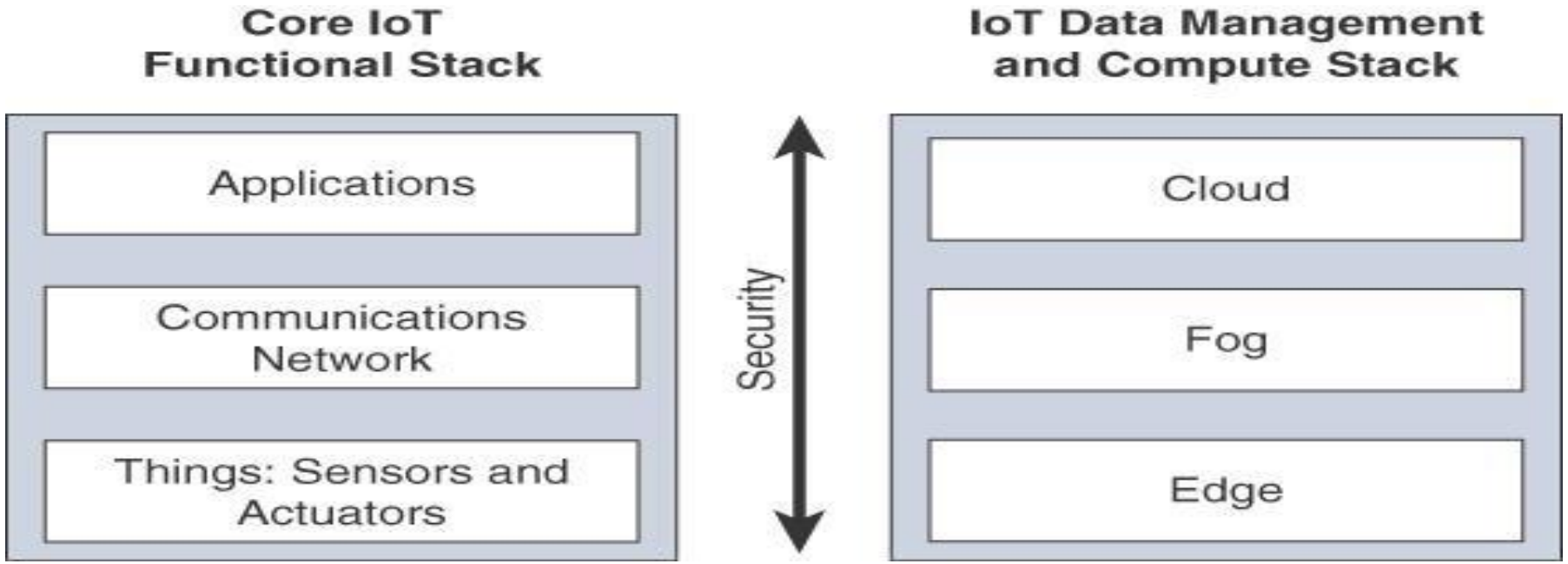
IoT Reference Model Layer	Functions
Layer 4: Data accumulation layer	Captures data and stores it so it is usable by applications when necessary. Converts event-based data to query-based processing.
Layer 5: Data abstraction layer	Reconciles multiple data formats and ensures consistent semantics from various sources. Confirms that the data set is complete and consolidates data into one place or multiple data stores using virtualization.
Layer 6: Applications layer	Interprets data using software applications. Applications may monitor, control, and provide reports based on the analysis of the data.
Layer 7: Collaboration and processes layer	Consumes and shares the application information. Collaborating on and communicating IoT information often requires multiple steps, and it is what makes IoT useful. This layer can change business processes and delivers the benefits of IoT.

IT and OT Responsibilities in the IoT Reference Model



A SIMPLIFIED IOT ARCHITECTURE

- The framework is presented as two parallel stacks:
- The IoT Data Management and Compute Stack and
- The Core IoT Functional Stack.



