

# Unit II

# IOT Architecture

Dr. Prabhakar D. Khandait

HOD (ETC)

KDKCE, Nagpur

# Unit II

## IOT Architecture (05)

- Introduction,
- Functional View,
- Information View,
- Deployment and Operational View,
- Other Relevant architectural views.
- Real-World Design Constraints- Introduction, Technical Design constraints ,
- IoT reference model.

# IoT Reference Architecture

- The Reference Architecture is a starting point for generating concrete architectures and actual systems.
- A concrete architecture addresses the concerns of multiple stakeholders of the actual system, and it is typically presented as a series of views that address different stakeholder concerns (Kruchten 1995; SEI CMU 2013; Rozanski & Woods 2005, 2011).

- A Reference Architecture, on the other hand, serves as a guide for one or more concrete system architects.
- However, the concept of views for the presentation of an architecture is also useful for the IoT Reference Architecture.
- Views are useful for reducing the complexity of the Reference Architecture blueprints by addressing groups of concerns one group at a time.
- However, since the IoT Reference Architecture does not contain details about the environment where the actual system is deployed, some views cannot be presented in detail or at all;
- for example, the view that shows the concrete Physical Entities and Devices for a specific scenario.

- The stakeholders for a concrete IoT system are the people who use the system (Human Users); the people who design, build, and test the Resources, Services, Active Digital Artifacts, and Applications; the people who deploy Devices and attach them to Physical Entities; the people who integrate IoT capabilities of functions with an existing ICT system (e.g. of an enterprise); the people who operate, maintain, and troubleshoot the Physical and Virtual Infrastructure; and the people who buy and own an IoT system or parts thereof (e.g. city authorities).

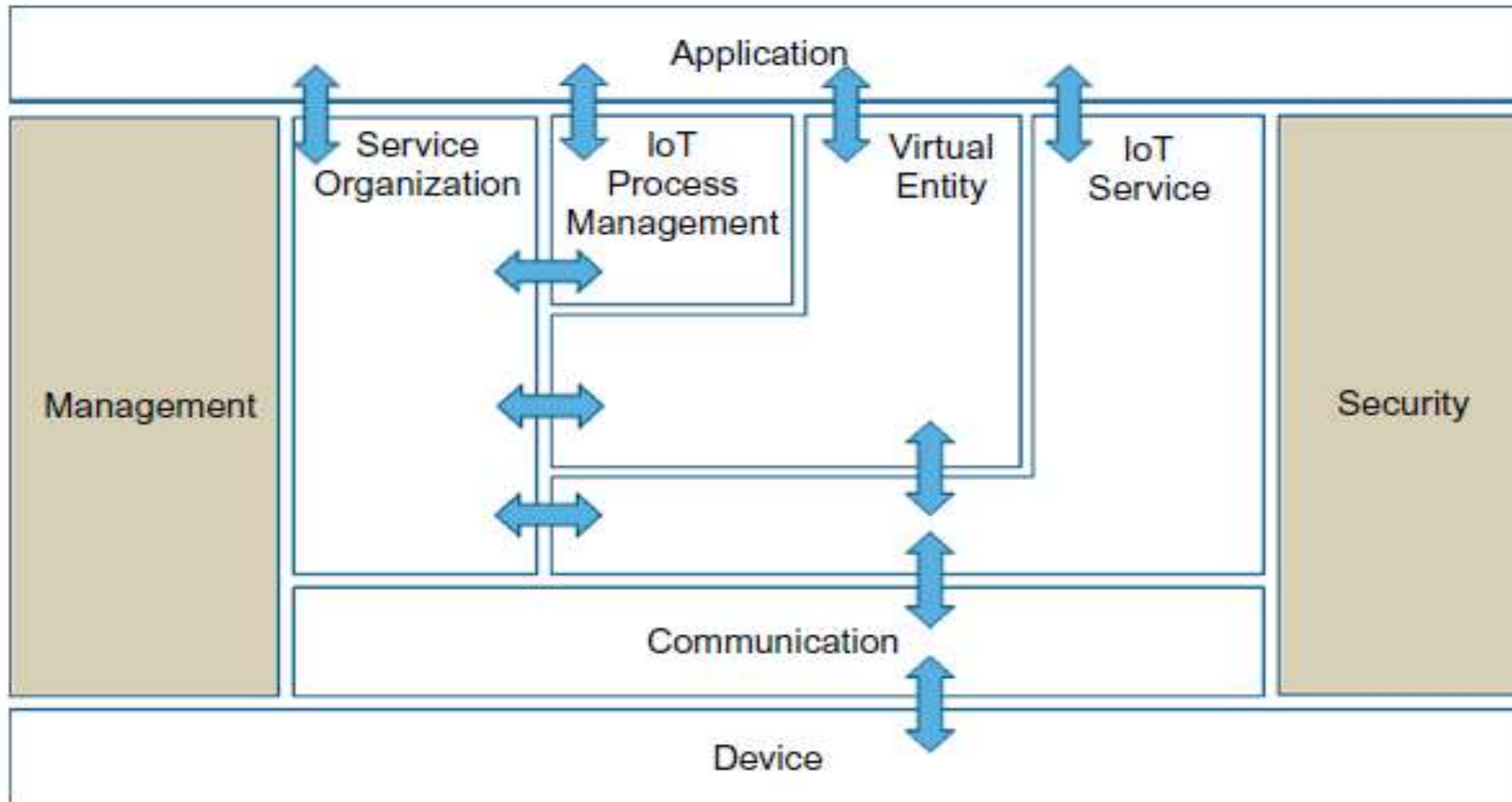
- In order to address the concerns of mainly the concrete IoT architect,
- and secondly the concerns of most of the above stakeholders, we have chosen to present the Reference Architecture as a set of architectural views (Kruchten 1995; SEI CMU 2013; Rozanski & Woods 2005, 2011):

- **Functional View:** Description of what the system does, and its main functions.
- **Information View:** Description of the data and information that the system handles.
- **Deployment and Operational View:** Description of the main real world components of the system such as devices, network routers, servers, etc.

# Functional model

- The IoT Functional Model aims at describing mainly the Functional Groups (FG) and their interaction with the ARM (Architectural Reference Model), while the **Functional View** of a Reference Architecture describes the functional components of an FG, interfaces, and interactions between the components.
- The Functional View is typically derived from the Functional Model in conjunction with high level requirements.





