IoT Architecture & Reference Models

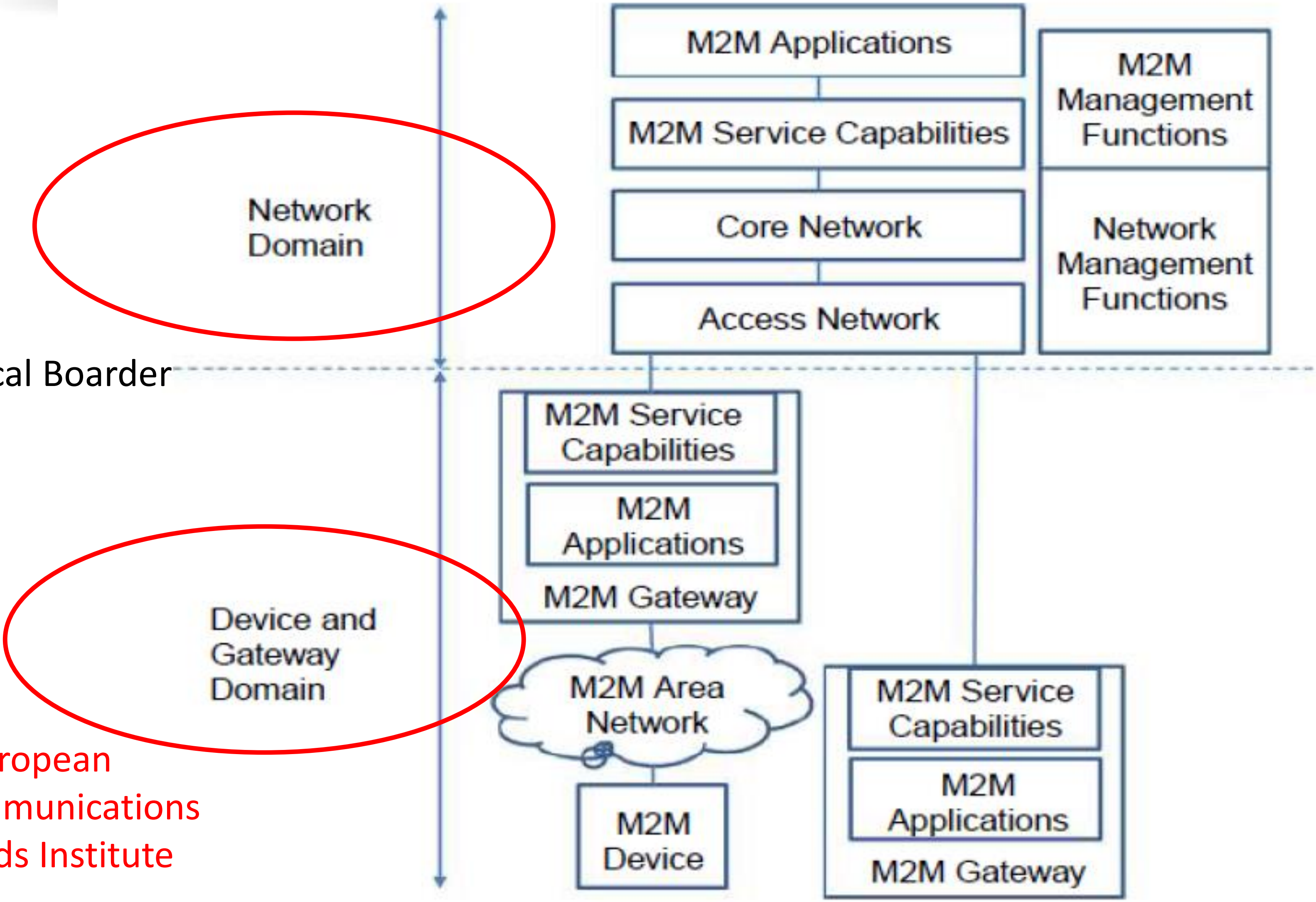
By,

Dr. Vidya Rao
Assistant Professor,
Dept of DSCA, MIT, MAHE

Outline

- ETSI M2M high-level architecture
- OGC architecture
- IoT reference model
  - Domain model
  - Information model
  - Functional model
  - Communication model
- IoT reference architecture.

ETSI M2M high-level architecture



M2M Applications

M2M Service Capabilities

Core Network

Access Network

M2M Management Functions

Network Management Functions

Network Domain

Topological Boarder

M2M Service Capabilities

M2M Applications

M2M Gateway

Device and Gateway Domain

M2M Area Network

M2M Device

M2M Service Capabilities

M2M Applications

M2M Gateway

ETSI- European Telecommunications Standards Institute

ETSI Architecture contd..

