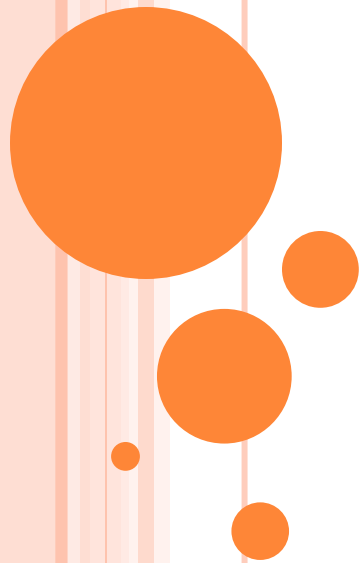


CHAPTER-2

IoT Network Architecture and Design



THIS CHAPTER EXPLORES THE FOLLOWING AREAS

- **Drivers Behind New Network Architectures**
- **Comparing IoT Architectures**
- **A Simplified IoT Architecture**
- **The Core IoT Functional Stack**
- **IoT Data Management and Compute Stack**



DRIVERS BEHIND NEW NETWORK ARCHITECTURES

- To implement any networking concepts designing and understanding the **network architecture** is of utmost importance.
- 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.**
- While **IT systems** are mostly concerned with **reliable** and continuous support of business applications such as email, web, databases 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.



IOT ARCHITECTURAL DRIVERS

- Scale
- Security
- Constrained Devices and Networks
- Data
- Legacy Device Support



i. Scale

- The scale of a typical IT network is on the order of several thousand devices—typically printers, mobile wireless devices, laptops, servers, and so on.
- But when a scale of a network goes from a few thousand endpoints to a few millions the IT engineers lack a required skills to design a network that is intended to support millions of routable IP endpoints.



ii. Security

- The frequency and impact of cyber attacks in recent years has increased dramatically. Protecting corporate data from intrusion and theft is one of the main functions of the IT department.
- IoT systems require consistent mechanisms of authentication, encryption, and intrusion prevention techniques that understand the behaviour of industrial protocols and can respond to attacks on critical infrastructure



- For optimum security, IoT systems must:
 - Be able to identify and authenticate all entities involved in the IoT service(i.e., 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



iii. Constrained Devices and Networks

- Most IoT sensors are designed for a single job, and they are typically small and inexpensive. This means they often have limited power, CPU, and memory, and they transmit only when there is something important.
- If an IT network has performance constraints, the solution is simple: **Upgrade to a faster network.**
- If too many devices are on one VLAN and are impacting performance, we can simply carve out a new VLAN and continue to scale as much as we need.



iv. Data

- IoT devices generate a mountain of data. In general, most IT shops don't really care much about the unstructured chatty data generated by devices on the network.
- However, in IoT the data is like gold, as it is what enables businesses to deliver new IoT services that enhance the customer experience, reduce cost, and deliver new revenue opportunities




- Although most IoT-generated data is unstructured, the insights it provides through analytics can revolutionize processes and create new business models.

- For ex :
 - Imagine a smart city with a few hundred thousand smart streetlights, all connected through an IoT network.

- However, when all this data is combined, it can become difficult to manage and analyze effectively.



v. 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, we 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.
- 

- In many cases, legacy devices are so old that they don't even support IP.

- For ex :
 - A factory may replace machines only once every 20 years—or perhaps even longer!

 - It does not want to upgrade multi-million-dollar machines just so it can connect them to a network for better visibility and control.

 - However, many of these legacy machines might support older protocols, such as serial interfaces, and use RS-232.

 - In this case, the IoT network must either be capable of some type of protocol translation or use a gateway device to connect these legacy endpoints to the IoT network

COMPARING IOT ARCHITECTURES

- In the past several years, architectural standards and frameworks have emerged to address the challenge of designing massive-scale IoT networks.
- The foundational concept in all these architectures is supporting data, process, and the functions that endpoint devices perform.
- Two of the best-known architectures
- **oneM2M**
- **IoT World Forum (IoTWF)**



THE ONEM2M IoT STANDARDIZED ARCHITECTURE

- **European Telecommunications Standards Institute (ETSI)** created the M2M Technical Committee in 2008.
- **The goal of this committee** was to
 - create a common architecture that would help accelerate the adoption of M2M applications and devices.



- In 2012 launched oneM2M as a global initiative designed to promote efficient M2M communication systems and IoT.
- create a **common services layer**, which can be readily embedded in field devices to allow communication with application servers.
- One of the greatest **challenges** in designing an IoT architecture is dealing with the heterogeneity of devices, software, and access methods.



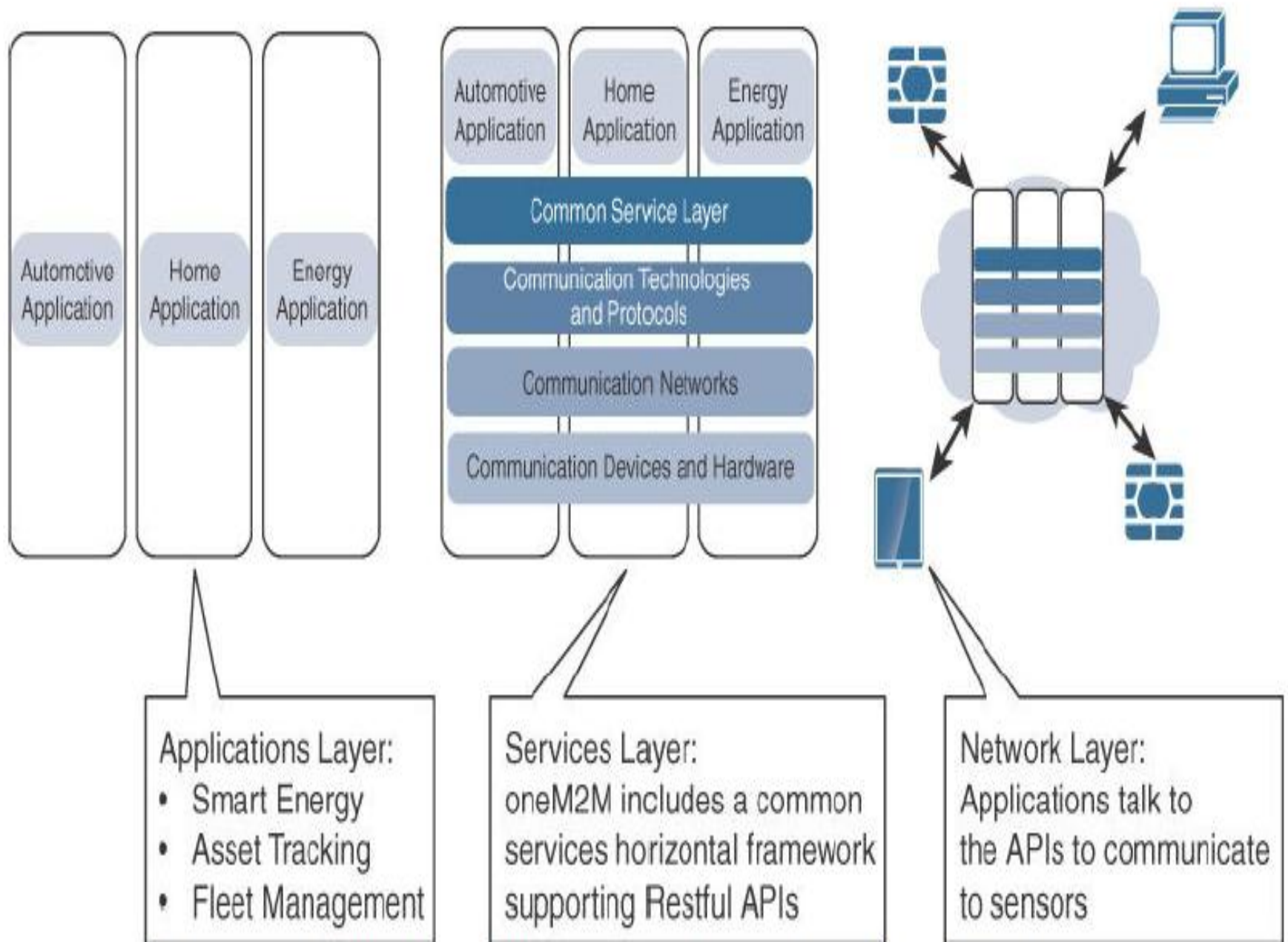


Figure 2-1 *The Main Elements of the oneM2M IoT Architecture*

- The oneM2M architecture divides IoT functions into three major domains: the **application layer**, the **services layer**, and the **network layer**.
- Let's examine each of these domains in turn:

- i. Applications 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 API definitions for interaction with business intelligence (BI) systems.




ii. 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.



iii. 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.
 - the communications infrastructure include wireless mesh technologies, such as IEEE 802.15.4, and wireless point-to-multipoint systems, such as IEEE 801.11
 - Also included are wired device connections, such as IEEE 1901 power line communications.
- 

THE IOT WORLD FORUM (IOTWF) STANDARDIZED ARCHITECTURE

- In 2014 the IoTWF architectural committee (led by Cisco, IBM, Rockwell Automation, and others) published a seven-layer IoT architectural reference model.
- visualizing IoT from a technical perspective.
- Each of the seven layers is broken down into specific functions, and security encompasses the entire model.
- Figure 2.2 details the IoT reference model published by the IoTWF



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)