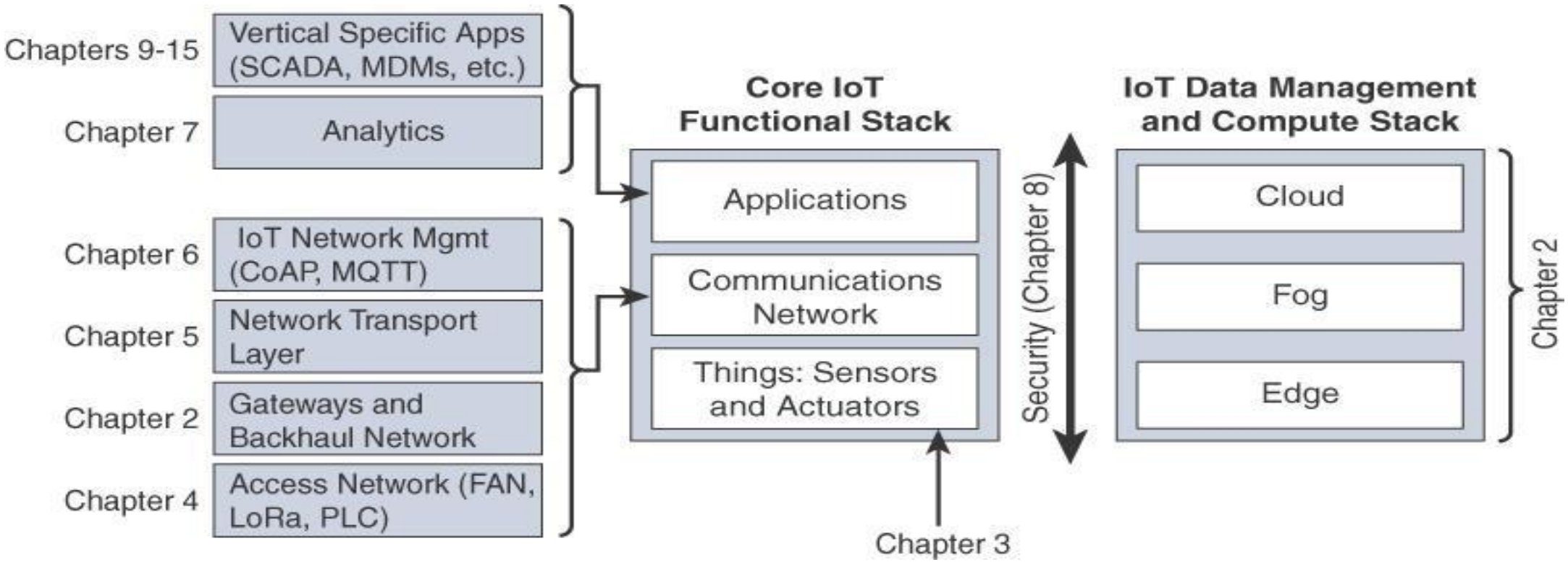
- The network communications layer of the IoT stack itself involves a significant amount of detail and incorporates a vast array of technologies.
- Consider for a moment the heterogeneity of IoT sensors and the many different ways that exist to connect them to a network.
- The network communications layer needs to consolidate these together, offer gateway and backhaul technologies, and ultimately bring the data back to a central location for analysis and processing.

- The network between the gateway and the data center is composed mostly of traditional technologies that experienced IT professionals would quickly recognize.
- These include tunneling and VPN technologies, IP-based quality of service (QoS), conventional Layer 3 routing protocols such as BGP and IP-PIM, and security capabilities such as encryption, access control lists (ACLs), and firewalls

- Unlike with most IT networks, the applications and analytics layer of IoT doesn’t necessarily exist only in the data center or in the cloud.
- Due to the unique challenges and requirements of IoT, it is often necessary to deploy applications and data management throughout the architecture in a tiered approach, allowing data collection, analytics, and intelligent controls at multiple points in the IoT system

- The three data management layers are
- The edge layer (data management within the sensors themselves),
- The fog layer (data management in the gateways and transit network), and
- The cloud layer (data management in the cloud or central data center).

Expanded View of the Simplified IoT Architecture



THE CORE IOT FUNCTIONAL STACK

- IoT networks are built around the concept of "things," or smart objects performing functions and delivering new connected services.
- These objects are "smart" because they use a combination of contextual information and configured goals to perform actions
- These actions can be self-contained (that is, the smart object does not rely on external systems for its actions); however, in most cases, the "thing" interacts with an external system to report information that the smart object collects, to exchange with other objects, or to interact with a management platform

- From an architectural standpoint, several components have to work together for an IoT network to be operational:
- **“Things” layer:**
- At this layer, the physical devices need to fit the constraints of the environment in which they are deployed while still being able to provide the information needed.
- **Communications network layer:**
- When smart objects are not self-contained, they need to communicate with an external system.
- In many cases, this communication uses a wireless technology.
- This layer has four sublayers:

- **Access network sub layer:**
- This is typically made up of wireless technologies such as 802.11ah, 802.15.4g, and LoRa.
- The sensors connected to the access network may also be wired.