**FIGURE**

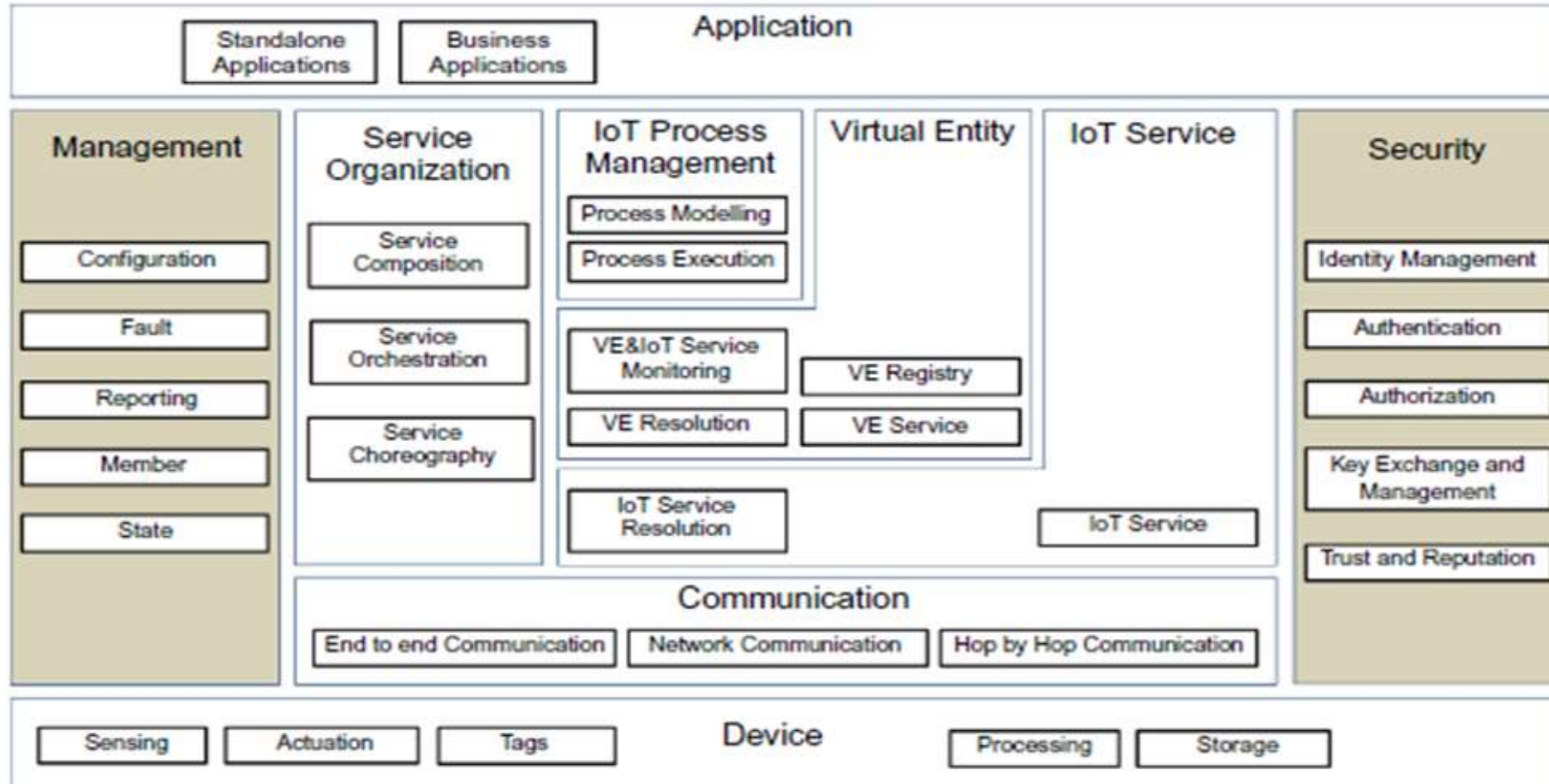
IoT-A Functional Model.

*(Carrez et al. 2013)*

# Functional view

- The functional view for the IoT Reference Architecture is presented in
- Figure 2.1, and is adapted from IoT-A (Carrez et al. 2013).
- It consists of the Functional Groups (FGs), each of which includes a set of Functional Components (FCs).
- It is important to note that not all the FCs are used in a concrete IoT architecture.

# Functional view....



**Fig 2.1 :IoT Functional View**

# Device and Application functional group

- The Device FG contains the Sensing, Actuation, Tag, Processing, Storage FCs, or simply components.
- These components represent the resources of the device attached to the Physical Entities of interest.
- The Application FG contains either standalone applications (e.g. for iOS, Android, Windows phone), or Business Applications that connect the IoT system to an Enterprise system.

# Communication functional group

- The Communication FG contains the End-to-End Communication, Network Communication, and Hop-by-Hop communication components:
- The Hop-by-Hop Communication is applicable in the case that devices are equipped with mesh radio networking technologies such as IEEE 802.15.4 for which messages have to traverse the mesh from node-to-node (hop-by-hop) until they reach a gateway node which forwards the message (if needed) further to the Internet.

# Communication functional group...

- The hop-by-hop FC is responsible for transmission and reception of
- physical and MAC layer frames to/from other devices.
- This FC has two main interfaces:
  - (a) one “southbound” to/from the actual radio on the device, and
  - (b) one “northbound” to/from the Network FC in the Communication FG.

- The Network FC is responsible for message routing & forwarding and the necessary translations of various identifiers and addresses.

The translations can be

- (a) between network layer identifiers to MAC and/or physical network identifiers,
- (b) between high-level human readable host/node identifiers to network layer addresses (e.g. Fully Qualified Domain Names (FQDN) to IP addresses, a function implemented by a Domain Name System (DNS) server), and
- (c) translation between node/service identifiers and network locators in case the higher layers above the networking layer use node or service identifiers that are decoupled from the node addresses in the network (e.g. Host Identity Protocol (HIP; Moskowitz & Nikander 2006) identifiers and IP addresses).

- The End-to-End Communication FC is responsible for end-to-end transport of application layer messages through diverse network and MAC/PHY layers.



# IoT Service functional group

- The IoT Service FG consists of two FCs: The IoT Service FC and the IoT Service Resolution FC:
- **The IoT Service FC** is a collection of service implementations, which interface the related and associated Resources.
- **The IoT Service Resolution FC** contains the necessary functions to realize a directory of IoT Services that allows dynamic management of IoT Service descriptions and discovery/lookup/resolution of IoT Services by other Active Digital Artifacts.

# Virtual Entity functional group

- The Virtual Entity FG contains functions that support the interactions between Users and Physical Things through Virtual Entity services.
- An example of such an interaction is the query to an IoT system of the form, “What is the temperature in the conference room Titan?”
- The Virtual Entity is the conference room “Titan,” and the conference room attribute of interest is “temperature.”

- Assuming that the room is actually instrumented with a temperature sensor, if the User had the knowledge of which temperature sensor is installed in the room (e.g. TempSensor #23),
- Then the User could re-formulate and re-target this query to, “What is the value of TempSensor #23?”

- The Virtual Entity Service FC enables the interaction between Users and Virtual Entities by means of reading and writing the Virtual Entity attributes (simple or complex), which can be read or written, of course.
- Some attributes (e.g. the GPS coordinates of a room) are static and non-writable by nature, and some other attributes are non-writable by access control rules.

- In general attributes that are associated with IoT Services, which in turn represent Sensor Resources, **can only be read**.
- There can be, of course, special Virtual Entities associated with the same Sensor Resource through another IoT Service that allow write operations.
- **The Virtual Entity Registry FC** maintains the Virtual Entities of interest for the specific IoT system and their associations.

# Information model

- According to the Data Information Knowledge Wisdom Pyramid
- (Rowley 2007),
- Information is defined as the enrichment of data (raw values without relevant or usable context) with the right context, so that queries about who, what, where, and when can be answered.
- Because the Virtual Entity in the IoT Domain Model is the “Thing” in the Internet of Things, the IoT information model captures the details of a Virtual Entity centric model.