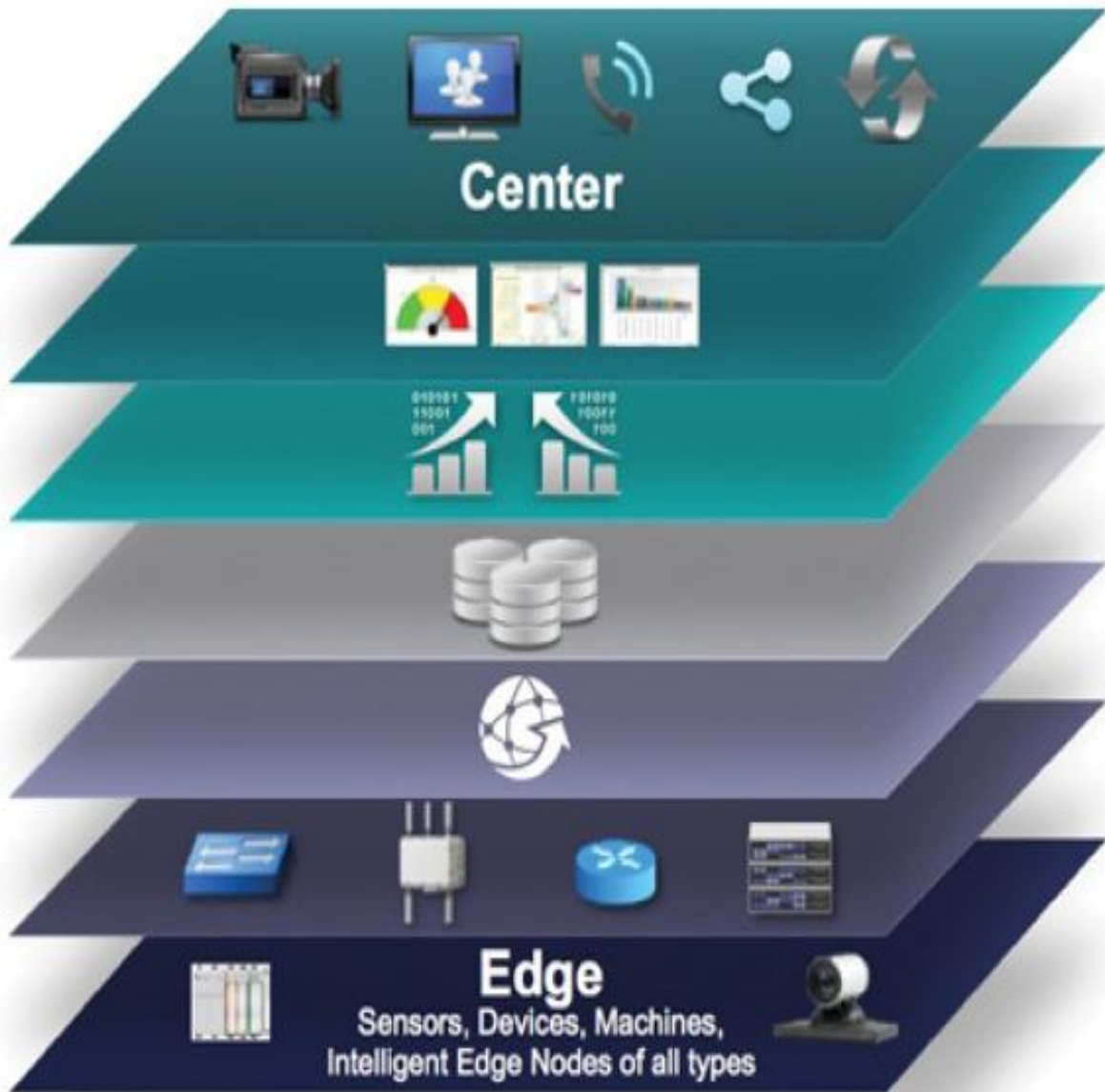



Figure 2.1 : IoT Reference Model Published by the IoT World Forum

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



- **Layer 1: Physical Devices and Controllers Layer**
 - This layer is home to the “things” in the Internet of Things
 - including 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.**
- 

- **Layer 2: Connectivity Layer**
- the focus is on connectivity.
- The most important **function** of this IoT layer is the **reliable and timely transmission of data.**

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



Figure 2-3 *IoT Reference Model Connectivity Layer Functions*

○ Layer 3: Edge Computing

- Edge computing is often referred to as the “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



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

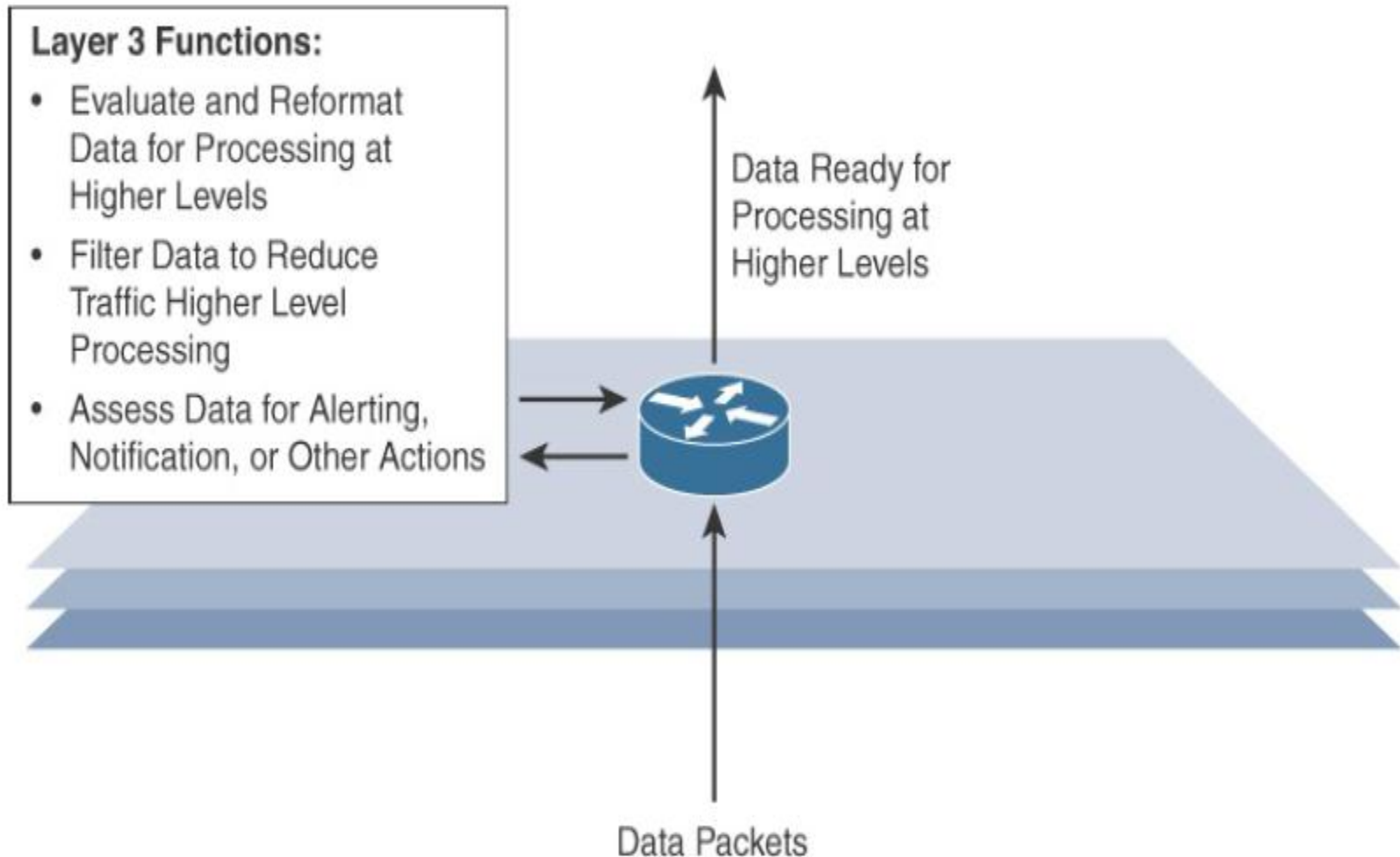


Figure 2-4 *IoT Reference Model Layer 3 Functions*

○ Upper Layers: Layers 4–7

- 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 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 visualization.
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. Collaboration on and communicating IoT information often requires multiple steps and it is what makes IoT useful. This layer can change business processes and delivers benefits of IoT

IT and OT Responsibilities in the IoT Reference Model

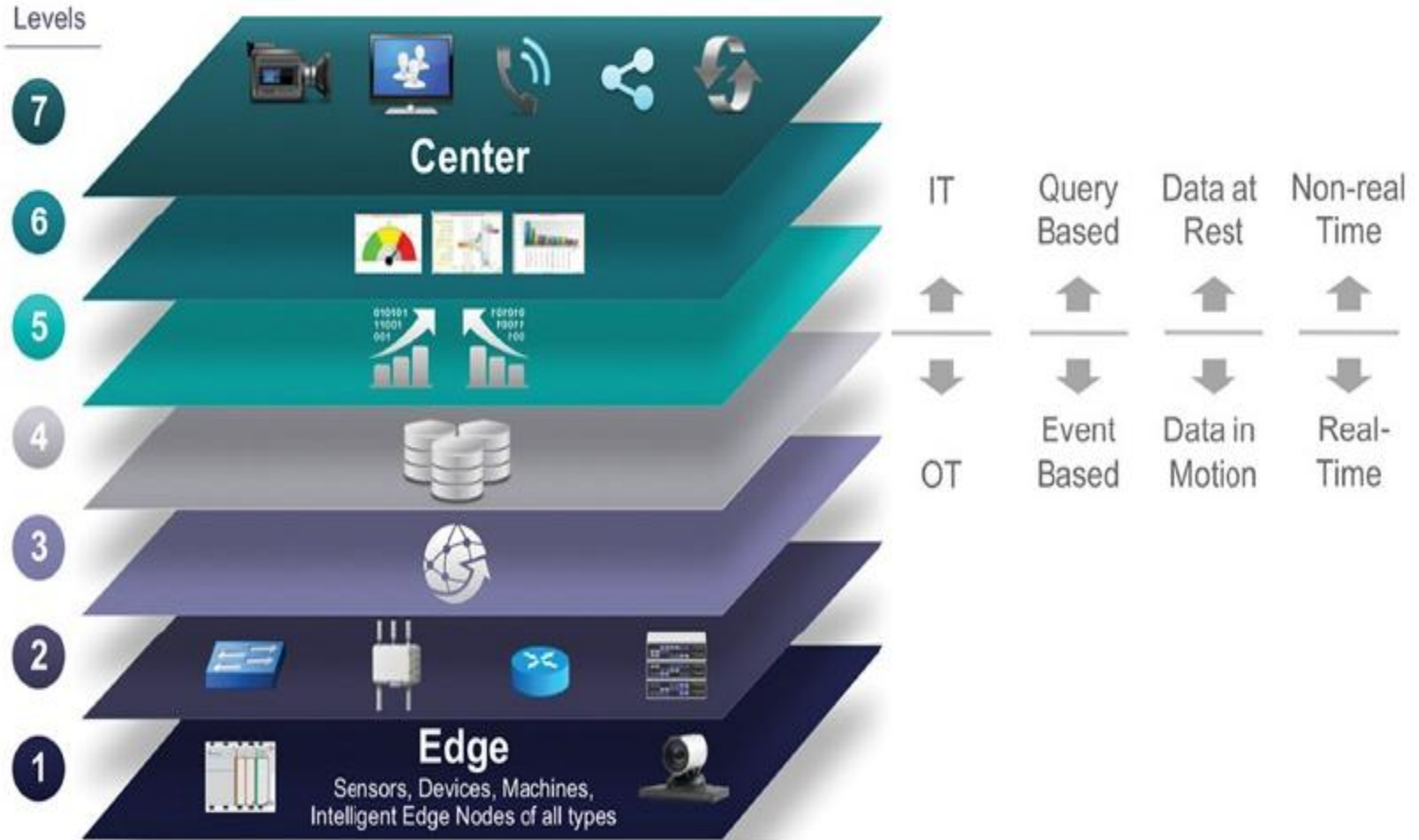


Figure 2-5 IoT Reference Model Separation of IT and OT

- The bottom of the stack is generally in the domain of OT.
- For an industry like oil and gas, this includes sensors and devices connected to pipelines, oil rigs, refinery machinery, and so on.
- The top of the stack is in the IT area and includes things like the servers, databases, and applications, all of which run on a part of the network controlled by IT.