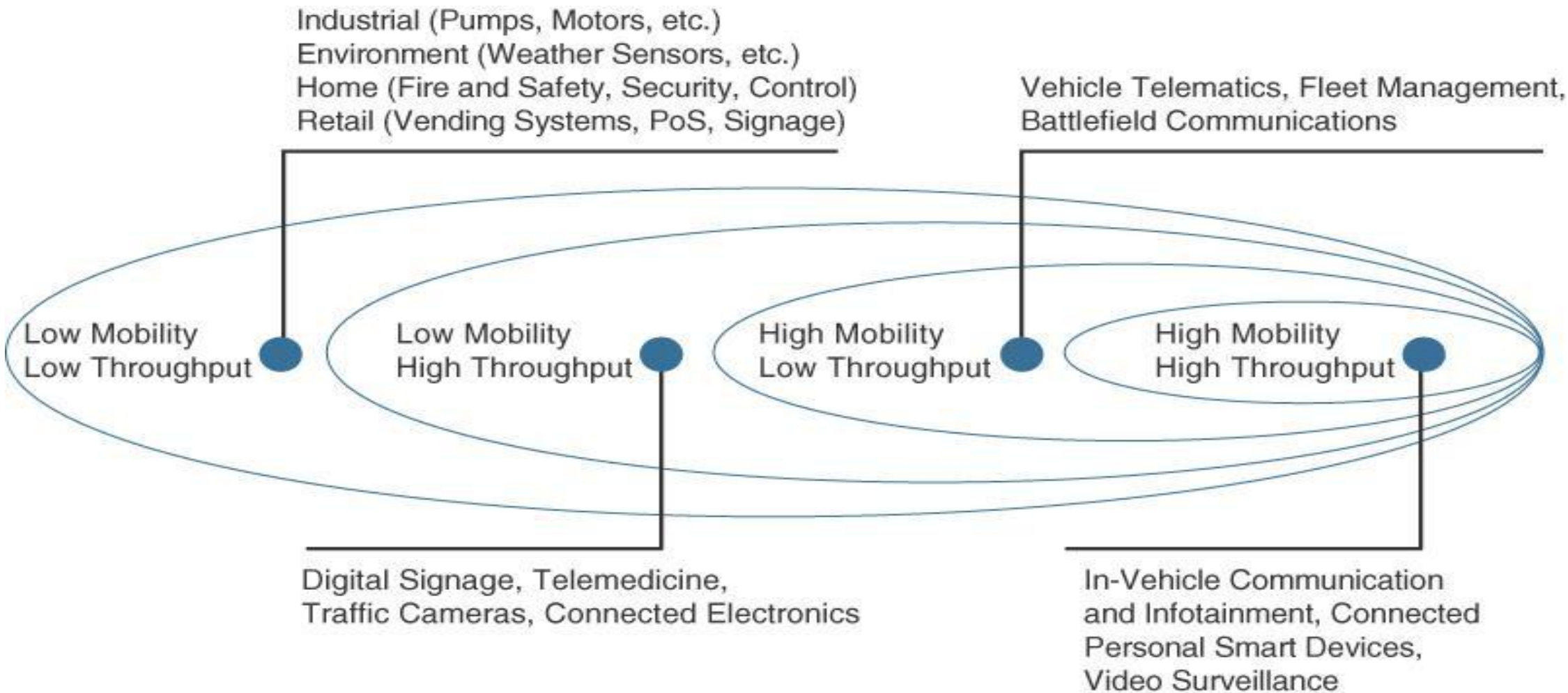
- **Gateways and backhaul network sub layer:**
- The gateway communicates directly with the smart objects.
- The role of the gateway is to forward the collected information through a longer-range medium (called the backhaul) to a headend central station where the information is processed
- **Network transport sub layer:**
- For communication to be successful, network and transport layer protocols such as IP and UDP must be implemented to support the variety of devices to connect and media to use

- **IoT network management sub layer:**
- Additional protocols must be in place to allow the headend applications to exchange data with the sensors.
- Examples include CoAP and MQTT.