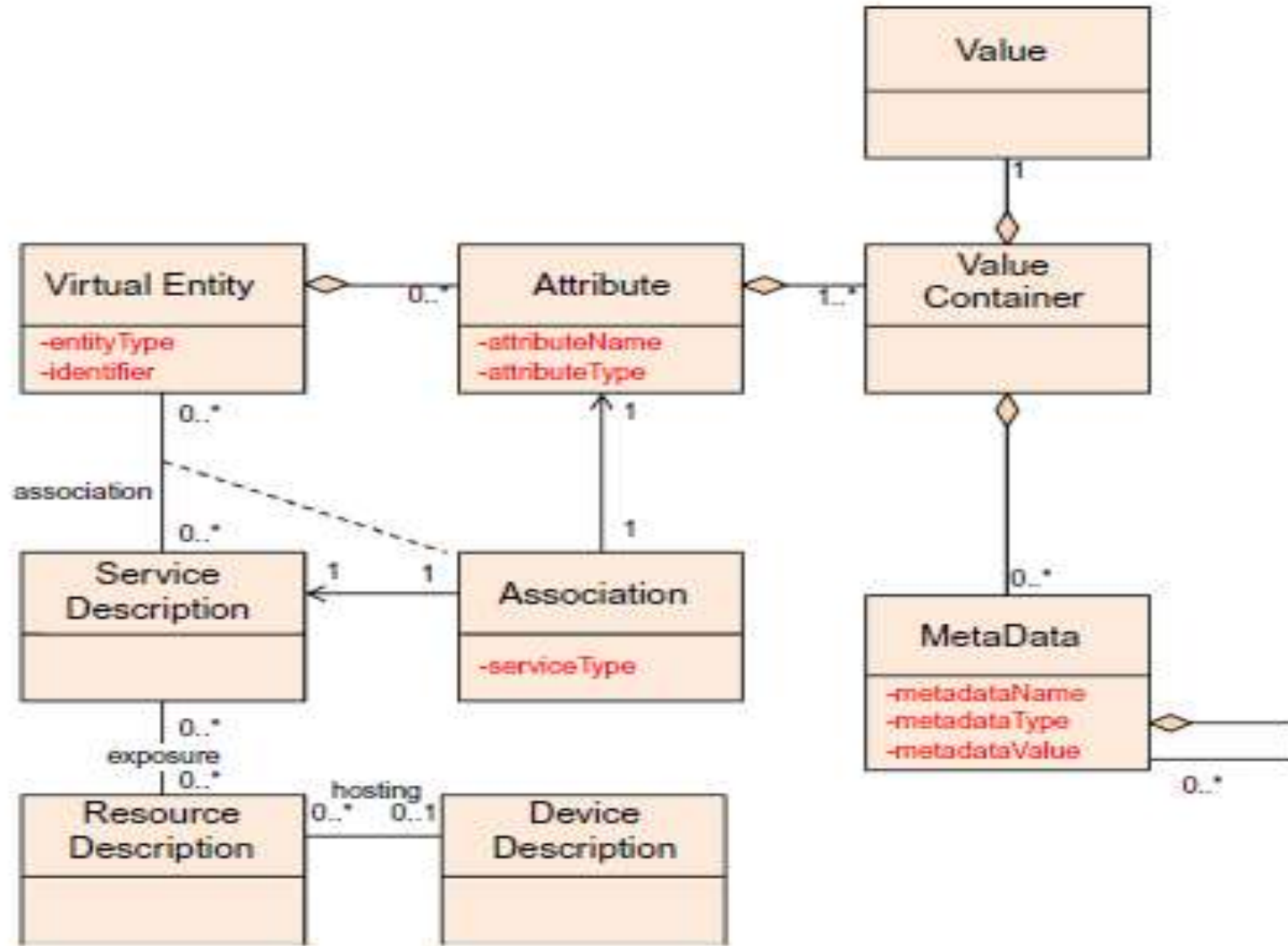
- Similar to the IoT Domain Model, the IoT Information Model is presented using Unified Modeling Language (UML) diagrams
- In UML models, attributes **represent the information, data, or properties that belong to instances of a classifier.**
- A classifier can have any number of attributes or none at all.
- Attributes describe a value or a range of values that instances of the classifier can hold.

- Each class in a UML diagram contains zero or more attributes.

These attributes are typically of simple types such as integers or text strings, and are represented with red text under the name of the class (e.g. entityType in the Virtual Entity class in **Figure next slide**).

A more complex attribute for a specific class A is represented as a class B, which is contained in class A with an aggregation relationship between class A and class B.





**FIGURE**

High-level IoT Information Model.

*(Adapted from Carrez et al. 2013)*

- **Metadata** means "data about data". Metadata is defined as the data providing information about one or more aspects of the data; it is used to summarize basic information about data that can make tracking and working with specific data easier.
- For example, a **digital image** may include metadata that describes the size of the image, its color depth, resolution, when it was created, the shutter speed, and other data.
- A text **document's metadata** may contain information about how long the document is, who the author is, when the document was written, and a short summary of the document.
- Metadata within **web pages** can also contain descriptions of page content, as well as key words linked to the content.

# Information View

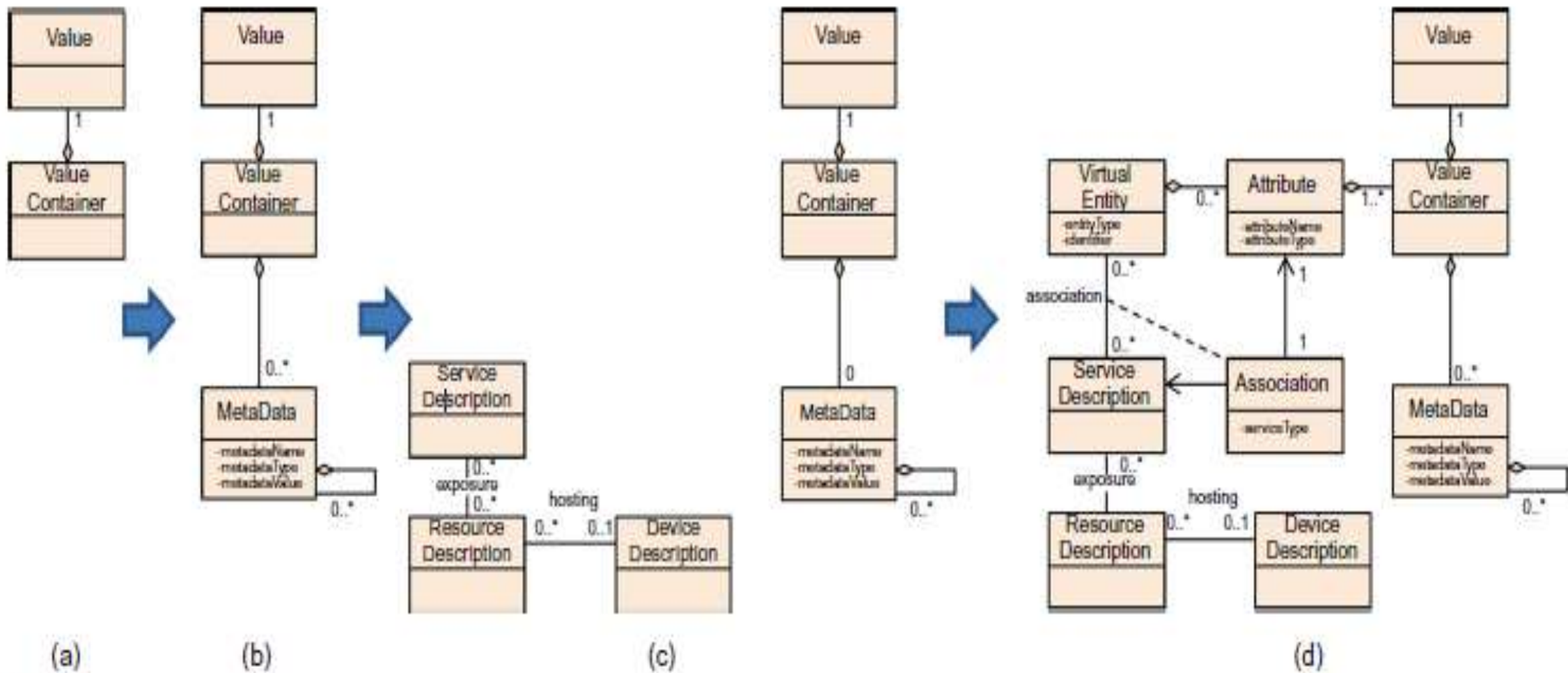
- The information view consists of
- (a) the description of the information handled in the IoT System, and (b) the way this information is handled in the system;
- In other words, the information lifecycle and flow (how information is created, processed, and deleted), and the information handling components.