- At the bottom, in the OT layers, the devices generate real-time data at their own rate
- data has to be buffered or stored at certain points within the IoT stack.
- Layering data management in this way throughout the stack helps the top four layers handle data at their own speed.



- the real-time “**data in motion**” close to the edge has to be organized and stored so that it becomes “**data at rest**” for the applications in the IT tiers.
- The IT and OT organizations need to work together for overall data management.



ADDITIONAL IoT REFERENCE MODELS

1. Purdue model for control Hierarchy
2. Industrial Internet Reference Architecture(IIRA) by Industrial Internet Consortium(IIC)
3. Internet of Things-Architecture(IoT-A)



A SIMPLIFIED IoT ARCHITECTURE

- we can describe an IoT framework that highlights the fundamental building blocks that are common to most IoT systems and which is intended to help you in designing an IoT network.
- This framework is presented as two parallel stacks:
 - i. **The IoT Data Management and Compute Stack**
 - ii. **The Core IoT Functional Stack**



A SIMPLIFIED IoT ARCHITECTURE

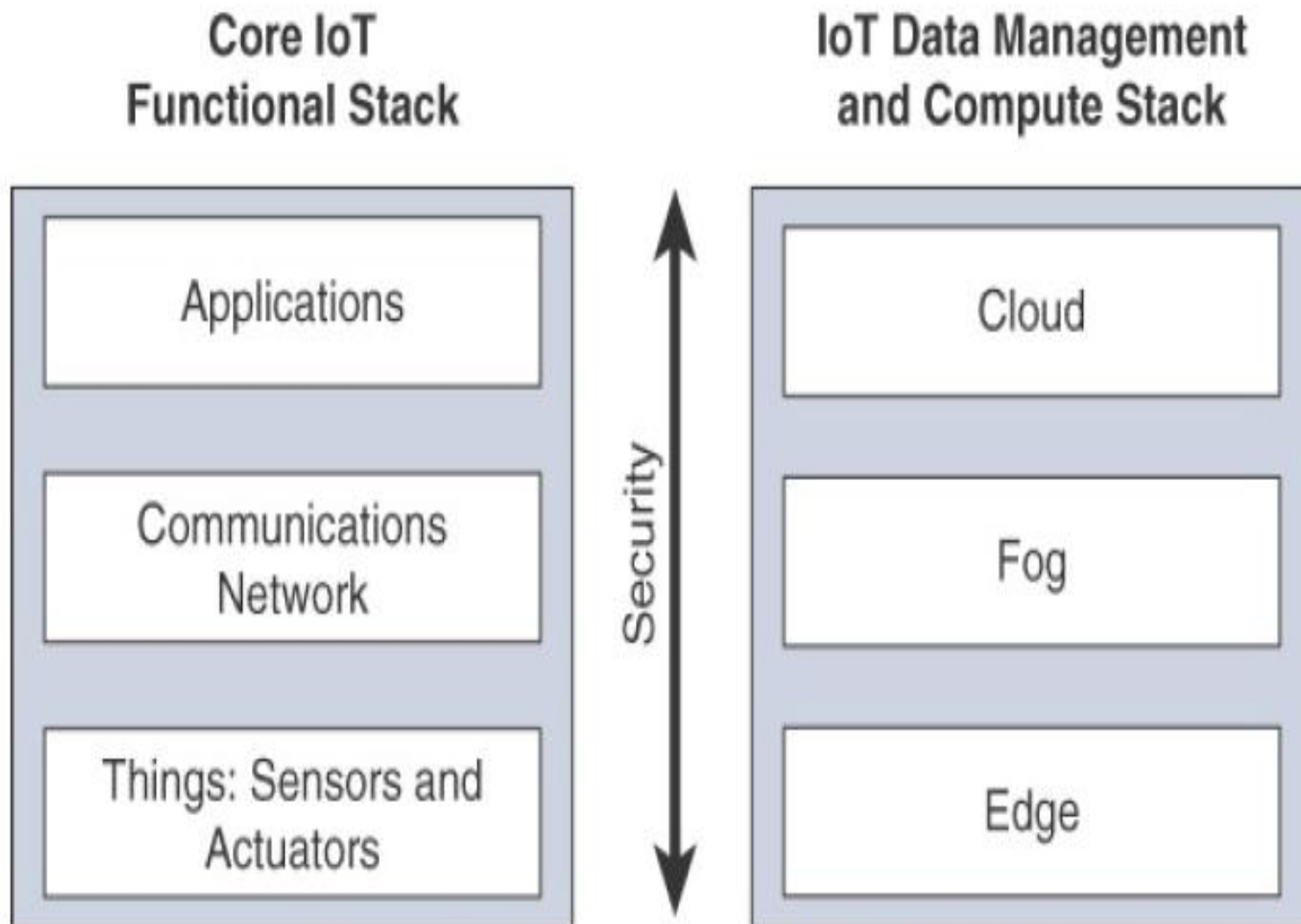


Figure 2-6 *Simplified IoT Architecture*

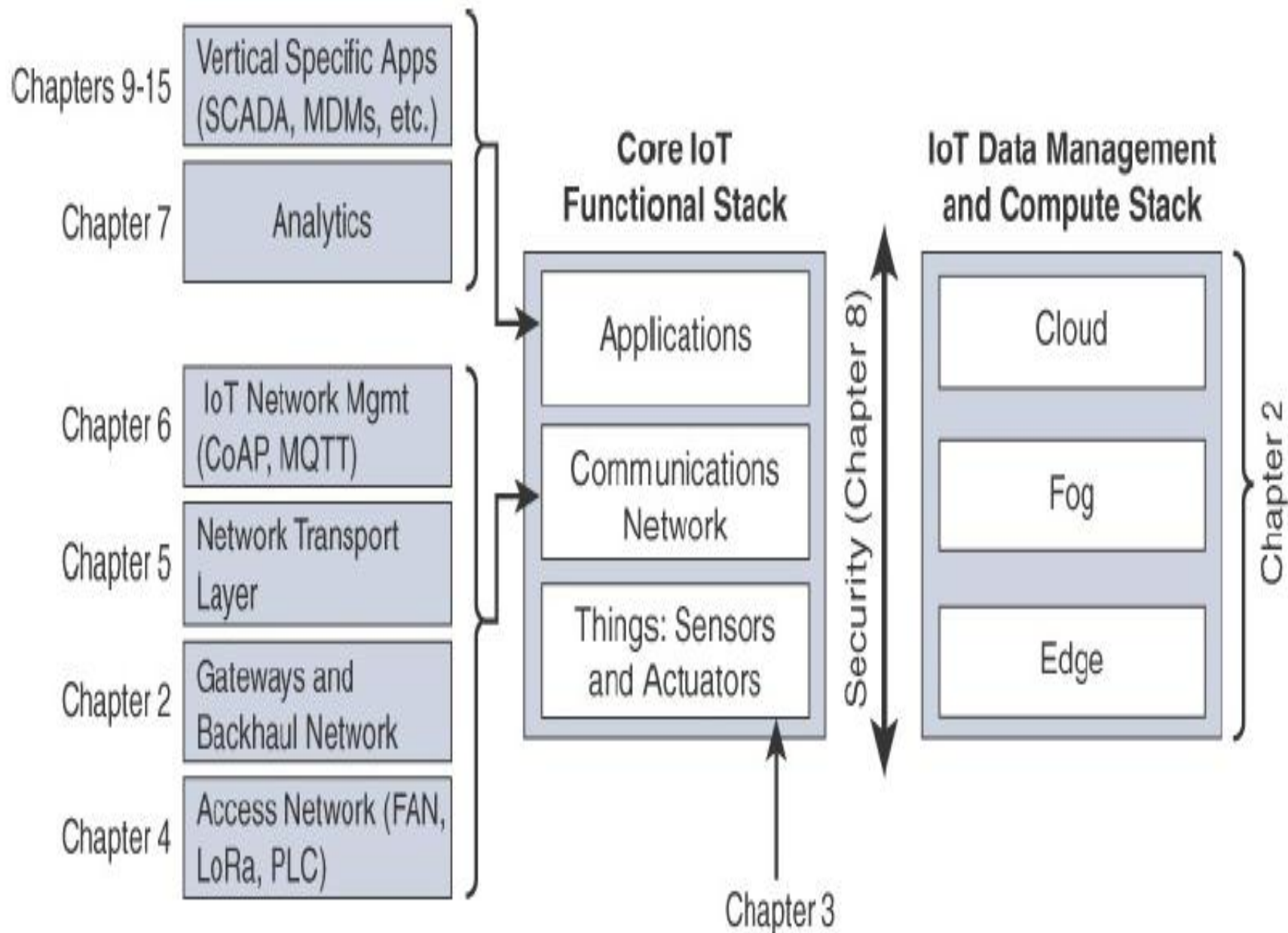


Figure 2-7 Expanded View of the Simplified IoT Architecture

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.
- “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**
- **Communications network layer**
 - **Access network sublayer**
 - **Gateways and backhaul network sublayer**
 - **Network transport sublayer**
 - **IoT network management sublayer**
- **Application and analytics layer**



LAYER 1: THINGS: SENSORS AND ACTUATORS LAYER

- Most IoT networks start from the object, or “thing,” that needs to be connected.
- From an architectural standpoint, the variety of smart object types, shapes, and needs drive the variety of IoT protocols and architectures.



- 1. Battery-powered or power-connected**
- 2. Mobile or static**
- 3. Low or high reporting frequency**
- 4. Simple or rich data**
- 5. Report range**
- 6. Object density per cell**



○ **Battery-powered or power-connected**

- This classification is based on whether the object carries its own energy supply or receives continuous power from an external power source.
- Battery-powered things can be moved more easily than line-powered objects.
- Batteries limit the lifetime and amount of energy that the object is allowed to consume, thus driving transmission range and frequency



○ **Mobile or static:**

- This classification is based on whether the “thing” should move or always stay at the same location.
- A sensor may be mobile because it is moved from one object to another or because it is attached to a moving object
- The frequency of the movement may also vary, from occasional to permanent.



○ **Low or high reporting frequency**

- This classification is based on how often the object should report monitored parameters.
- A rust sensor may report values once a month.
- A motion sensor may report acceleration several hundred times per second.
- Higher frequencies drive higher energy consumption



○ Simple or rich data

- This classification is based on the quantity of data exchanged at each report cycle.
- A humidity sensor in a field may report a simple daily index and while an engine
- sensor may report hundreds of parameters, from temperature to pressure, gas velocity etc.
- Richer data typically drives higher power consumption.



○ **Report range**

- This classification is based on the distance at which the gateway is located.