- **Application and analytics layer:**
- At the upper layer, an application needs to process the collected data,
- To control the smart objects when necessary,
- To make intelligent decision based on the information collected and, in turn, instruct the “things” or other systems to adapt to the analyzed conditions and change their behaviors or parameters.

Layer 1: Things: Sensors and Actuators Layer

- Battery-powered or power-connected
- Mobile or static
- Low or high reporting frequency
- Simple or rich data
- Report range
- Object density per cell


Industrial (Pumps, Motors, etc.)
Environment (Weather Sensors, etc.)
Home (Fire and Safety, Security, Control)
Retail (Vending Systems, PoS, Signage)
Vehicle Telematics, Fleet Management,
Battlefield Communications
Low Mobility
Low Throughput
Low Mobility
High Throughput
High Mobility
Low Throughput
High Mobility
High Throughput
Digital Signage, Telemedicine,
Traffic Cameras, Connected Electronics
In-Vehicle Communication
and Infotainment, Connected
Personal Smart Devices,
Video Surveillance

- Once connected to a network, your smart objects exchange information with other systems
- **Analytics Versus Control Applications**
- **Analytics application:**
- This type of application collects data from multiple smart objects, processes the collected data, and displays information resulting from the data that was processed
- The important aspect is that the application processes the data to convey a view of the network that cannot be obtained from solely looking at the information displayed by a single smart object.

- **Control application:**
- This type of application controls the behavior of the smart object or the behavior of an object related to the smart object.
- For example, a pressure sensor may be connected to a pump. A control application increases the pump speed when the connected sensor detects a drop in pressure.
- Control applications are very useful for controlling complex aspects of an IoT network with a logic that cannot be programmed inside a single IoT object

- **Data Versus Network Analytics**
- **Data analytics:**
- This type of analytics processes the data collected by smart objects and combines it to provide an intelligent view related to the IoT system.
- At a very basic level, a dashboard can display an alarm when a weight sensor detects that a shelf is empty in a store
- In a more complex case, temperature, pressure, wind, humidity, and light levels collected from thousands of sensors may be combined and then processed to determine the likelihood of a storm and its possible path

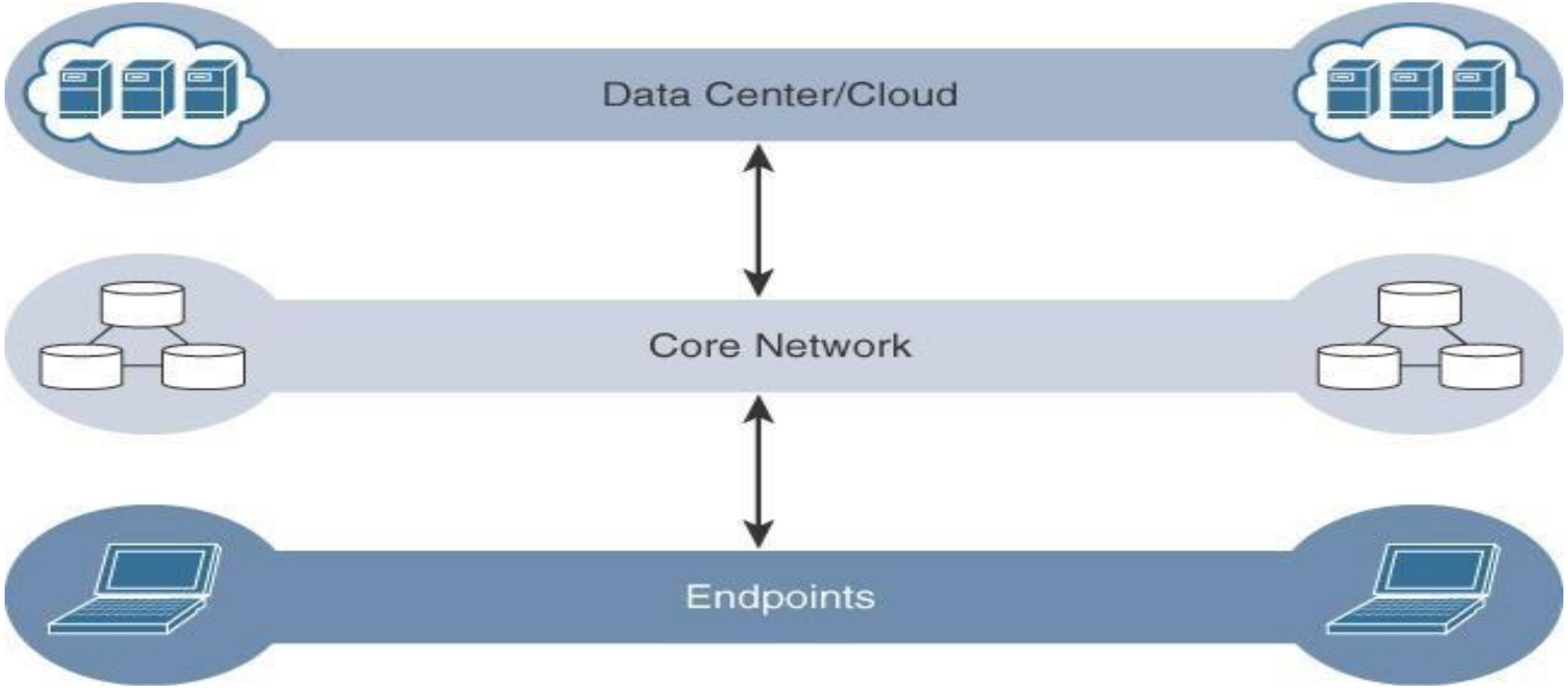
- **Network analytics:**
- Most IoT systems are built around smart objects connected to the network.
- A loss or degradation in connectivity is likely to affect the efficiency of the system. Such a loss can have dramatic effects.