# Information description

- The pieces of information handled by an IoT system
- Virtual Entity context information, i.e. the attributes (simple or complex) as represented by parts of the IoT Information model.
- IoT Service output itself is another important part of information generated by an IoT system. For example, Sensor or a Tag Service.
- Virtual Entity descriptions in general, which contain not only the attributes coming from IoT Devices (e.g. ownership information).

# Information description.....

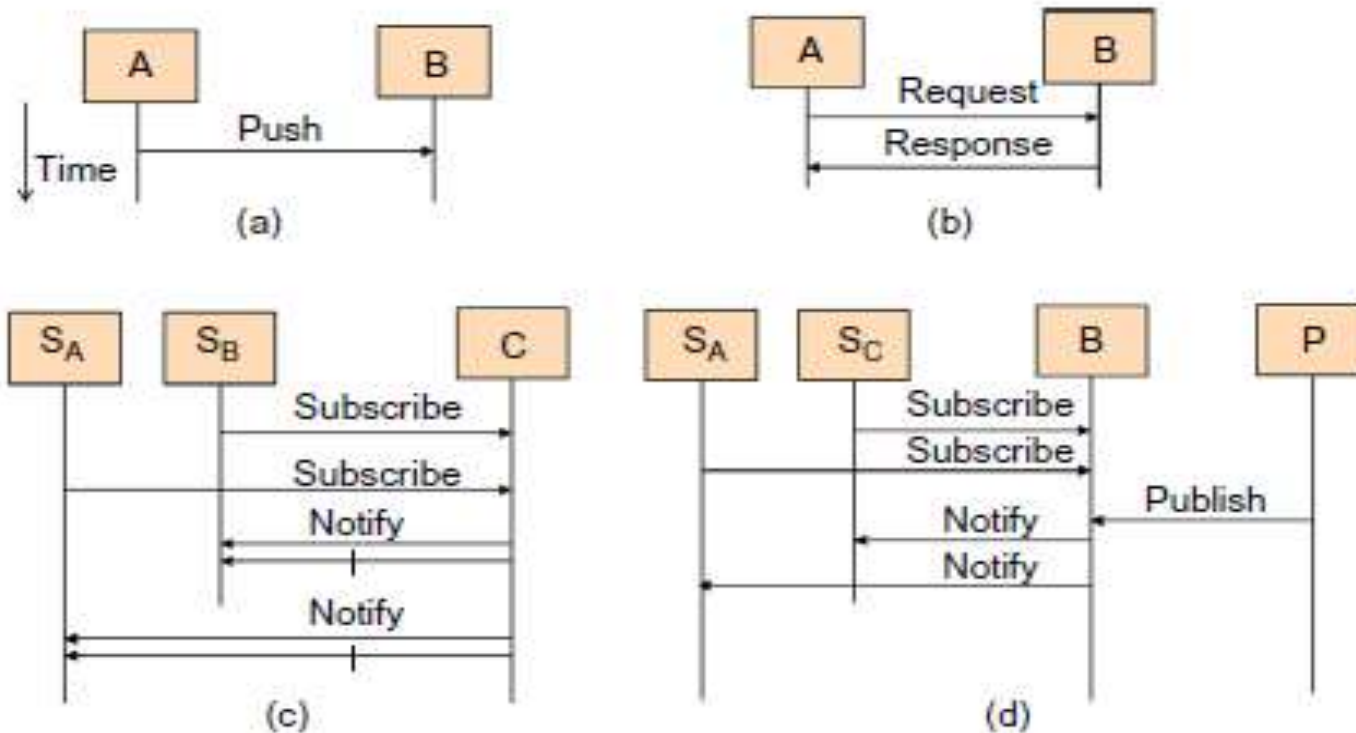
- Virtual Entity Associations with other Virtual Entities (e.g. Room #12 is on floor#7)
- **Resource Descriptions** → type of resource (e.g. sensor), identity, associated Services, and Devices.
- **Device Descriptions** → device capabilities (e.g. sensors, radios).
- **Descriptions of Composed Services** → the model of how a complex service is composed of simpler services.
- **IoT Business Process Model describes** → the steps of a business process utilizing other IoT-related services.
- Management information such as state information from operational FCs used for fault/performance purposes, configuration snapshots, reports, membership information, etc.



**FIGURE**

Information-enrichment process.