○ **Object density per cell**

- This classification is based on the number of smart objects over a given area, connected to the same gateway



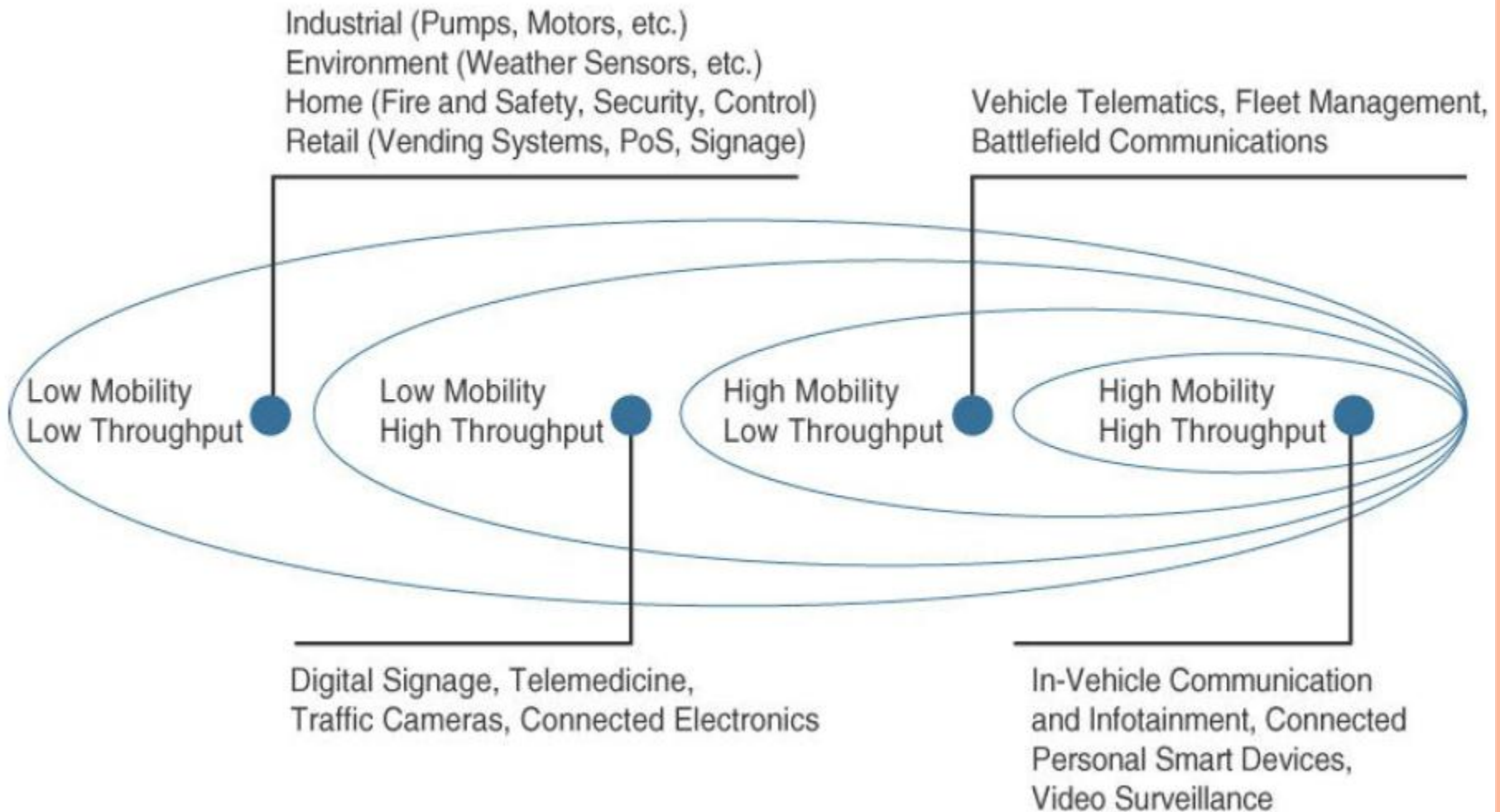
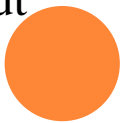


Figure 2.8: Example of Sensor Applications Based on Mobility and Throughput



LAYER 2: 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:
 - i. Access Network Sublayer
 - ii. Gateways and backhaul network sublayer
 - iii. Network transport sublayer
 - iv. IoT network management sublayer




I. ACCESS NETWORK SUBLAYER

- There is a direct relationship between the IoT network technology we choose and the type of connectivity topology this technology allows
- Each technology was designed with a certain number of use cases in mind
 - what to connect, where to connect, how much data to transport at what interval and over what distance.



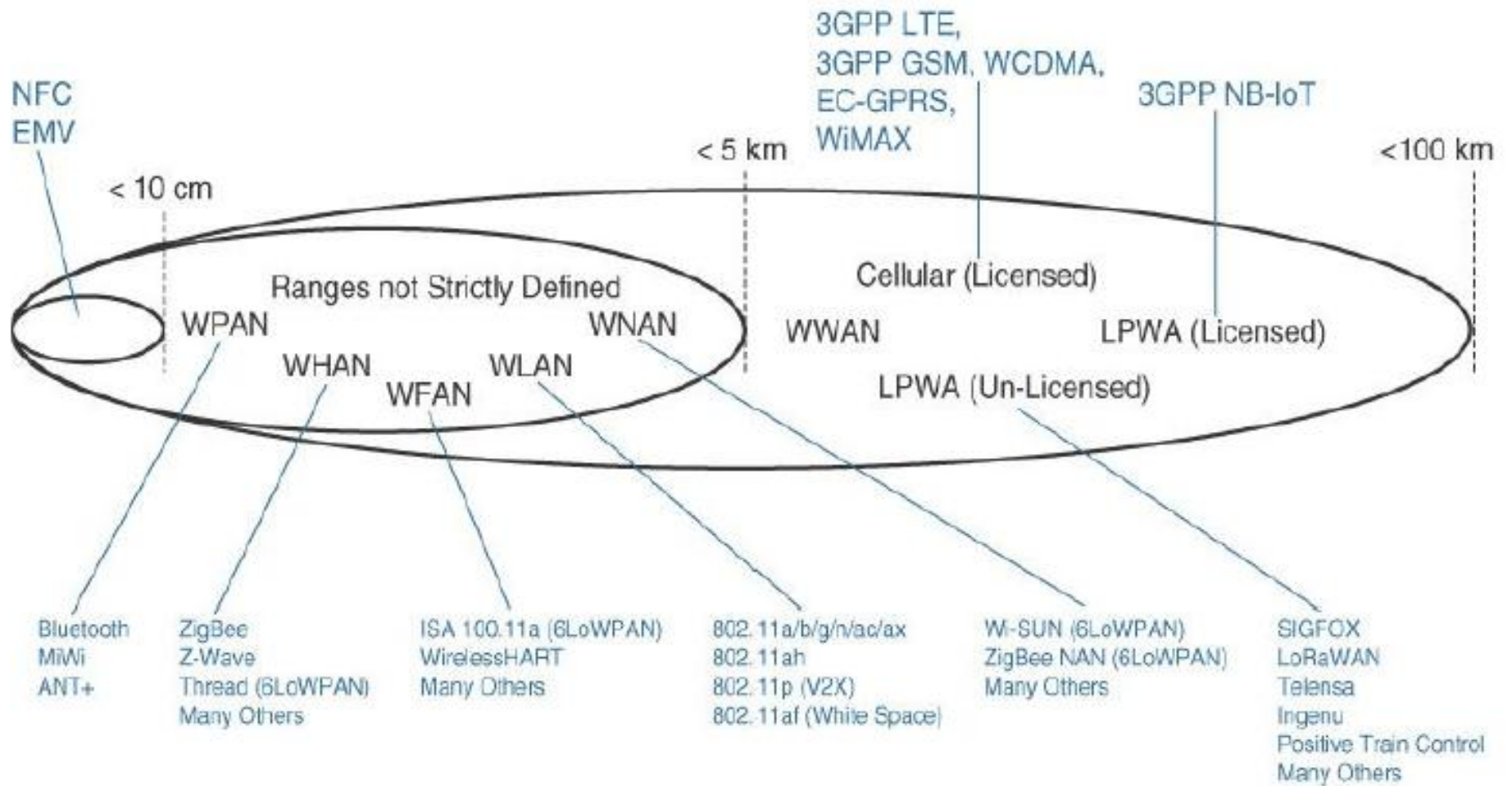
- As IoT continues to grow exponentially, we will encounter a wide variety of applications and special use cases.
- For each of them, an access technology will be required. IoT sometimes reuses existing access technologies whose characteristics match more or less closely the IoT use case requirements.
- One key parameter determining the choice of access technology is **the range between the smart object and the information collector.**



- Figure 2.9 lists some access technologies we may encounter in the IoT world and the expected transmission distances.
 - **Range estimates are grouped by category** names that illustrate the environment or the vertical where data collection over that range is expected.
 - Common groups are as follows:
 - **PAN (personal area network):** Scale of a few meters. This is the personal space around a person. A common wireless technology for this scale is **Bluetooth**.
 - **HAN (home area network):** Scale of a few tens of meters. At this scale, common wireless technologies for IoT include **ZigBee and Bluetooth Low Energy**
- 

- **NAN (neighbourhood area network):** Scale of a few hundreds of meters. The term NAN is often used to refer to a group of house units from which data is collected.
- **FAN (field area network):** Scale of several tens of meters to several hundred meters. FAN typically refers to an outdoor area larger than a single group of house units.
- **LAN (local area network):** Scale of up to 100 m. This term is very common in networking, and it is therefore also commonly used in the IoT space when standard networking technologies (such as Ethernet or IEEE 802.11) are used.





WPAN: Wireless Personal Area Network
 WHAN: Wireless Home Area Network
 WFAN: Wireless Field (or Factory) Area Network
 WLAN: Wireless Local Area Network

WNAN: Wireless Neighborhood Area Network
 WWAN: Wireless Wide Area Network
 LPWA: Low Power Wide Area

Figure 2.9 : Access Technologies and Distances

- Figure 2.10 demonstrates four technologies representing WHAN to WLAN ranges and compares the throughput and range that can be achieved in each case.