Device and Gateway Domain

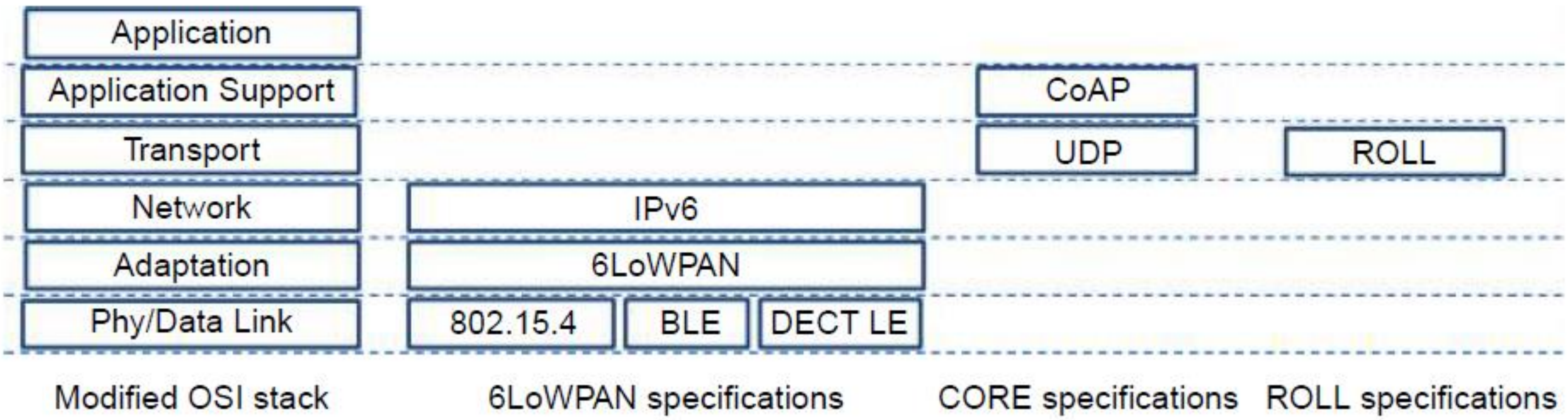
- M2M Device: Direct connection or Through one or more M2M Gateway
- M2M Area Network
- M2M Gateway

Network Domain

- Access Network
- Core Network
- M2M Service Capabilities
- M2M Applications
- Network Management Functions
- M2M Management Functions

IETF architecture for IoT

Internet Engineering Task Force Architecture



Modified OSI stack	6LoWPAN specifications	CORE specifications	ROLL specifications
Application			
Application Support		CoAP	
Transport		UDP	ROLL
Network	IPv6		
Adaptation	6LoWPAN		
Phy/Data Link	802.15.4 BLE DECT LE		

IETF Working Groups and Specification Scope

6LoWPAN (IPv6 over Low-power WPAN), CoRE (Constrained RESTful Environments), and ROLL (Routing Over Low power and Lossy networks).

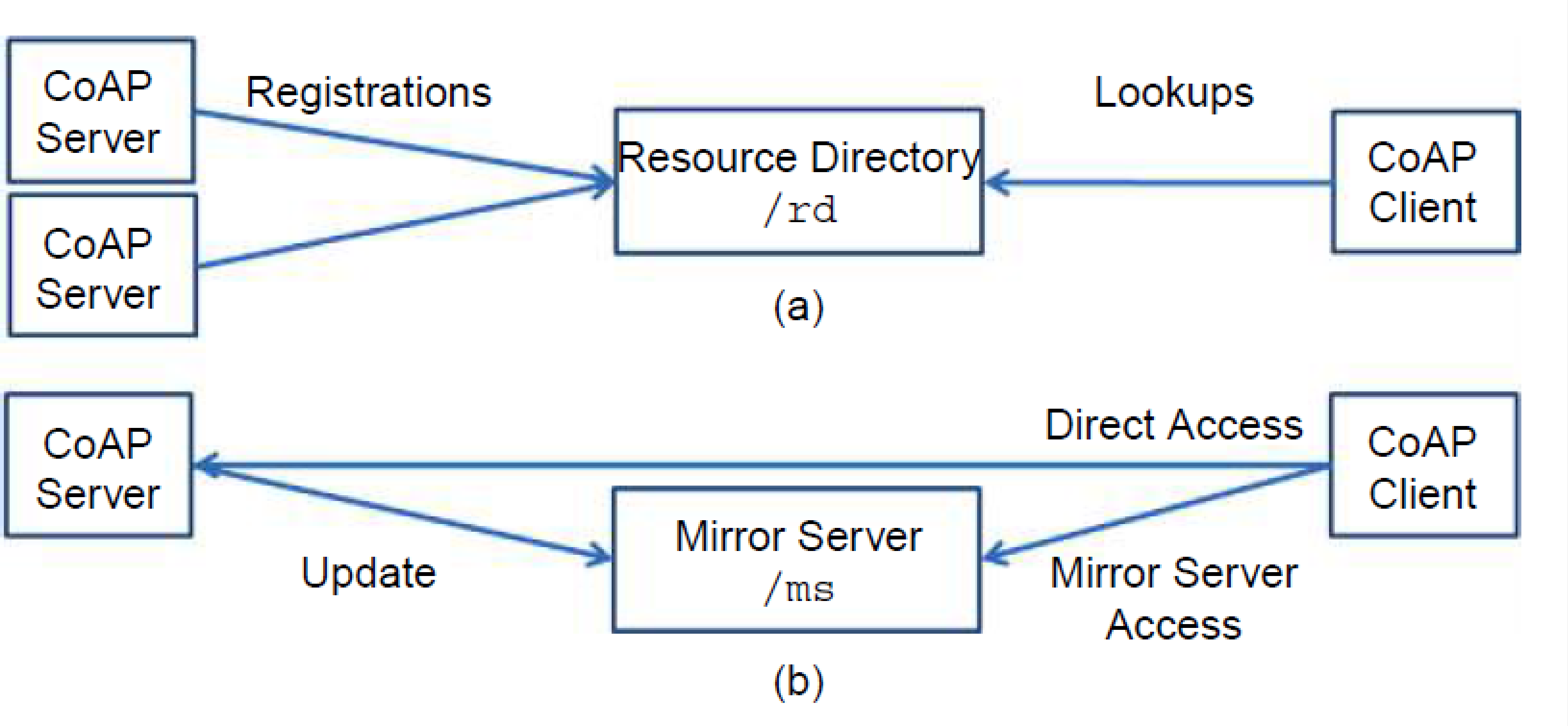
IETF Architecture contd..