# IOT Information handling



**FIGURE**

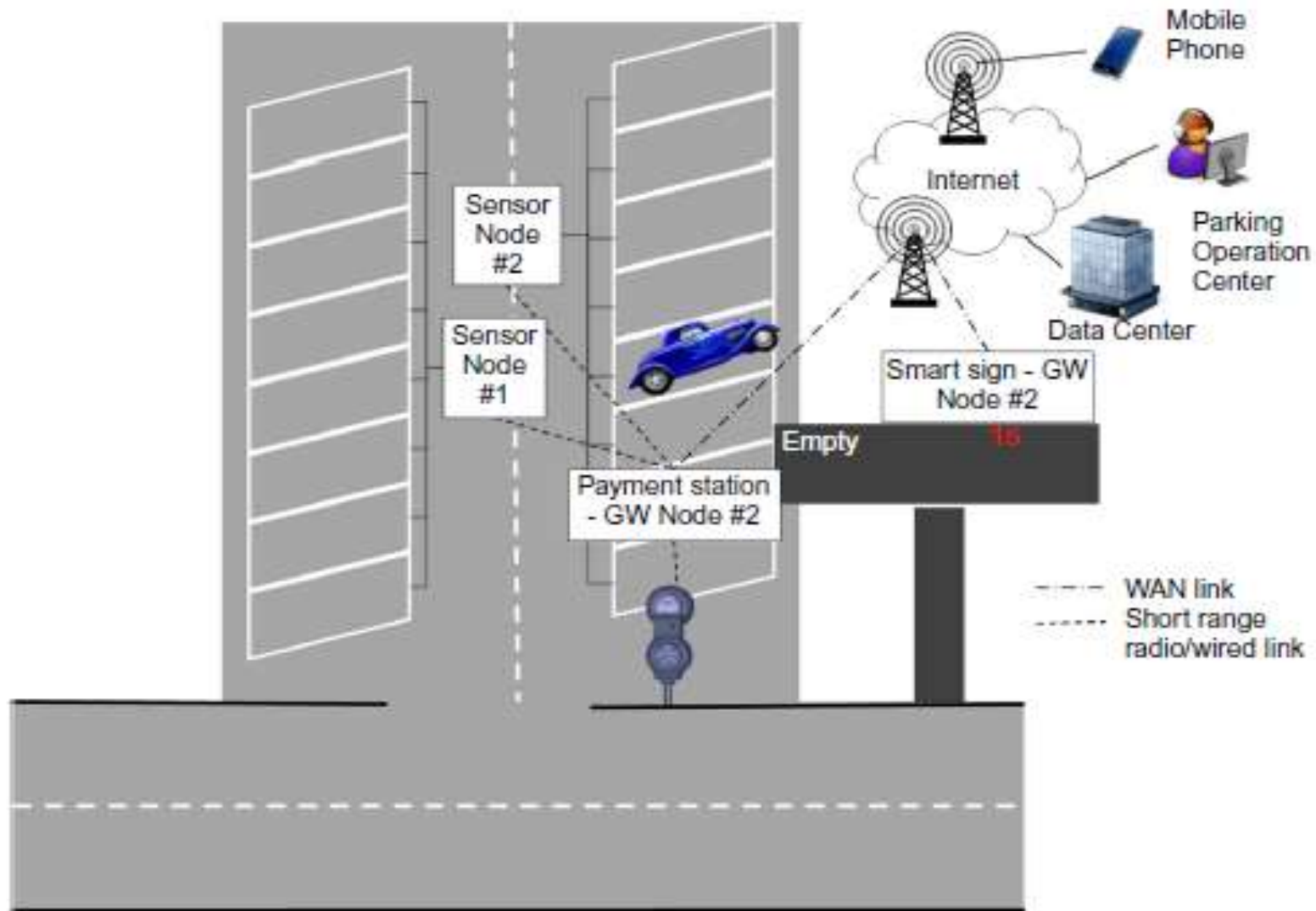
Information exchange patterns.

*(Adapted from IoT-A; Carrez et al. 2013)*

# Deployment and operational view

- The Deployment and Operational View depends on the specific actual use case and requirements, and therefore we present here one way of realizing the Parking Lot example seen earlier.
- It is by no means an exhaustive or complete example.
- Figure on next slide depicts the Devices view as Physical Entities deployed in the parking lot, as well as the occupancy sign.





**FIGURE**

Parking Lot Deployment and Operational View, Devices.

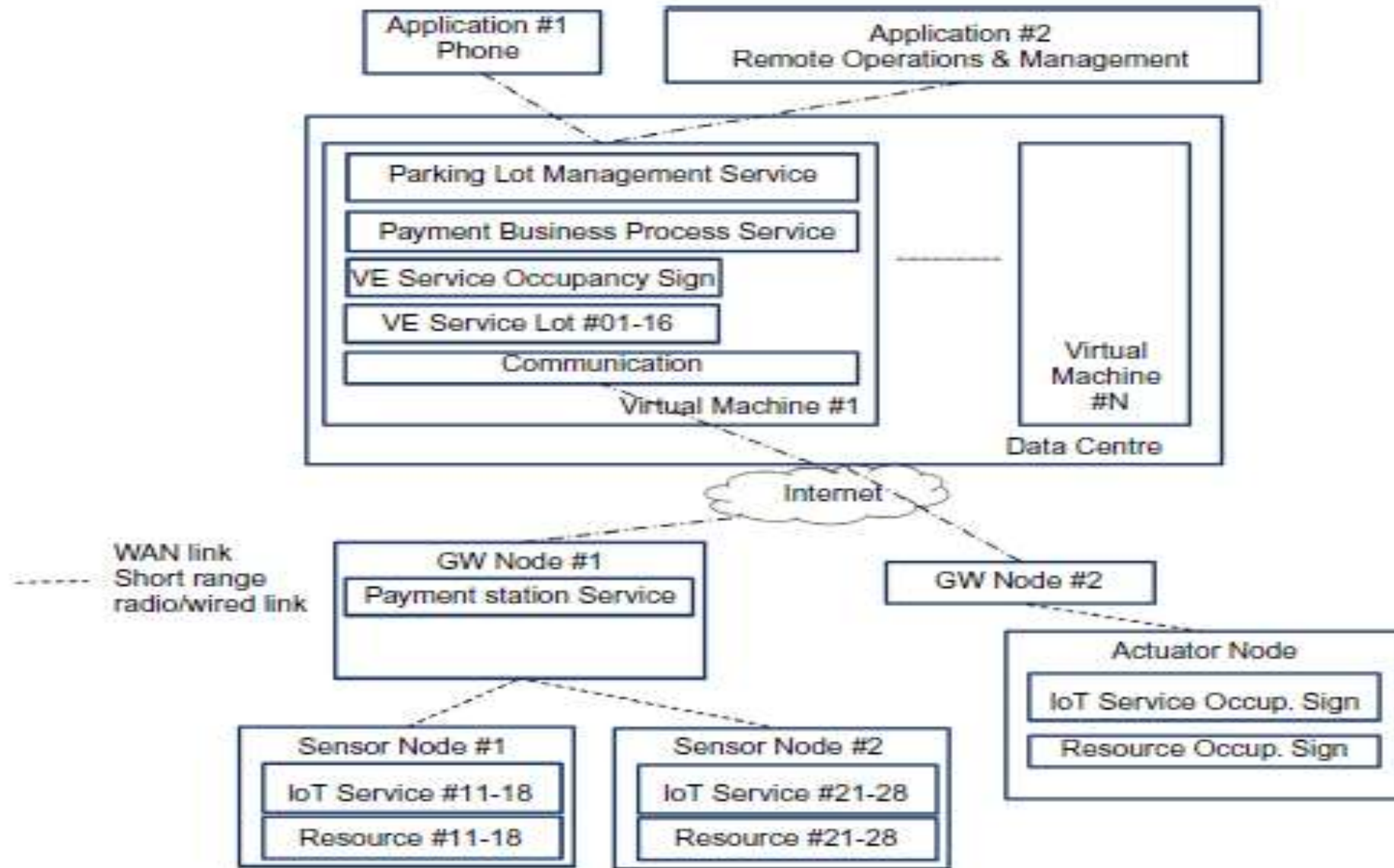
# Parking lot Example....