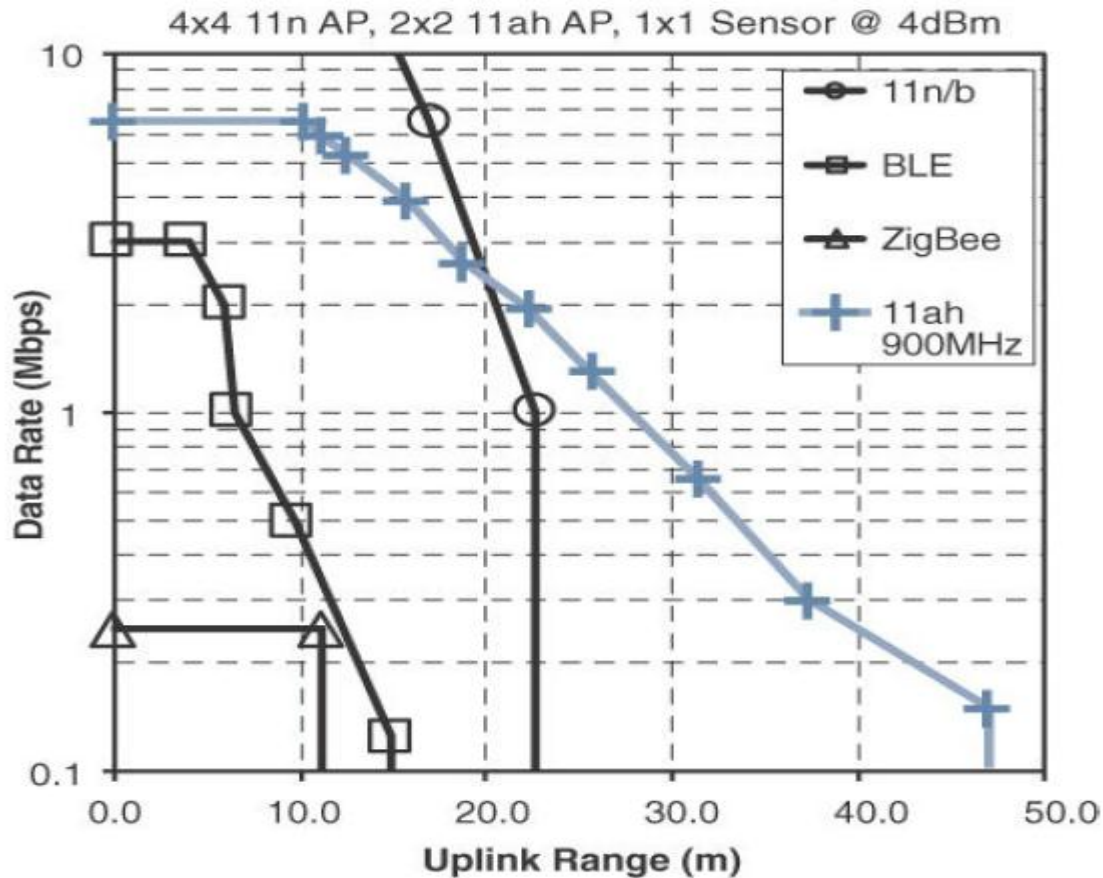


Figure 2.10 : Range Versus Throughput for Four WHAN to WLAN Technologies

- Figure 2.11 combines cost, range, power consumption, and typical available bandwidth for common IoT access technologies.

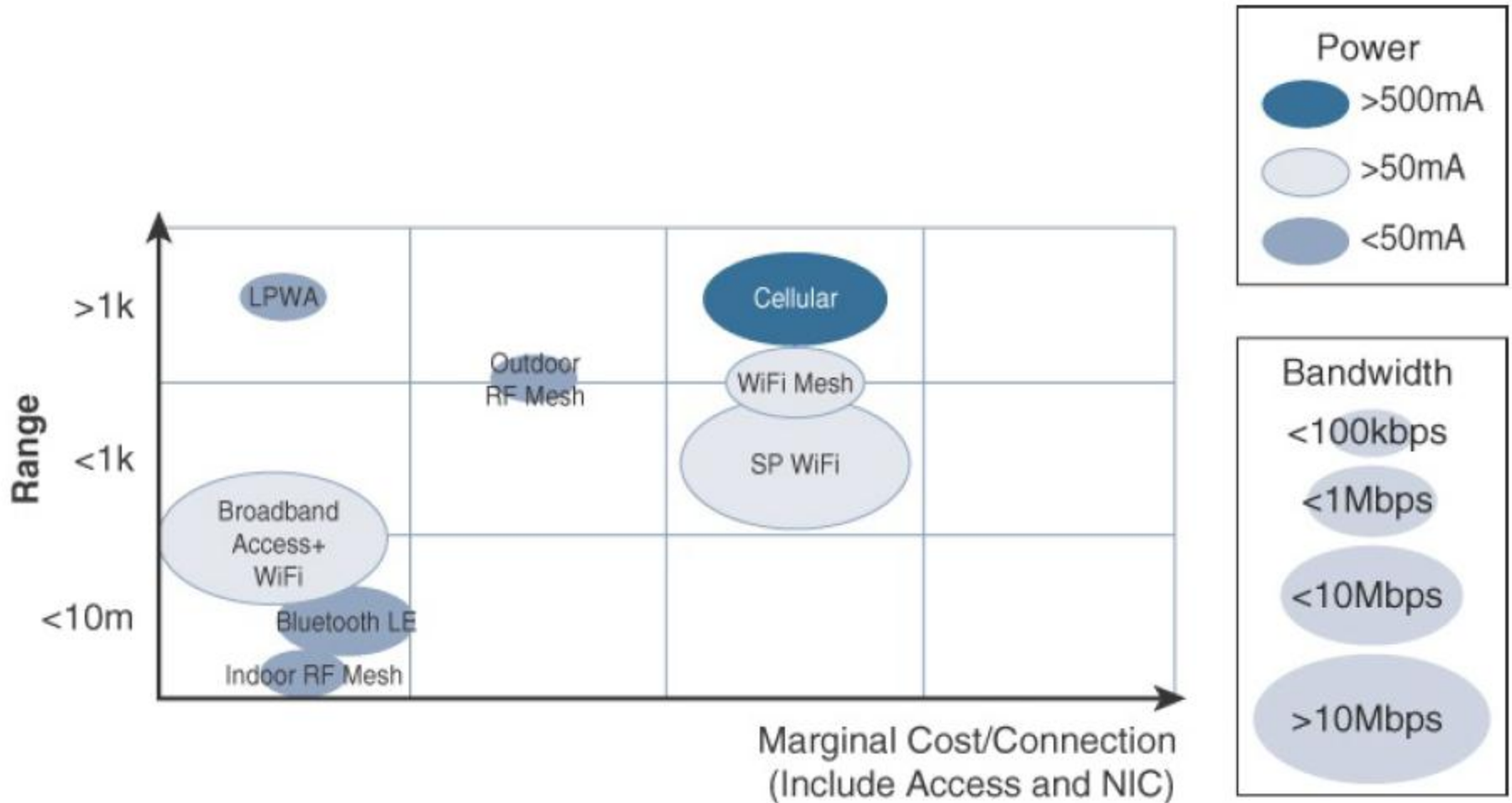


Figure 2.11 : Comparison Between Common Last-Mile Technologies in Terms of Range Versus Cost, Power, and Bandwidth

Communication topologies:

- Some technologies offer flexible connectivity structure to extend communication possibilities

1. Point-to-point topologies:

- These topologies allow one point to communicate with another point.
- several technologies are referred to as “point-to-point” when each object establishes an **individual session with the gateway**



2. Point-to-multipoint topologies:

- These topologies allow **one point to communicate with more than one other point.**
- Most IoT technologies where one or more than one gateways communicate with multiple smart objects are in this category



- To form a network, a device needs to connect with another device.
- When both devices fully implement the **protocol stack** functions,
 - they can form a peer-to peer network.
- The sensor which can implement a subset of protocol functions to perform just a specialized part (communication with the coordinator). Such a device is called a **reduced-function device (RFD)**.



- An RFD **cannot** be a coordinator. An RFD also cannot implement direct communications to another RFD.
- The coordinator that implements the full network functions is called, by contrast, a **full-function device** (FFD).

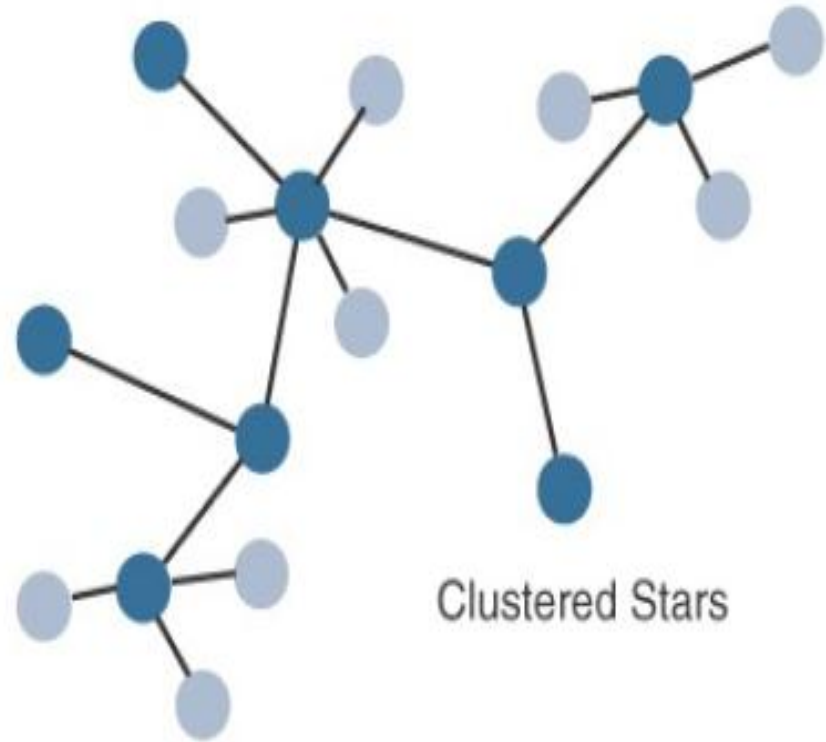


- An FFD can communicate directly with another FFD or with more than one FFD, forming multiple peer-to-peer connections.
- Topologies where each FFD has a unique path to another FFD are called **cluster tree topologies**. FFDs in the cluster tree may have RFDs, resulting in a **cluster star topology**.
- Figure 2.12 illustrates these topologies
- Other point-to-multipoint technologies allow a node to have more than one path to another node, forming a **mesh topology**.