- 6LoWPAN (IPv6 over Low-power WPAN), CoRE (Constrained RESTful Environments), and ROLL (Routing Over Low power and Lossy networks).
- Each set of specifications makes an attempt to address a different part of the communication stack of a constrained device.
- One layer called Application Support which includes the Presentation and Session Layers combined.
- One intermediate layer is introduced: the Adaptation Layer
- It positioned between the Physical/Data Link and the Network Layer and whose main function is to adapt the Network Layer packets to Phy/Link layer packets among others.
- An example of an adaptation layer is the 6LoWPAN layer designed to adapt IPv6 packets to IEEE 8021.5.4/Bluetooth Low Energy (BLE)/DECT Low Energy packets.

IETF Architecture contd..

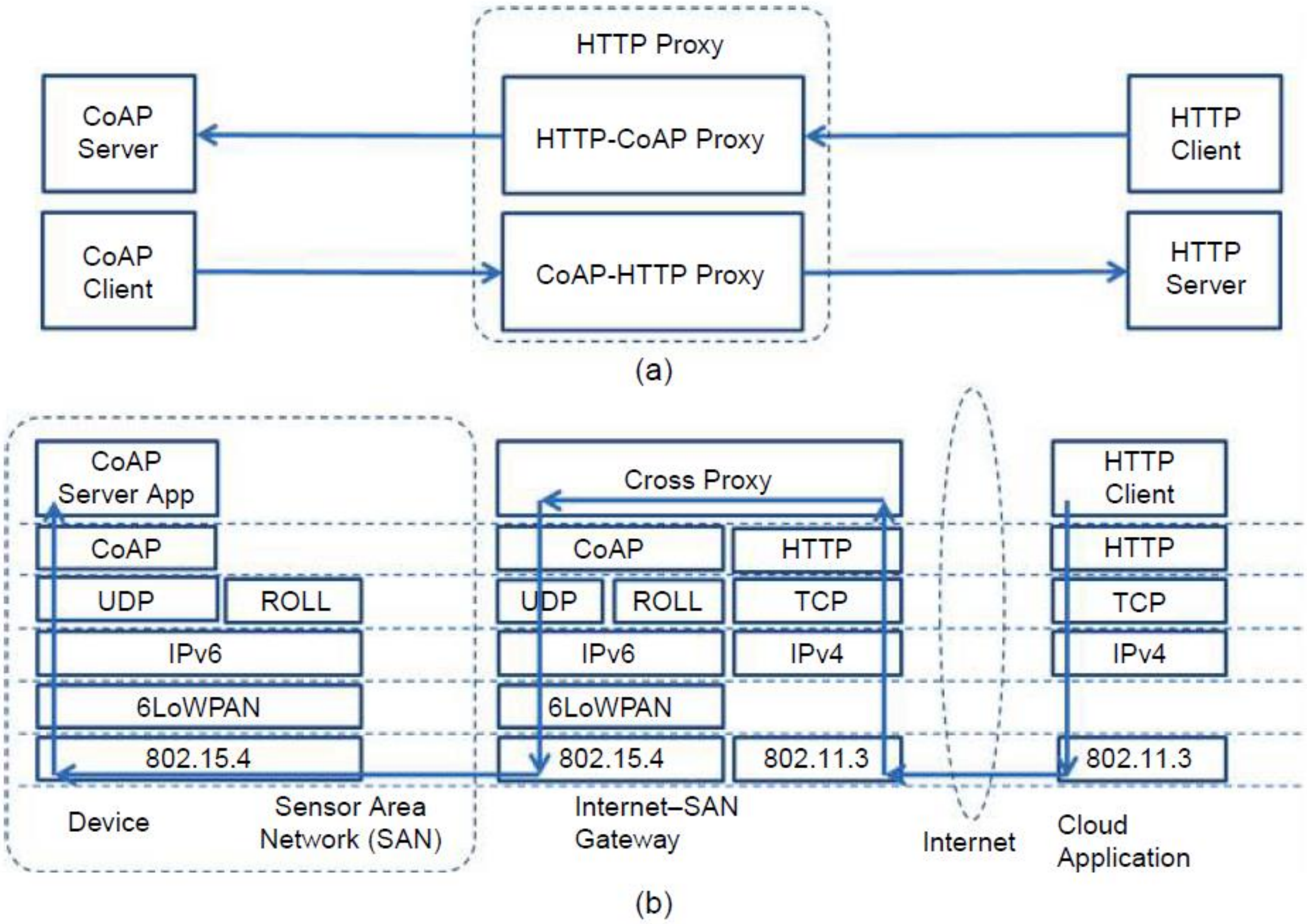
- An example of an Application Support Layer is IETF Constrained Application Protocol (CoAP), which provides reliability and RESTful operation support to applications; however, it does not describe the specific names of resources a node should host.
- The IETF CoAP draft specification describes the Transport and Application Support Layers, which essentially defines the transport packet formats, reliability support on top of UDP, and a RESTful application protocol with GET/PUT/POST/DELETE methods similar to HTTP with CoAP clients operating on CoAP server resources.

IETF CoRE Functional Components



IETF Constrained RESTful Environments (a) Resource Directory, (b) Mirror Server.

IETF CoRE HTTP Proxy



(a) possible configurations, (b) example layer interaction upon a request from a HTTP Client to a CoAP Server via a HTTP Proxy

Open Geospatial Consortium architecture

OGC includes working groups like:

- The Sensor Web Enablement (SWE)
- Domain working group, which develops standards for sensor system model (e.g. Sensor Model Language, or SensorML),
- Sensor information models (Observations & Measurements, or O&M).

The functionality that is targeted by OGC SWE includes:

- Discovery of sensor systems and observations that meet an application’s criteria.
- Discovery of a sensor’s capabilities and quality of measurements.
- Retrieval of real-time or time-series observations in standard encodings.
- Tasking of sensors to acquire observations.
- Subscription to, and publishing of, alerts to be issued by sensors or sensor services based upon certain criteria.

OGC Architecture contd..