- There are two sensor nodes (#1 and #2), each of which are connected to eight metal/car presence sensors.
- The two sensor nodes are connected to the payment station through wireless or wired communication.
- The payment station acts both as a user interface for the driver to pay and get a payment receipt as well as a communication gateway that connects the two sensor nodes and the payment interface physical devices (displays, credit card slots, coin/note input/output, etc.) with the Internet through Wide Area Network (WAN) technology.
- The occupancy sign also acts as a communication gateway for the actuator node (display of free parking spots)

# Parking lot Example....

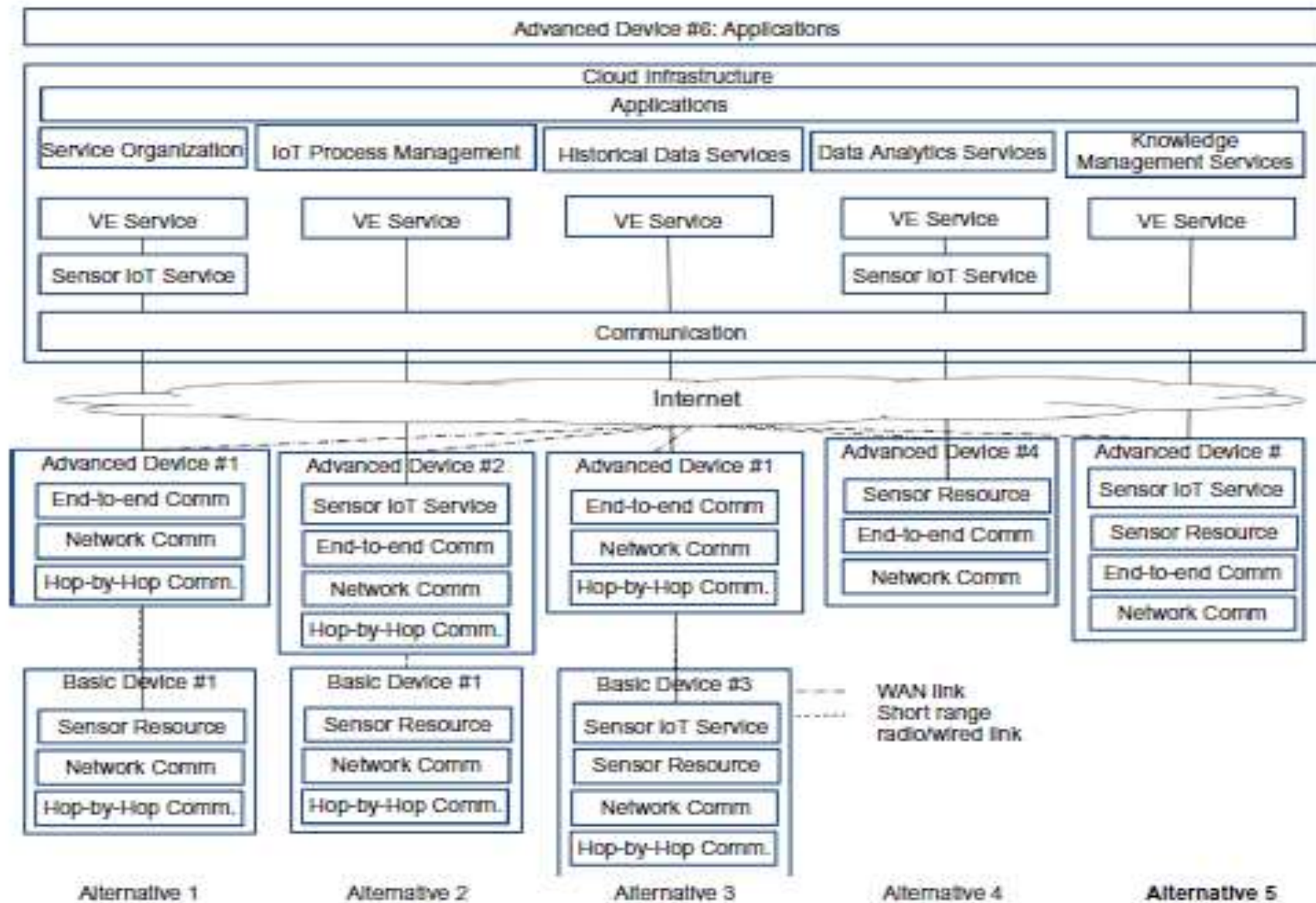
- and we assume that because of the deployment, a direct connection to the payment station is not feasible (e.g. wired connectivity is too prohibitive to be deployed or sensitive to vandalism).
- The physical gateway devices connect through a WAN technology to the Internet and towards a data center where the parking lot management system software is hosted as one of the virtual machines on a **Platform as a Service** (PaaS) configuration.
- The two main applications connected to this management system are human user mobile phone applications and parking operation center applications.
- We assume that the parking operation center manages several other parking lots using similar physical and virtual infrastructure.

- Figure on next slide shows two views superimposed, the deployment and functional views, for the parking lot example.
- Note that several FGs and FCs are omitted here for simplicity purposes, and certain non-IoT-specific Services appear in the figure because an IoT system is typically part of a larger system.



**FIGURE**

Parking Lot Deployment & Operational View, Resources, Services, Virtual Entities, Users.



**FIGURE**

Mapping IoT Domain Model concepts to Deployment View

# Other relevant architectural views

- Apart from these functional views, there are a few more that are very important for a system that interfaces the physical world.
- The two most important are the **Physical Entity View and the Context View**.

- **The Physical Entity View** describes the Physical Entities from the IoT Domain Model in terms of physical properties (e.g. dimensions for spaces/objects).
- The description of the Physical Entities may also include the relationship between Physical Entities (e.g. an entity is included in the other and may be stationary in a specific location or mobile).
- The large number of possibilities for Physical Entities cannot be captured in a Reference Architecture, nevertheless, an architect needs to outline all the details of the Physical Entities from the beginning in order to assess if any Physical Property affects the rest of the architectural views and models.



- According to Rozanski & Woods (2011), the Context of a system, “describes the relationships, dependencies, and interactions between the system and its environment (people, systems, external entities with which it interacts).”
- Therefore, the Context View should capture external entities interacting with the system, impact of the system on its environment, external entity properties/identities, system scope and responsibilities ,etc.
- Since the possibilities for external entities and their interactions with an IoT system depend on the assumptions on the actual system, this view is also constructed in the beginning of the design process because it sets the boundary conditions for the problem at hand.

- In the parking lot example above, we described briefly parts of the Physical Entity and Context View without explicit individual presentations of these views.
- For example, **the dimensions of the parking spots are a Physical Entity property**, and the fact that there are **sixteen parking spots physically** placed in a possibly gated parking lot, the fact that there is an occupancy display near the parking lot on the roadside, and other details outline both Physical Entity properties as well as relationships between the system and its environment.

# Real-World Design Constraints- Introduction

- The technical design constraints to illustrate the questions that need to be taken into account when developing and implementing M2M and IoT solutions in the real world.

## Technical design constraints hardware is popular again

- The IoT will see additional circuitry built into a number of existing products and machines from washing machines to meters.
- Giving these things an identity, and the ability to represent themselves online and communicate with applications and other things, represents a significant, widely recognized opportunity.

- For manufacturers of products that typically contain electronic components, this process will be relatively straightforward.
- Selection of appropriate communications technologies that can be integrated with legacy designs (e.g. motherboards) will be relatively painless.
- The operational environments and the criticality of the information transmitted to and from these products, however, will present some unconventional challenges and design considerations.

- The IoT will, on the other hand, allow for the development of novel applications in all imaginable scenarios.
- Emerging applications of M2M and wireless sensor and actuator networks have seen deployment of sensing capabilities in the wild that allow stakeholders to optimize their businesses, glean new insight into relevant physical and environmental processes, and understand and control situations that would have previously been inaccessible.

- The technical design of any M2M or IoT solution requires a fundamental understanding of the specificity of the intended application and business proposition, in addition to heterogeneity of existing solutions.
- Developing an end-to-end instance of an M2M or IoT solution requires the careful selection, and in most cases, development of a number of complementary technologies.
- This can be both a difficult conceptual problem and integration challenge, and requires the involvement of the key stakeholder (s) on a number of conceptual and technological levels.