Star Topology



Clustered Stars

- Full Function Device
- Reduced Function Device

Figure 2-12 *Star and Clustered Star Topologies*



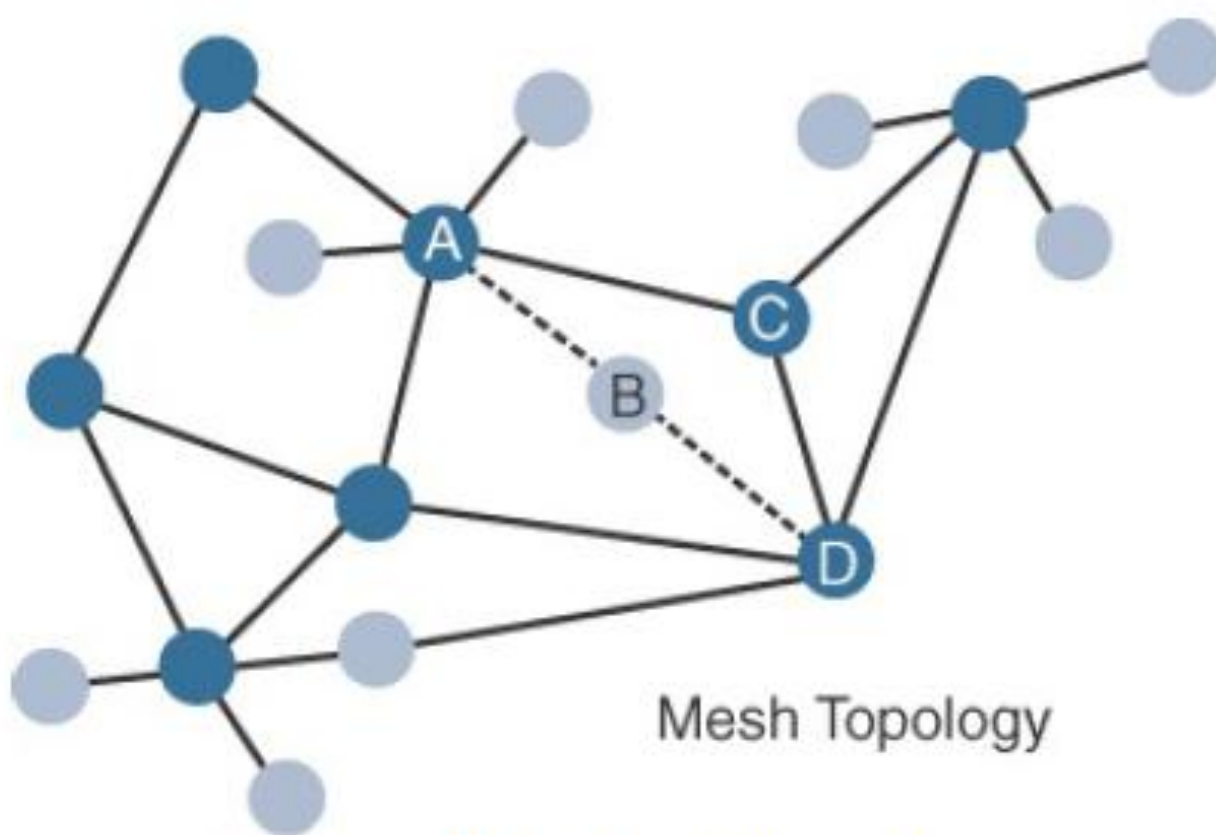


Figure 2-13 *Mesh Topology*


- Nodes A and D are too far apart to communicate directly. In this case, communication can be relayed through nodes B or C. Node B may be used as the primary relay



GATEWAYS AND BACKHAUL SUBLAYER

- **Data** collected from a **smart object** may need to be **forwarded** to a **central station** where data is processed.
- As this station is often in a **different location** from the smart object, data directly received from the sensor through an **access technology** needs to be forwarded to **another medium** (the backhaul) and transported to the **central station**.
- Ex: A dedicated short-range communication (DSRC) allows vehicle-to-vehicle and vehicle-to-infrastructure communication.



- The **gateway** is in charge of this inter-medium communication.
 - In most cases, the smart objects are static or mobile within a limited area.
 - The gateway is often static. However, some IoT technologies do not apply this model.
-
- When the smart object's operation is controlled from a local site, and when the environment is stable (for example, factory or oil and gas field), **Ethernet** can be used as a backhaul.
 - WiMAX (802.16) is an example of a longer-range technology. WiMAX can achieve ranges of up to 50 kilometers with rates of up to 70 Mbps
- 

Technology	Type and Range	Architectural Characteristics
Ethernet	Wired, 100 m max	Requires a cable per sensor/sensor group; adapted to static sensor position in a stable environment; range is limited; link is very reliable
Wi-Fi (2.4 GHz, 5 GHz)	Wireless, 100 m (multipoint) to a few kilometers (P2P)	Can connect multiple clients (typically fewer than 200) to a single AP; range is limited; adapted to cases where client power is not an issue (continuous power or client battery recharged easily); large bandwidth available, but interference from other systems likely; AP needs a cable
802.11ah (Halo W, Wi-Fi in sub-1 GHz)	Wireless, 1.5 km (multipoint), 10 km (P2P)	Can connect a large number of clients (up to 6000 per AP); longer range than traditional Wi-Fi; power efficient; limited bandwidth; low adoption; and cost may be an issue
WiMAX (802.16)	Wireless, several kilometers (last mile), up to 50 km (backhaul)	Can connect a large number of clients; large bandwidth available in licensed spectrum (fee-based); reduced bandwidth in license-free spectrum (interferences from other systems likely); adoption varies on location
Cellular (for example, LTE)	Wireless, several kilometers	Can connect a large number of clients; large bandwidth available; licensed spectrum (interference-free; license-based)

Table 2-4 *Architectural Considerations for WiMAX and Cellular Technologies*

NETWORK TRANSPORT SUBLAYER

- We know that a **hierarchical communication architecture** in which a series of smart objects report to a **gateway** that conveys the reported data over another **medium and up to a central station**.
- However, practical implementations are often flexible, with **multiple transversal communication paths**



○ communication structure involve

- peer-to-peer
- point-to-point
- point-to-multipoint (gateway or head-end)
- unicast and multicast communications (software update to one or multiple systems)



- **communication** occurs over **multiple media**
- *For ex: power lines inside our house or a short-range wireless system like indoor Wi-Fi and/or ZigBee*
- a **longer-range** wireless system to the **gateway**, and yet another **wireless** or **wired** medium for backhaul transmission.
- To allow for such **communication structure**, a *network protocol* with *specific characteristics* needs to be **implemented**.



- The **protocol** needs to be open and **standard-based** to accommodate **multiple industries** and **multiple media**.
- **Scalability** (to accommodate thousands or millions of sensors in a single network) and **security** are also common requirements.
- **IP is a protocol** that matches all these **requirements**.
- The flexibility of IP allows this protocol to be embedded in objects of very different natures, exchanging **information over very different media, including low-power, lossy, and low-bandwidth networks**.



- Finally, the **transport layer protocols built above IP** (UDP and TCP) can easily be leveraged to decide whether the network should **control the data packet delivery** (with TCP) or whether the **control task should be left to the application (UDP)**.



IoT NETWORK MANAGEMENT SUBLAYER

- **IP, TCP, and UDP** bring connectivity to IoT networks.
- **Upper-layer protocols** need to take care of **data transmission** between the **smart objects** and **other systems**.
- **Multiple protocols** have been **created to solve IoT data communication problems**.
- Some networks depends on a **push model** and some other networks depends on a **pull model**



- some **IoT implementers** have suggested **HTTP** which has a **client and server** component.
- But HTTP is something of a fat protocol and was **not designed to operate in *constrained environments*** with **low memory, low power, low bandwidth, and a high rate of packet failure.**




- Despite these limitations, **other web-derived** protocols have been suggested for the IoT space.
- One example is **WebSocket**.
 - WebSocket is part of the **HTML5 specification**, and provides a **simple bidirectional connection** over a **single connection**



- **WebSocket** is often combined with other protocols, such as **MQTT** to handle the **IoT-specific** part of the communication.
- With the same logic of reusing well-known methods, Extensible Messaging and Presence Protocol (**XMPP**) was created.
- XMPP is based on instant messaging and presence.
 - It allows the exchange of data between two or more systems and supports presence and contact list maintenance



- To respond to the **limits of web-based protocols**, another protocol was created by the IETF Constrained Restful Environments (CoRE) working group: **Constrained Application Protocol (CoAP)**.
- CoAP uses some methods similar to those of HTTP (such as Get, Post, Put, and Delete) **but implements a shorter list**, thus limiting the size of the header.
- CoAP also runs on **UDP** (whereas HTTP typically uses TCP). CoAP also adds a feature that is lacking in HTTP and very useful for IoT.
- Another common IoT protocol utilized in these middle to upper layers is **Message Queue Telemetry Transport (MQTT)**. 

- **MQTT uses a broker-based architecture.**
- The sensor can be set to be an MQTT publisher (publishes a piece of information),
- the application that needs to receive the information can be set as the MQTT subscriber, and
- any intermediary system can be set as a broker to relay the information between the publisher and the subscriber(s).

- **MQTT runs over TCP.** A consequence of the reliance on TCP is that an MQTT client typically holds a connection open to the broker at all times.



LAYER 3: APPLICATIONS AND ANALYTICS LAYER

- Once connected to a network, our smart objects exchange information with other systems.
- As soon as our IoT network spans more than a few sensors, the power of the Internet of Things appears in the applications that make use of the information exchanged with the smart objects
- From an architectural standpoint, basic classification can be as follows:
 - **Analytics Versus Control Applications**
 - **Data Versus Network Analytics**



- **Analytics Versus Control Applications**

- **Multiple applications** can help increase the **efficiency** of an IoT network.
- Each application collects data and provides a range of functions based on analysing the collected data.

- **Analytics Application**

- **Control Application**



○ 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 display can be about any aspect of the IoT network, from **historical reports, statistics, or trends to individual system states.**

○ Control Application

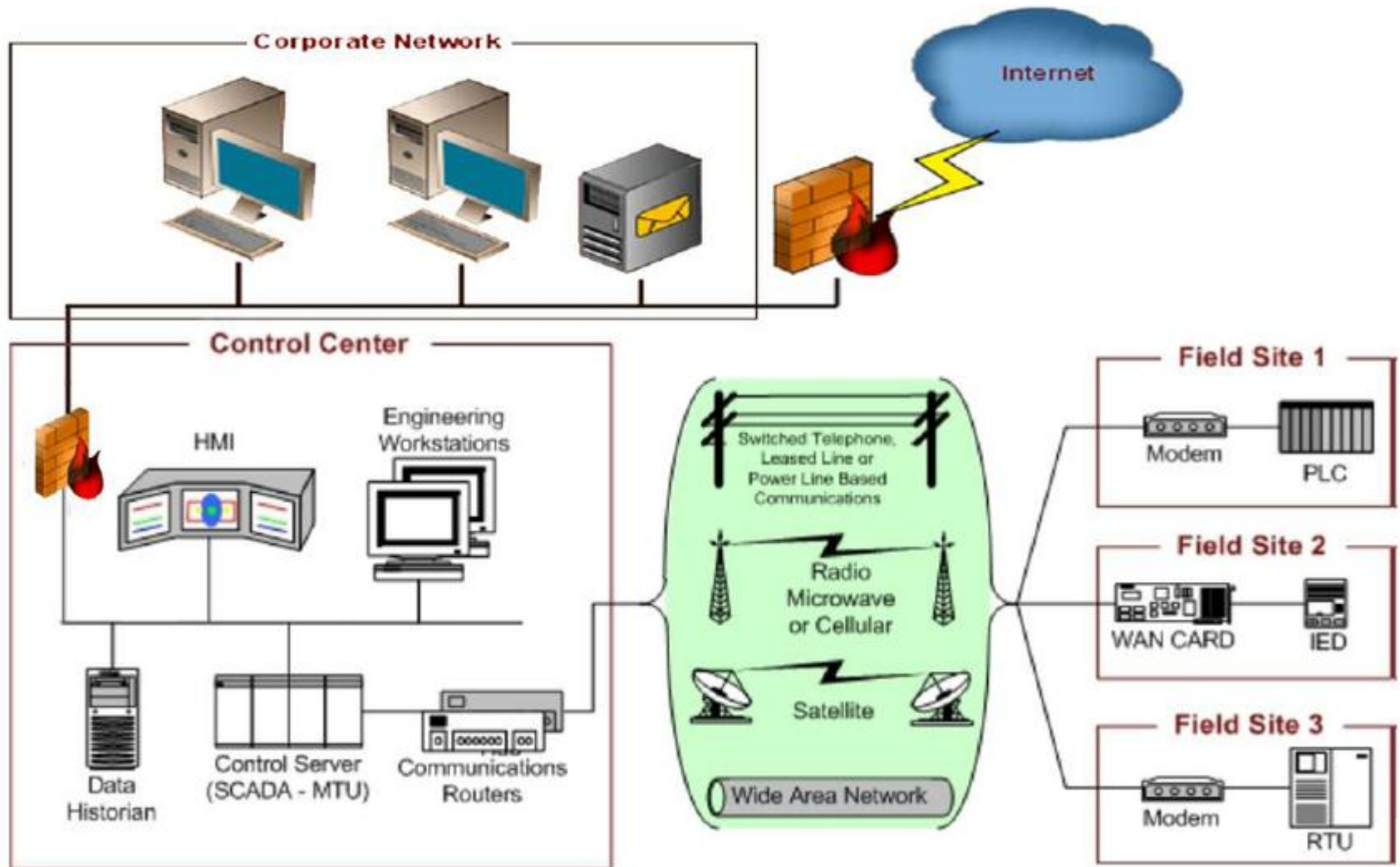
- This type of application **controls the behaviour** of the smart object or the **behaviour of an object** related to the smart object.
- For ex : a pressure sensor may be connected to a pump