IOT DATA MANAGEMENT AND COMPUTE STACK

- The data generated by IoT sensors is one of the single biggest challenges in building an IoT system
- In sensor networks, the vast majority of data generated is unstructured and of very little use on its own
- In most cases, the processing location is outside the smart object.
- A natural location for this processing activity is the cloud.
- Smart objects need to connect to the cloud, and data processing is centralized
- One advantage of this model is simplicity

- Limitations
- As data volume, the variety of objects connecting to the network, and the need for more efficiency increase, new requirements appear, and those requirements tend to bring the need for data analysis closer to the IoT system
- Minimizing latency
- Conserving network bandwidth
- Increasing local efficiency

Data management in traditional IT systems



IoT issues to be addressed

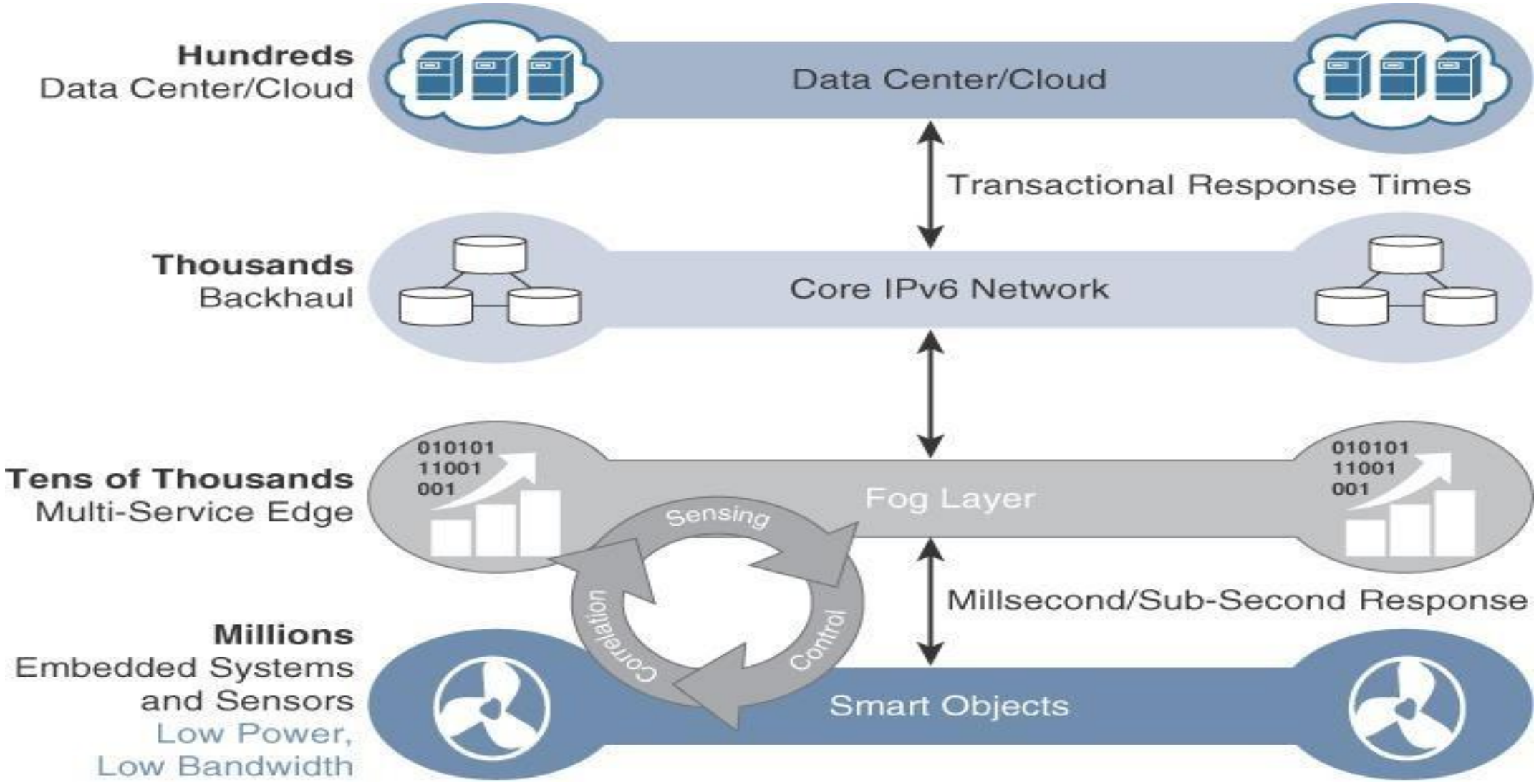
- Bandwidth in last-mile IoT networks is very limited
- Latency can be very high
- Network backhaul from the gateway can be unreliable and often depends on 3G/LTE or even satellite links
- The volume of data transmitted over the backhaul can be high
- Big data is getting bigger