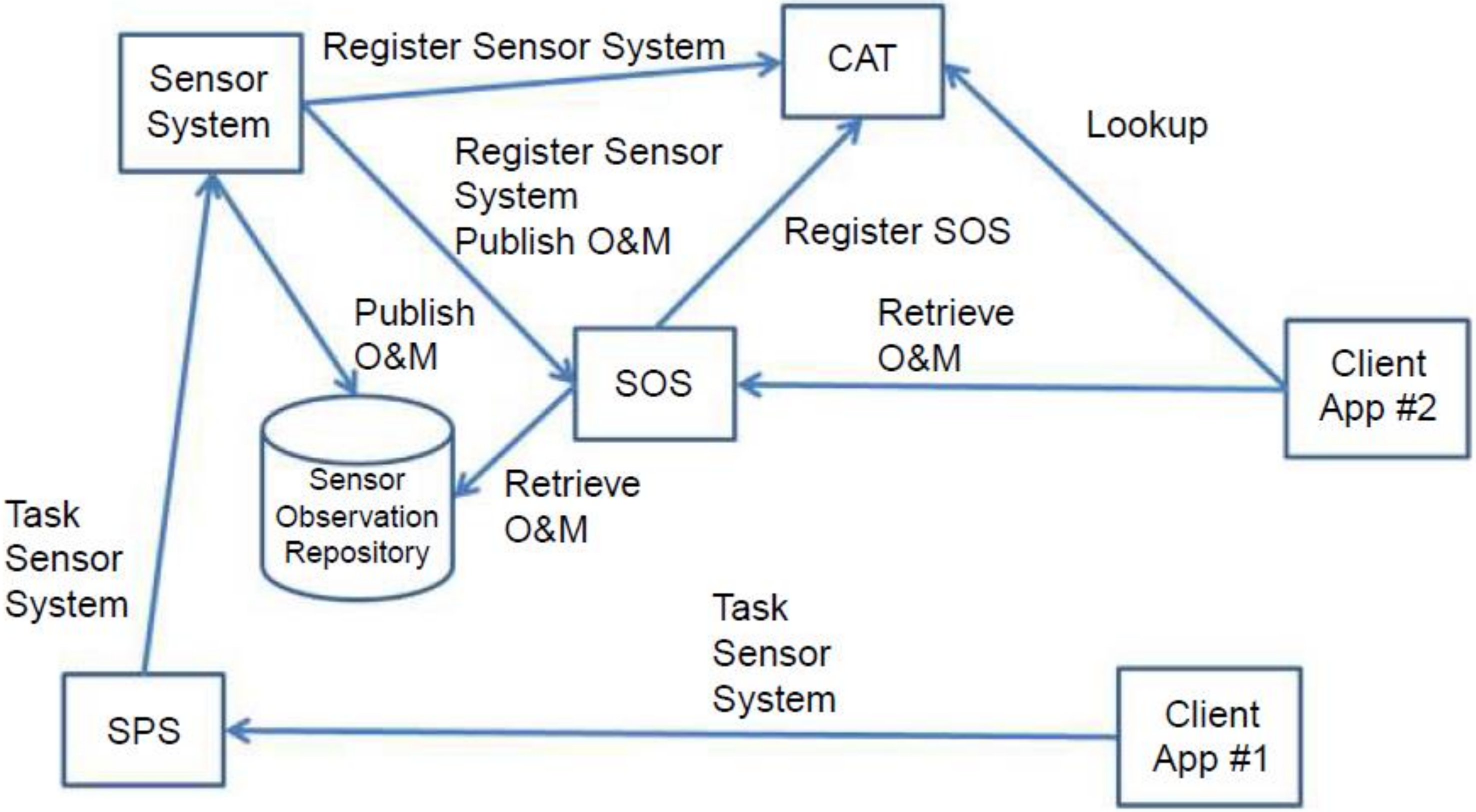


Fig. OGC functional architecture and interact

SPS-Sensor Planning Service, **SOS-**Sensor Observation Service, **CAT-** catalog, **O&M** – operational and management.

OGC Architecture contd..

OGC SWE includes the following standards:

- **SensorML and Transducer Model Language (TML):** Include a model and an XML schema for describing sensor and actuator systems and processes
- **Observations and Measurements (O&M):** Is a model and an XML schema for describing the observations and measurements for a sensor (Observations and Measurements, O&M).
- **SWE Common Data model:** For describing low-level data models (e.g. serialization in XML) in the messages exchanged between OGC SWE functional entities.
- **Sensor Observation Service (SOS):** Is a service for requesting, filtering, and retrieving observations and sensor system information.
- **Sensor Planning Service (SPS):** This is a service for applications requesting a user-defined sensor observations and measurements acquisition
- **PUCK protocol – Plug-and-work:** Which defines a protocol for retrieving sensor metadata for serial port (RS232) or Ethernet-enabled sensor devices.

OGC Architecture contd..

- OGC follows the SOA paradigm, maintains the descriptions of the existing OGC services, including the Sensor Observation and Sensor Planning Services.
- Upon installation the sensor system using the PUCK protocol retrieves the SensorML description of sensors and processes, and registers them with the Catalog so as to enable the discovery of the sensors and processes by client applications.
- The Sensor System also registers to the SOS and the SOS registers to the Catalog.

OGC Architecture contd..

- A client application #1 requests from the Sensor Planning Service that the Sensor System be tasked to sample its sensors every 10 seconds and publish the measurements using O&M and the SWE Common Data model to the SOS.
- Another client application #2 looks up the Catalog, aiming at locating an SOS for retrieving the measurements from the Sensor System.
- The application receives the contact information of the SOS and requests from the sensor observations from the specific sensor system from the SOS.
- As a response, the measurements from the sensor system using O&M and the SWE
- Common Data model are dispatched to the client application #2.
- The main objective of the OGC standards is to enable data, information, and service interoperability.