- Typically, it can be considered to be a combinatorial optimization problem where the optimal solution is the one that satisfies all functional and nonfunctional requirements, whilst simultaneously delivering a satisfactory cost-benefit ratio.
- This is particularly relevant for organizations wishing to compete with existing offerings, or for start-up ventures in novel application areas. Typically, capital costs in terms of “commissioning” and operational costs in “maintenance” must be considered. These may be balanced by resultant optimizations.



# Devices and networks

- Devices that form networks in the M2M Area Network domain must be selected, or designed, with certain functionality in mind.
- At a minimum, they must have an energy source (e.g. batteries, increasingly EH), computational capability (e.g. an MCU), appropriate communications interface (e.g. a Radio Frequency Integrated Circuit (RFIC) and front end RF circuitry), memory (program and data), and sensing (and/or actuation) capability.

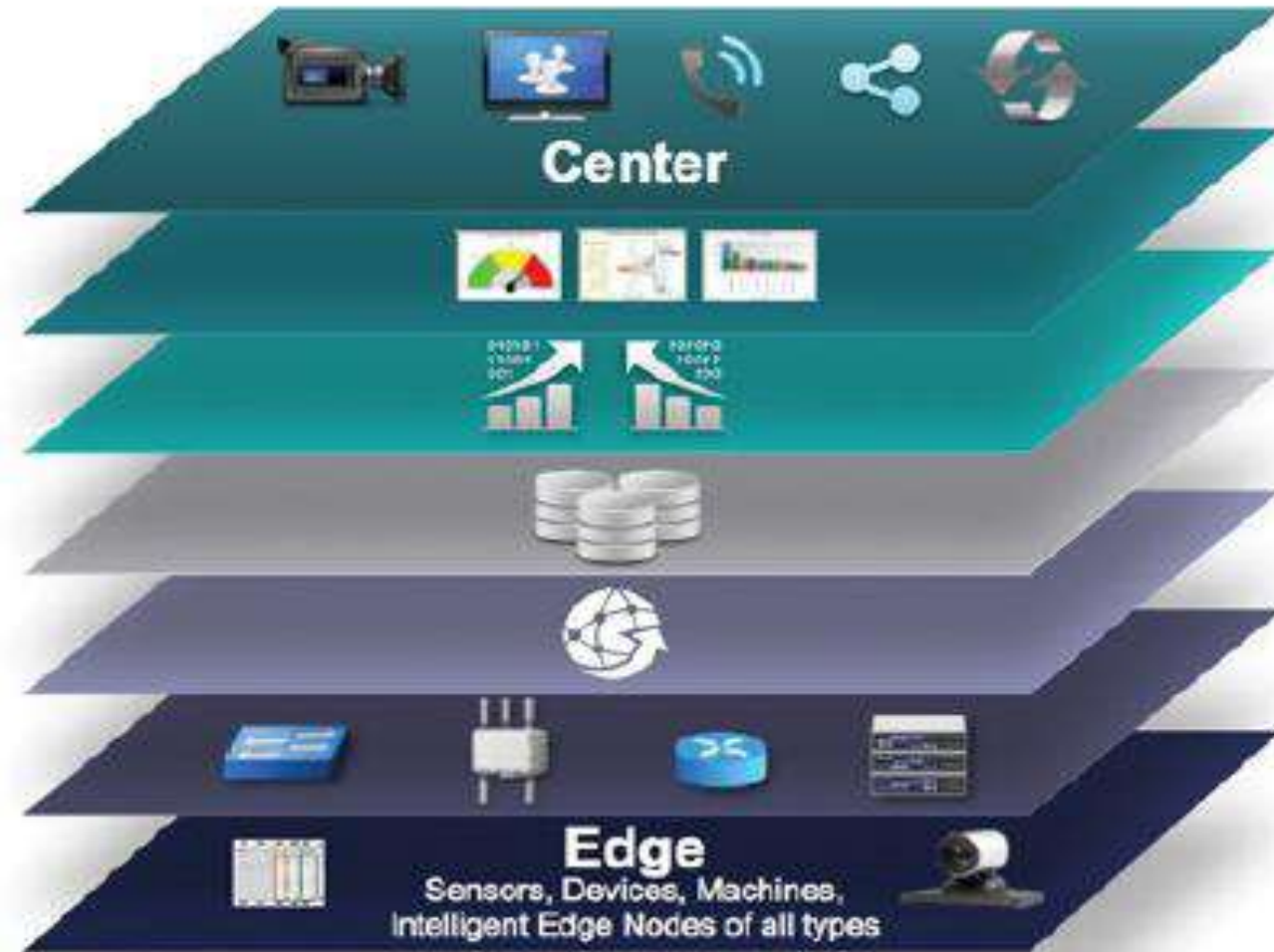
# IoT reference model

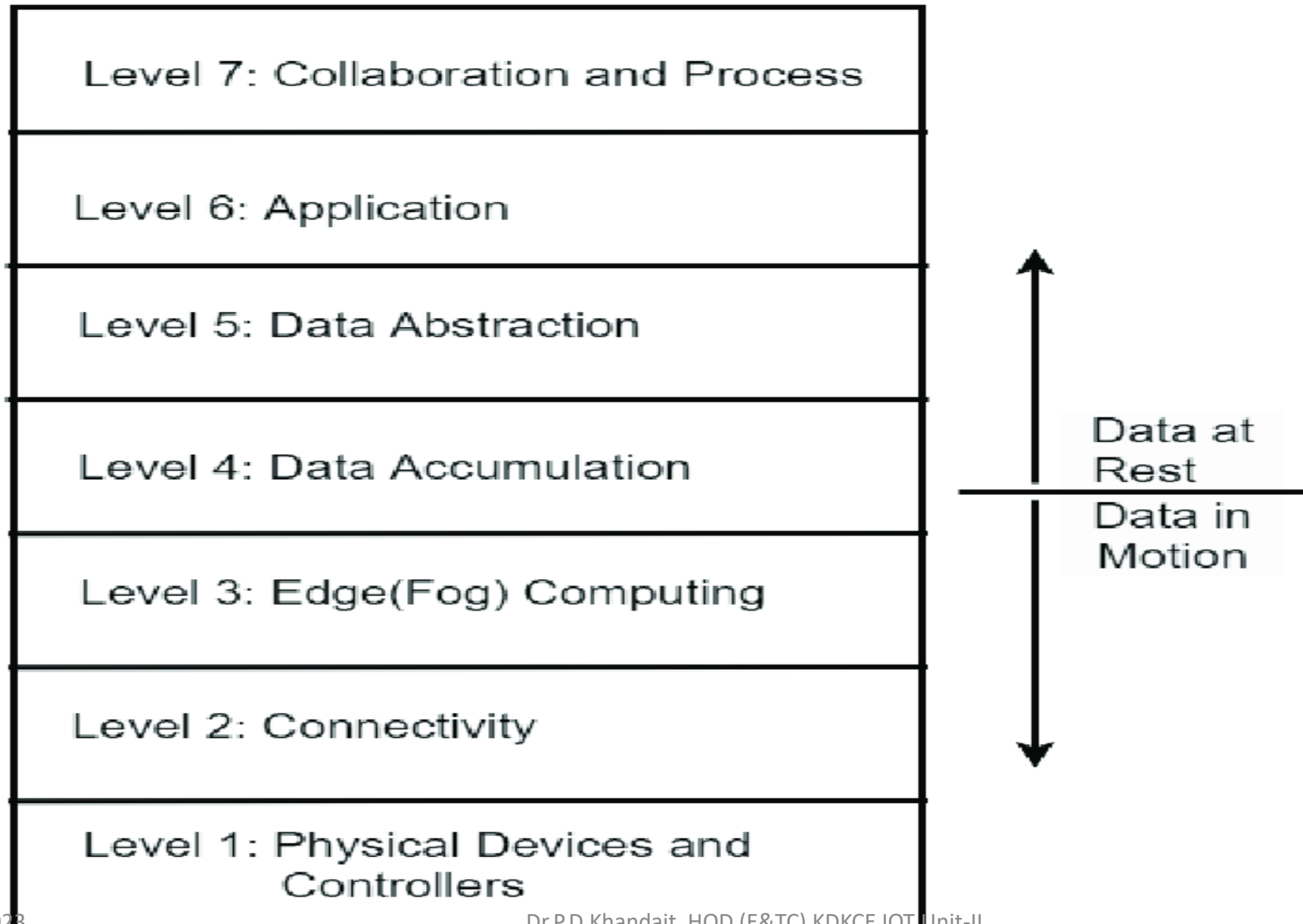
- In 2014 **The IoT World Forum (IoTWF) Standardization** architectural committee (led by Cisco, IBM, Rockwell Automation, and others) published a seven-layer IoT architectural reference model.
- While various IoT reference models exist, the one put forth by the IoT World Forum offers a clean, simplified perspective on IoT and includes edge computing, data storage, and access. It provides a succinct way of visualizing IoT from a technical perspective. \
- Each of the seven layers is broken down into specific functions, and security encompasses the entire model. Figure below details the IoT Reference Model published by the IoTWF.