Fog Computing

- The solution to the challenges mentioned in the previous section is to distribute data management throughout the IoT system, as close to the edge of the IP network as possible
- Any device with computing, storage, and network connectivity can be a fog node.
- Examples include industrial controllers, switches, routers, embedded servers, and IoT gateways.
- Analyzing IoT data close to where it is collected minimizes latency, offloads gigabytes of network traffic from the core network, and keeps sensitive data inside the local network.

- An advantage of this structure is that the fog node allows intelligence gathering (such as analytics) and control from the closest possible point, and in doing so, it allows better performance over constrained networks

The IoT Data Management and Compute Stack with Fog Computing



- The defining characteristic of fog computing are as follows:
- **Contextual location awareness and low latency**
- **Geographic distribution**
- **Deployment near IoT endpoints**
- **Wireless communication between the fog and the IoT endpoint**
- **Use for real-time interactions**

Edge Computing

- Computing resides directly in the sensors and IoT devices.
- New classes of IoT endpoints have enough compute capabilities to perform at least low-level analytics and filtering to make basic decisions

The Hierarchy of Edge, Fog, and Cloud

- It is important to stress that edge or fog computing in no way replaces the cloud
- This model suggests a hierarchical organization of network, compute, and data storage resources.
- At each stage, data is collected, analyzed, and responded to when necessary, according to the capabilities of the resources at each layer.
- As data needs to be sent to the cloud, the latency becomes higher.
- The advantage of this hierarchy is that a response to events from resources close to the end device is fast and can result in immediate benefits, while still having deeper compute resources available in the cloud when necessary.

- It is important to note that the heterogeneity of IoT devices also means a heterogeneity of edge and fog computing resources.
- While cloud resources are expected to be homogenous, it is fair to expect that in many cases both edge and fog resources will use different operating systems, have different CPU and data storage capabilities, and have different energy consumption profiles.

- The most time-sensitive data is analyzed on the edge or fog node closest to the things generating the data.
- Data that can wait seconds or minutes for action is passed along to an aggregation node for analysis and action.
- Data that is less time sensitive is sent to the cloud for historical analysis, big data analytics, and long-term storage

- In summary, when architecting an IoT network, you should consider the amount of data to be analyzed and the time sensitivity of this data.
- Understanding these factors will help you decide whether cloud computing is enough or whether edge or fog computing would improve your system efficiency.
- Fog computing accelerates awareness and response to events by eliminating a round trip to the cloud for analysis
- It avoids the need for costly bandwidth additions by offloading gigabytes of network traffic from the core network.
- It also protects sensitive IoT data by analyzing it inside company walls.