- Why you took MCA?
- What do you feel about IoT?
- Any improvement from my side needed?
- Any difficulty in understanding?
- Anything more you want from IoT?
- Have you planned anything for future?
- What are you doing in life?

IoT Reference Models

IoT Reference Models (contd..)

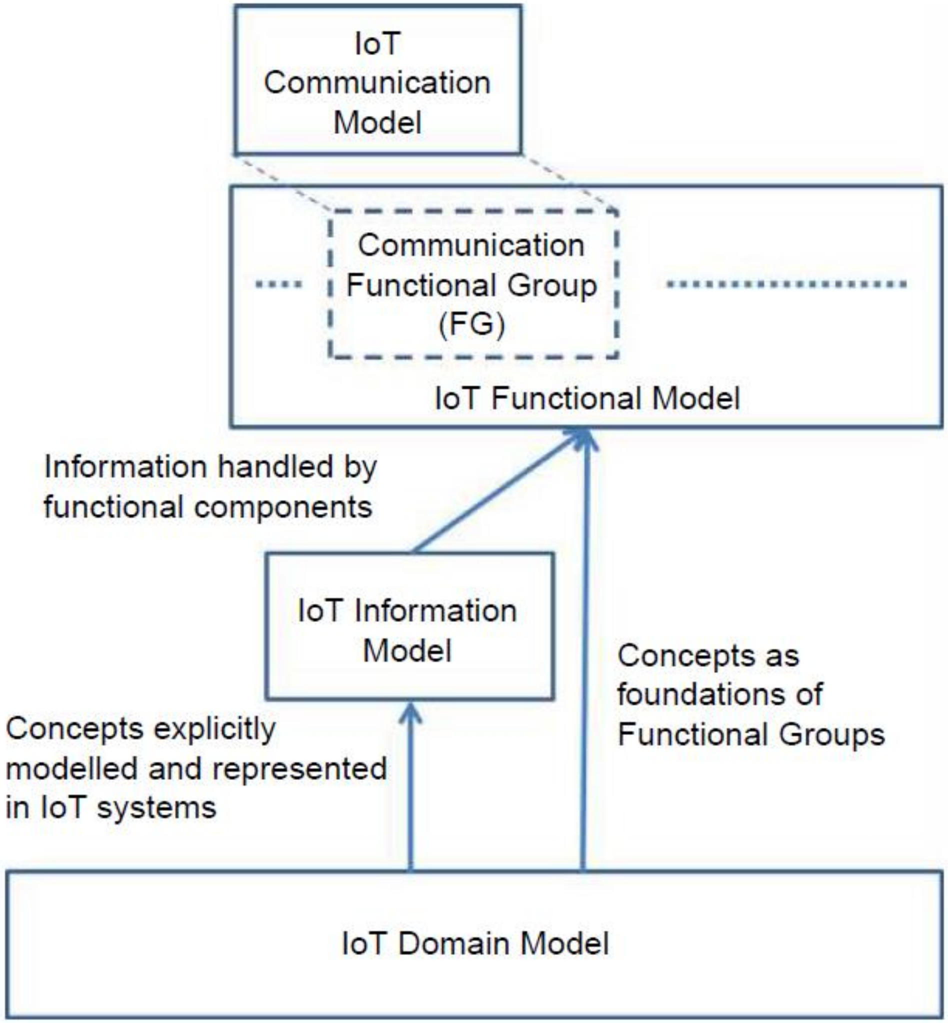

IoT Communication Model
Communication Functional Group (FG)
IoT Functional Model
Information handled by functional components
IoT Information Model
Concepts as foundations of Functional Groups
Concepts explicitly modelled and represented in IoT systems
IoT Domain Model

FIGURE 7.1

IoT Reference Model.

IoT Reference Models

- IoT domain model
- Information model
- Functional model
- Communication model
- Safety, privacy, trust, security model

IoT domain model

- A domain model defines the main concepts of a specific area of interest, in this case, the IoT.
- These concepts are expected to remain unchanged over the course of time.
- The domain model captures the basic attributes of the main concepts and the relationship between these concepts.
- A domain model also serves as a tool for human communication between people working in the domain in question and between people who work across different domains.

IoT domain model (contd..)