# IoT reference model

## Levels

- 7 Collaboration & Processes**  
(Involving People & Business Processes)
- 6 Application**  
(Reporting, Analytics, Control)
- 5 Data Abstraction**  
(Aggregation & Access)
- 4 Data Accumulation**  
(Storage)
- 3 Edge Computing**  
(Data Element Analysis & Transformation)
- 2 Connectivity**  
(Communication & Processing Units)
- 1 Physical Devices & Controllers**  
(The "Things" in IoT)





- the IoT Reference Model defines a set of levels with control flowing from the center (this could be either a cloud service or a dedicated data center), to the edge, which includes sensors, devices, machines, and other types of intelligent end nodes.
- In general, data travels up the stack, originating from the edge, and goes northbound to the center.
- **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

# The Seven Layers

- **Layer 1: Physical Devices and Controllers Layer**

- The first layer of the IoT Reference Model is the 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**

- In the second layer of the IoT Reference Model, the focus is on connectivity.
- 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).
- Note that, the connectivity layer encompasses all networking elements of IoT and doesn't really distinguish between the last-mile network (the network between the sensor/endpoint and the IoT gateway), gateway, and backhaul networks. Functions of the connectivity layer are detailed in Figure next slide.

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



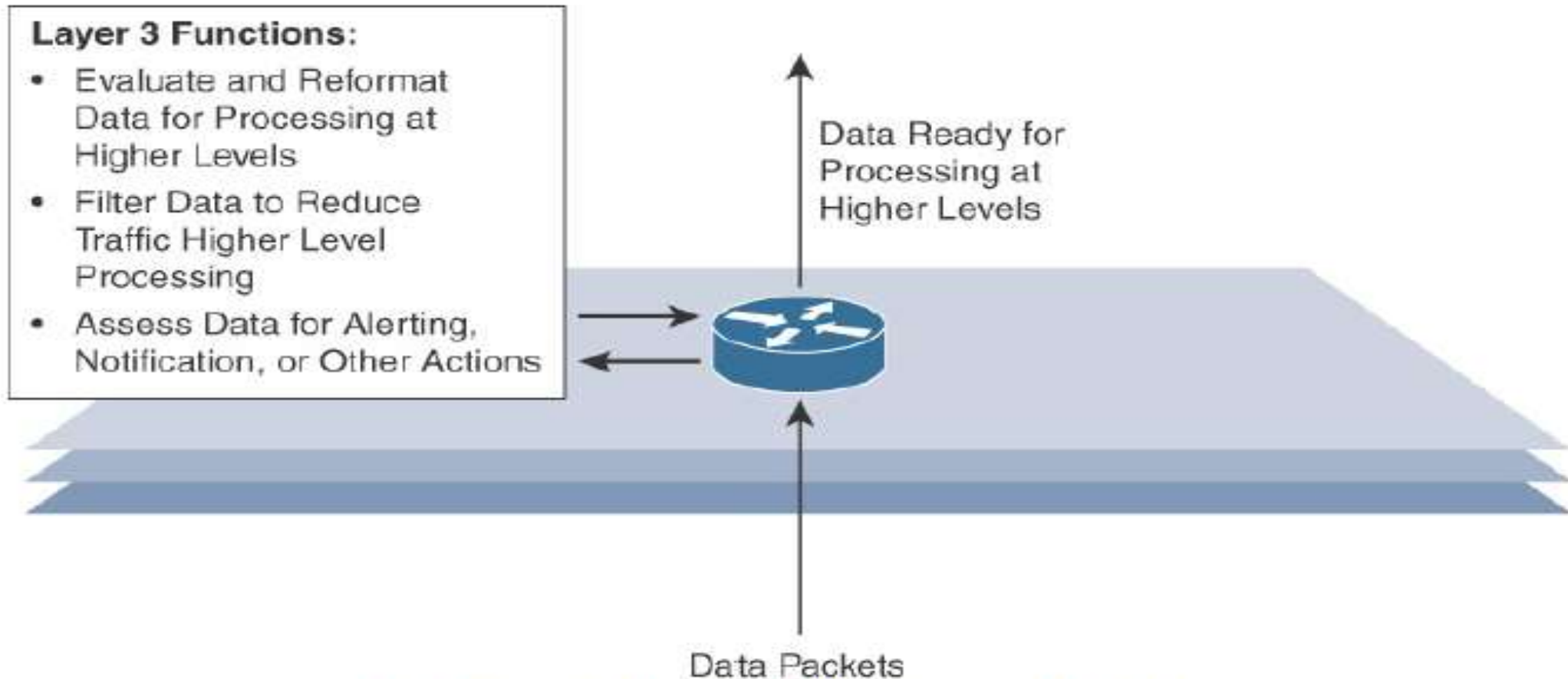
**Figure** *IoT Reference Model Connectivity Layer Functions*



- **Layer 3: Edge Computing Layer**

- Edge computing is the role of Layer 3. Edge computing is often referred to as the “fog” layer and is discussed in “Fog Computing,” .
- 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. Figure next slide highlights the functions handled by Layer 3 of the IoT Reference Model.

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



**Figure** *IoT Reference Model Layer 3 Functions*

## Layer 3: Edge Computing Layer ....

- 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. Thus, a critical function is assessing the data to see if predefined thresholds are crossed and any action or alerts need to be sent.

# Upper Layers: Layers 4–7

- The upper layers deal with handling and processing the IoT data generated by the bottom layer.
- For the sake of completeness, Layers 4–7 of the IoT Reference Model are summarized in Table next slide.

<b>IoT Reference Model Layer</b>	<b>Functions</b>
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.

**Table :** *Summary of Layers 4-7 of the IoTWF Reference Model*