- An example of control system architecture is SCADA.
- SCADA was developed as a universal method to **access remote systems and send instructions.**
- One example where SCADA is widely used is in the **control and monitoring of remote terminal units (RTUs)** on the electrical distribution grid.



SUPERVISORY CONTROL AND DATA ACQUISITION (SCADA)



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.

- **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 massive scale of IoT networks is fundamentally driving new architectures. As per the projections by Cisco nearly 50 billion devices will be connected to the IoT networks by the year 2020.
- In fact, the data generated by IoT sensors is one of the single biggest challenges in building an IoT system.



- In the case of modern IT networks, the data sourced by a computer or server is typically generated by the client/server communications model, and it serves the needs of the application.
- 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**. Objects just need to connect to a central cloud application.
- This model also has some limitations.
 - The data volume increases with more number of smart objects being connected to the network which in turn creates a new requirements.



- These new requirements include the following
 1. **Minimizing latency**
 2. **Conserving network bandwidth**
 3. **Increasing local efficiency**



- Data management in traditional IT systems is very simple.
- The endpoints (laptops, printers, IP phones, and so on) communicate over an IP core network to servers in the data center or cloud.
- Data is generally stored in the data center, and the physical links from access to core are typically high bandwidth, meaning access to IT data is quick.



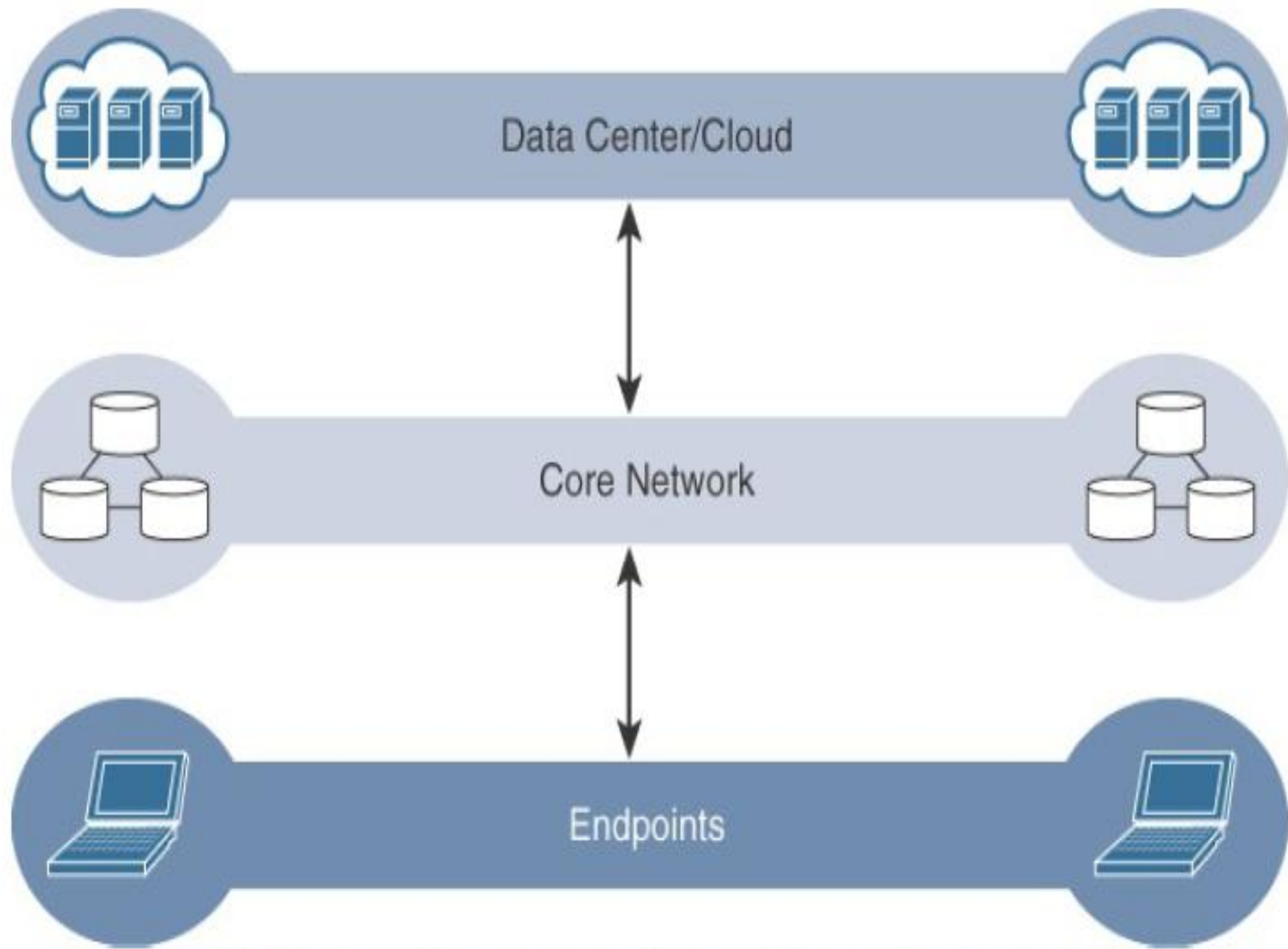



Figure 2-14 *The Traditional IT Cloud Computing Model*

IoT systems function differently. Several data-related problems need to be addressed:

1. Bandwidth in last-mile IoT networks is very limited.
 2. Latency can be very high.
 3. Network backhaul from the gateway can be unreliable and often depends on 3G/LTE or even satellite links
 4. The volume of data transmitted over the backhaul can be high, and much of the data may not really be that interesting(such as simple polling messages).
 5. Big data is getting bigger. The concept of storing and analyzing all sensor data in the cloud is impractical.
- 

○ Fog Computing

- The solution to the challenges encountered in data management is to **distribute data management throughout the IoT system**, as close to the edge of the IP network as possible.
- The best-known embodiment of edge services in IoT is **fog computing**.
- Any device with computing, storage, and network connectivity can be a **fog node**.
- Examples include industrial controllers, switches, routers, embedded servers, and IoT gateways



INDUSTRIAL IoT DATA PROCESSING LAYER STACK

CLOUD LAYER

Big Data Processing
Business Logic
Data Warehousing

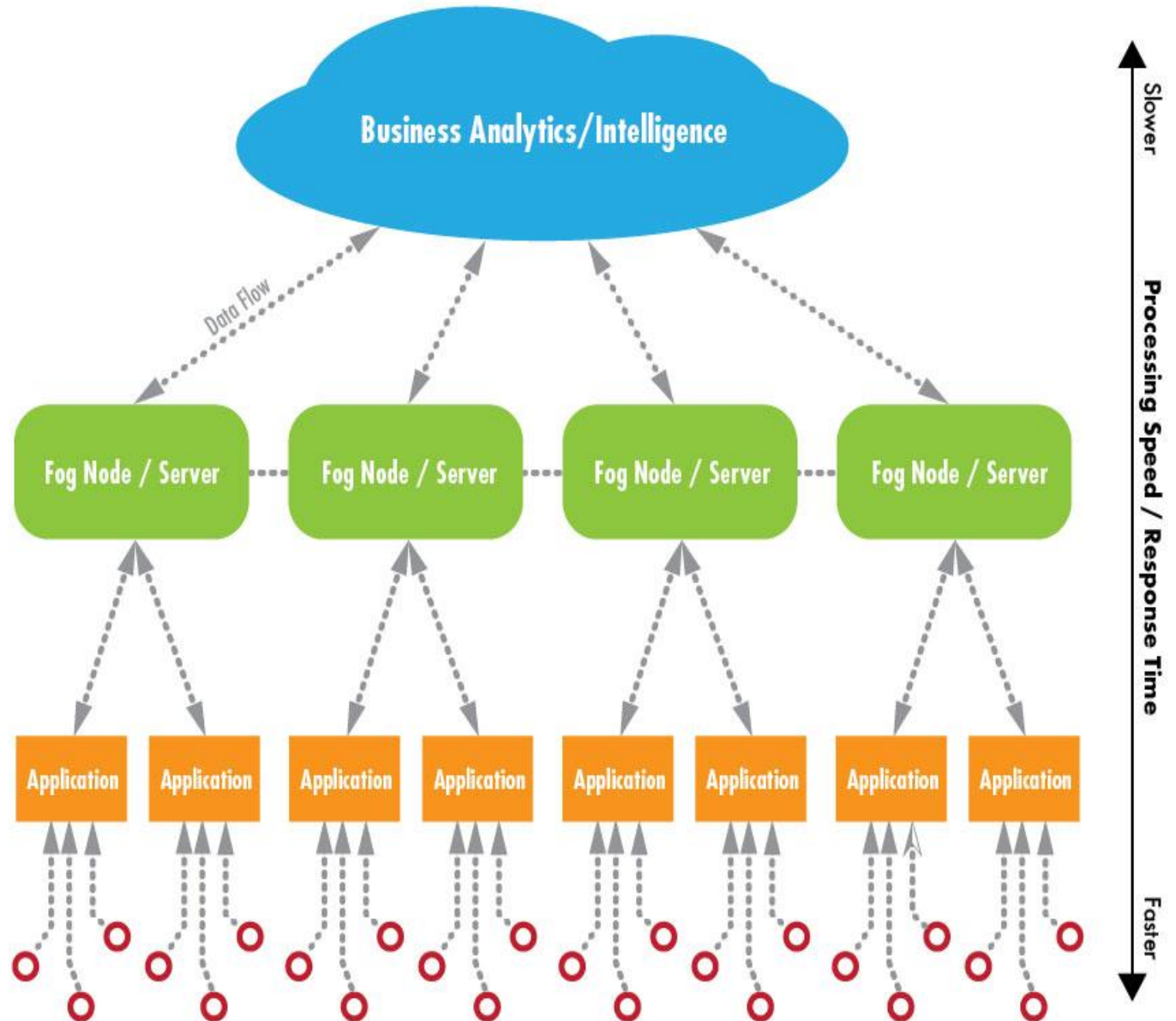
FOG LAYER

Local Network
Data Analysis & Reduction
Control Response
Virtualization/Standardization