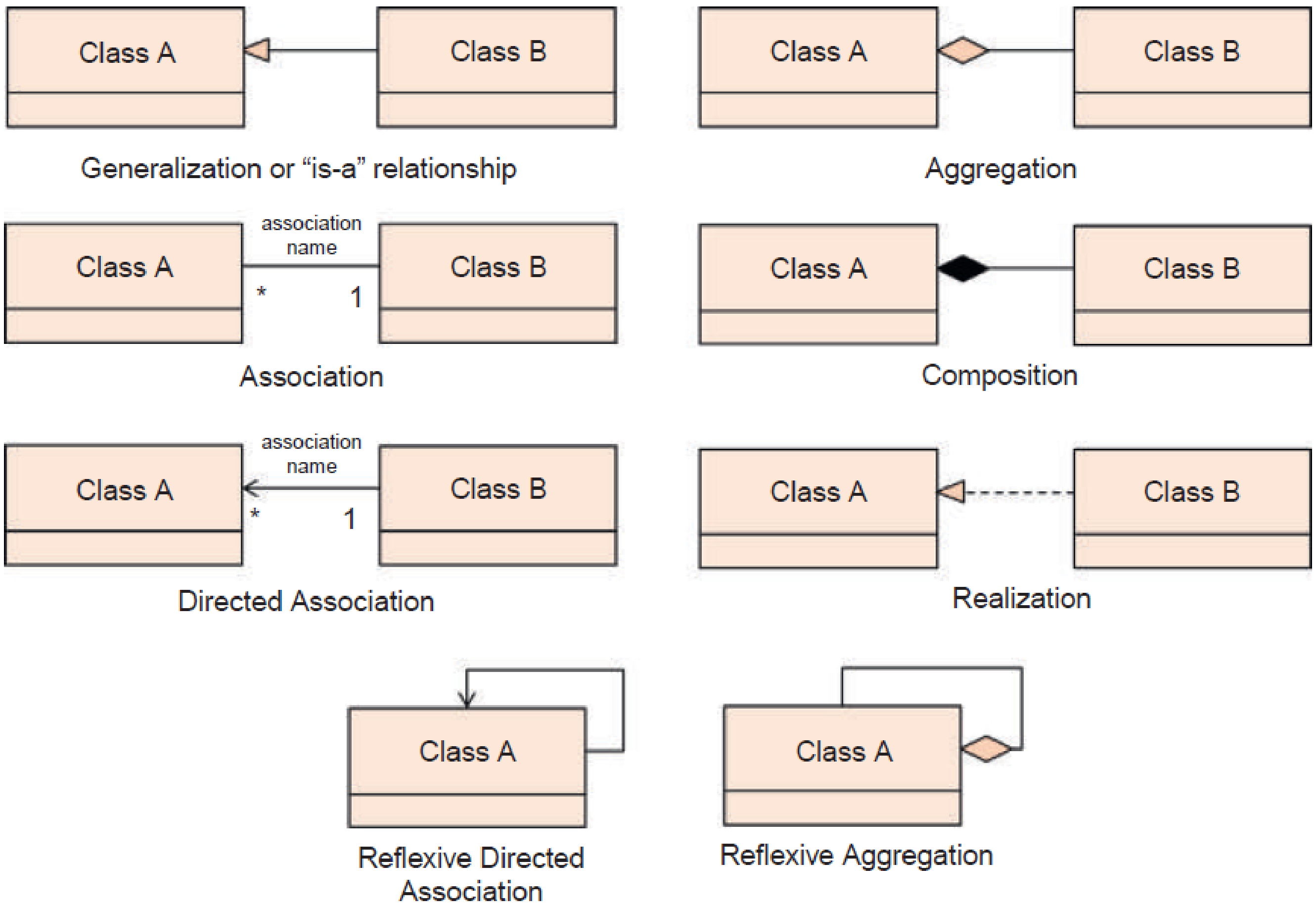


FIGURE 7.4

UML Class diagram main modeling concepts.

IoT domain model (contd..)

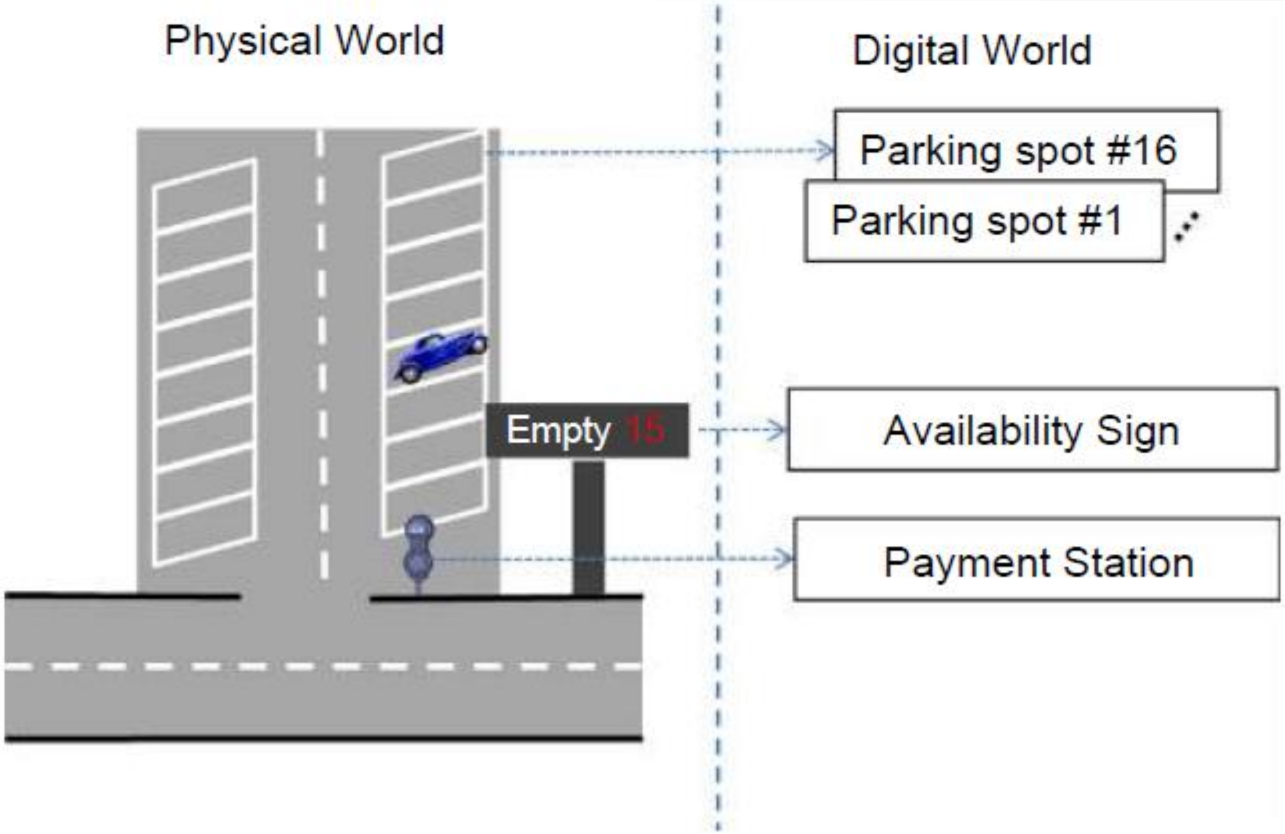


FIGURE 7.5

Physical vs. Virtual World.

IoT domain model (contd..)

- Monitoring a parking lot with 16 parking spots.
- The parking lot includes a payment station for drivers to pay for the parking spot after they park their cars.
- Physical world and digital world.
- Internet serves a rather virtual world of content and services (although these services are hosted on real physical machines)
- IoT is all about interaction through the Internet with physical Things.
- As interaction with the physical world is the key for the IoT; it needs to be captured in the domain model.
- A User and a Physical Entity are two concepts that belong to the domain model.
- A User can be a Human User, and the interaction can be physical (e.g. parking the car in the parking lot). The physical interaction is the result of the intention of the human to achieve a certain goal (e.g. park the car).
- The objects, places, and things represented as Physical Entities are the same as Assets.
- A Physical Entity is represented in the digital world as a Virtual Entity.

IoT domain model (contd..)

IoT Domain Model, **three kinds of Device** types are the most important:

1. Sensors:

- These are simple or complex Devices that typically involve a transducer that converts physical properties such as temperature into electrical signals.
- These Devices include the necessary conversion of analog electrical signals into digital signals.

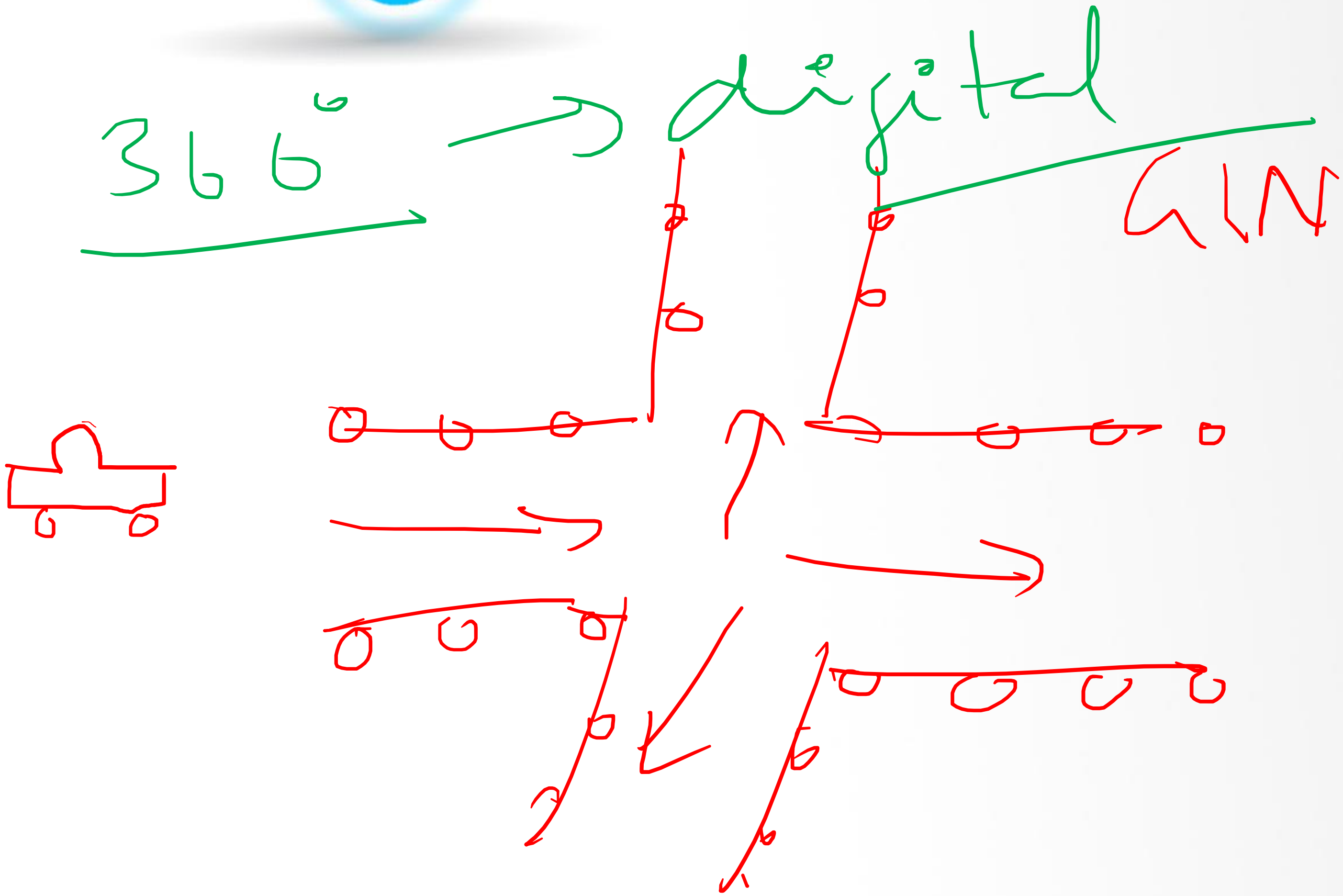
2. Actuators:

- These are also simple or complex Devices that involve a transducer that converts electrical signals to a change in a physical property (e.g. turn on a switch or move a motor).
- These Devices also include potential communication capabilities, storage of intermediate commands, processing, and conversion of digital signals to analog electrical signals.

3. Tags:

- Tags in general identify the Physical Entity that they are attached to.
- In reality, tags can be Devices or Physical Entities but not both, as the domain model shows.

IoT domain model (contd..)



IoT domain model (contd..)

IoT Services can be classified into **three main classes** according to their level of abstraction:

1. **Resource-Level Services** typically expose the functionality of a Device by exposing the on-Device Resources. In addition, these services typically handle quality aspects such as security, availability, and performance issues. An example of such a Network Resource is a historical database of measurements of a specific resource on a specific
Device.