EDGE LAYER

Large Volume Real-time Data Processing
At Source/On Premises Data Visualization
Industrial PCs
Embedded Systems
Gateways
Micro Data Storage

Sensors & Controllers (data origination)



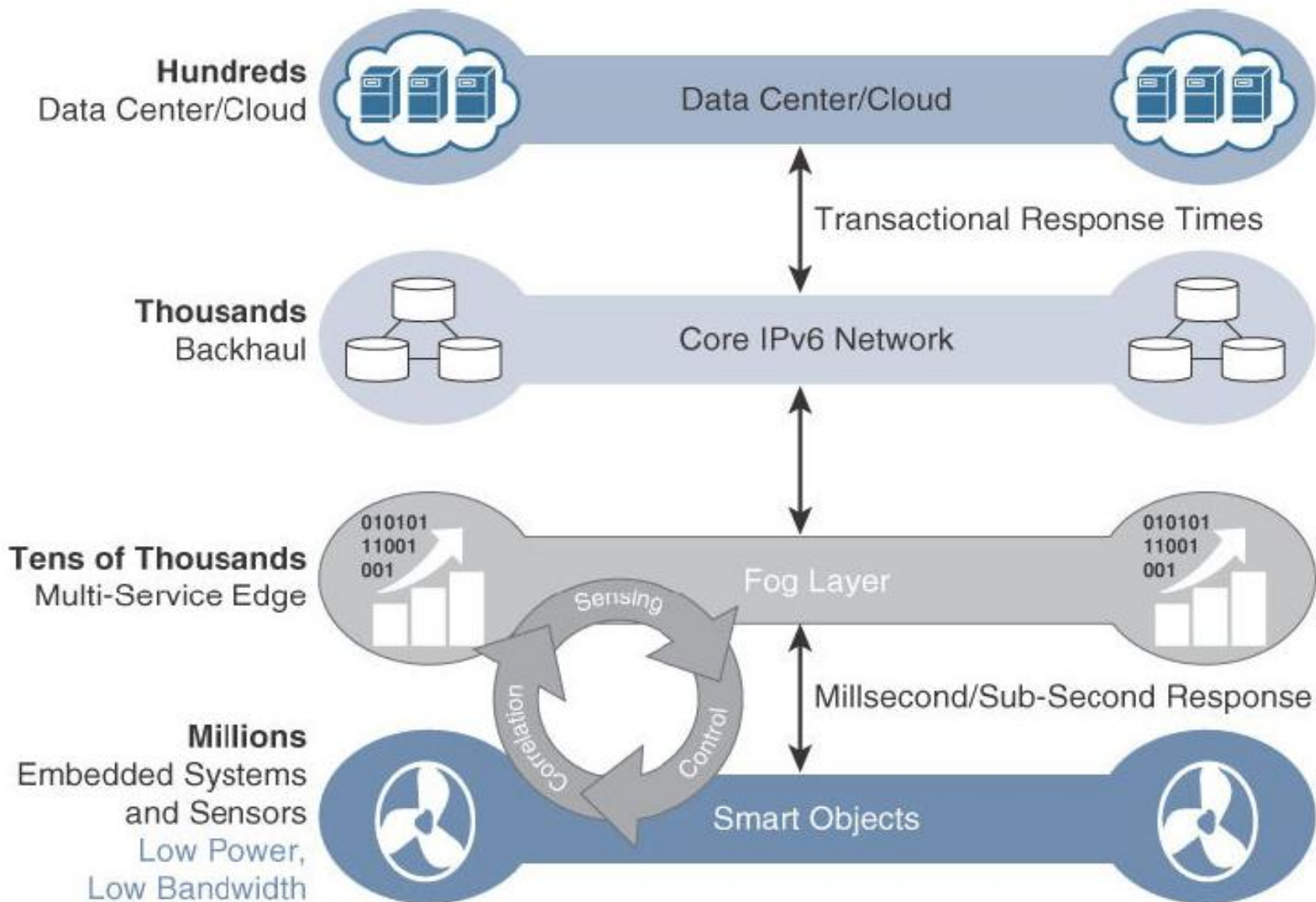


Figure 2-15 *The IoT Data Management and Compute Stack with Fog Computing*

○ The characteristic of fog computing are as follows:

- 1. Contextual location awareness and low latency**
- 2. Geographic distribution**
- 3. Deployment near IoT endpoints**
- 4. Wireless communication between the fog and the IoT endpoint**
- 5. Use for real-time interactions**



- **Contextual location awareness and low latency:**
 - The fog node sits as close to the IoT endpoint as possible to deliver distributed computing.

- **Geographic distribution:**
 - In sharp contrast to the more centralized cloud, the services and applications targeted by the fog nodes demand widely distributed deployments.

- **Deployment near IoT endpoints:**
 - Fog nodes are typically deployed in the presence of a large number of IoT endpoints.



➤ **Wireless communication between the fog and the IoT endpoint:**

- Although it is possible to connect wired nodes, the advantages of fog are greatest when dealing with a large number of endpoints, and wireless access is the easiest way to achieve such scale.

➤ **Use for real-time interactions:**

- Important fog applications involve real-time interactions rather than batch processing.
- Pre-processing of data in the fog nodes allows upper-layer applications to perform batch processing on a subset of the data.



EDGE COMPUTING

- **Edge computing** as “a part of a distributed **computing** topology in which information processing is located close to the **edge** – where things and people produce or consume that information.”
- Edge compute–capable meters are able to communicate with each other to share information on small subsets.
- Edge computing is also sometimes called “mist” computing.



THE HIERARCHY OF EDGE, FOG, AND CLOUD

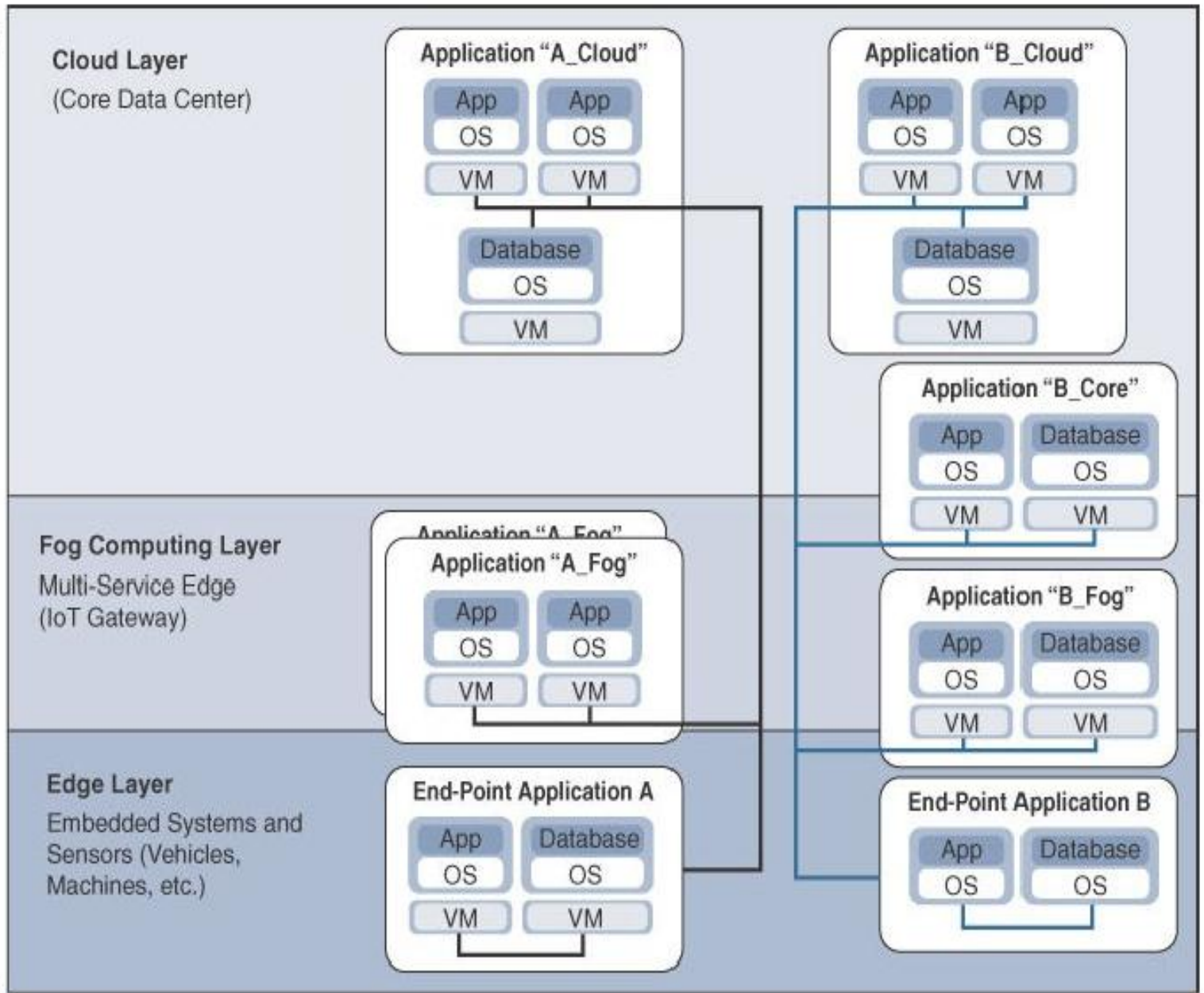
- This model suggests a hierarchical organization of network, compute, and data storage resources.
- At each stage, data is collected, analyzed, and responded to when necessary, according to the capabilities of the resources at each layer.
- advantage of hierarchy
 - response to events from resources close to the end device is fast and can result in immediate benefits
 - resources available in the cloud when necessary.



- Figure 2.16 illustrates the hierarchical nature of edge, fog, and cloud computing across an IoT system.
- From an architectural standpoint, fog nodes closest to the network edge receive the data from IoT devices.
- The fog IoT application then directs different types of data to the optimal place for analysis:



High Latency



Low Latency

Figure 2-16 *Distributed Compute and Data Management Across an IoT System*

- The fog IoT application directs different types of data for analysis:
 - 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.