2. **Virtual Entity-Level Services** provide information or interaction capabilities about Virtual Entities, and as a result the Service interfaces typically include an identity of the Virtual Entity.

3. **Integrated Services** are the compositions of Resource-Level and Virtual Entity-Level services, or any combination of both service classes.

Information model

- 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.
- IoT information model captures the details of a Virtual Entity centric model.
- Association class contains information about the specific association between a Virtual Entity and a related Service.
- On a high-level, the IoT Information Model maintains the necessary information about Virtual Entities and their properties or attributes.
- These properties/attributes can be static or dynamic and enter into the system in various forms, e.g. by manual data entry or reading a sensor attached to the Virtual Entity.
- Virtual Entity attributes can also be digital synchronized copies of the state of an actuator.

Information model (contd..)

- The associated services are related to Resources and Devices as seen from the IoT Domain Model.
- A Virtual Entity object contains simple attributes/properties:

  (a) entityType to denote the type of entity, such as a human, car, or room (the entity type can be a reference to concepts of a domain ontology, e.g. a car ontology);

  (b) a unique identifier; and

  (c) zero or more complex attributes of the class Attributes.
- The class Attributes should not be confused with the simple attributes of each class.
- **This class Attribute is used as a grouping mechanism for complex attributes of the Virtual Entity.**
- Objects of the class Attributes, in turn, contain the simple attributes with the self-descriptive names attributeName and attribute type.

Information model (contd..)

- The attribute serviceType can take two values:

  Sensor (a) “INFORMATION,” if the associated service is a sensor service (i.e. allows reading of the sensor), or

  Job (b) “ACTUATION,” if the associated service is an actuation service (i.e. allows an action executed on an actuator).

- In both cases, the eventual value of the attribute will be a result of either reading a sensor or controlling an actuator.

Information model (contd..)

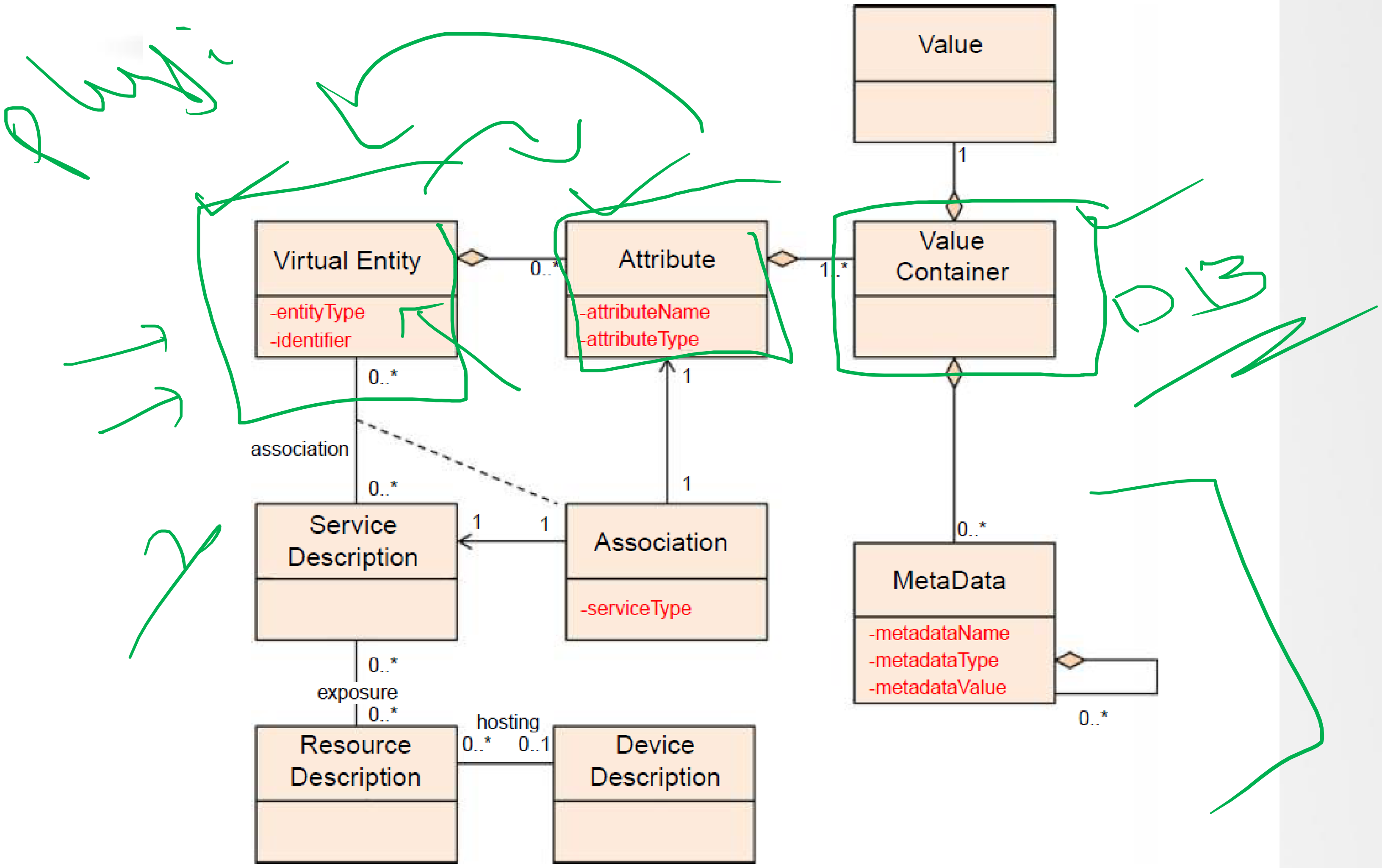


FIGURE 7.8

High-level IoT Information Model.

Information model (contd..)

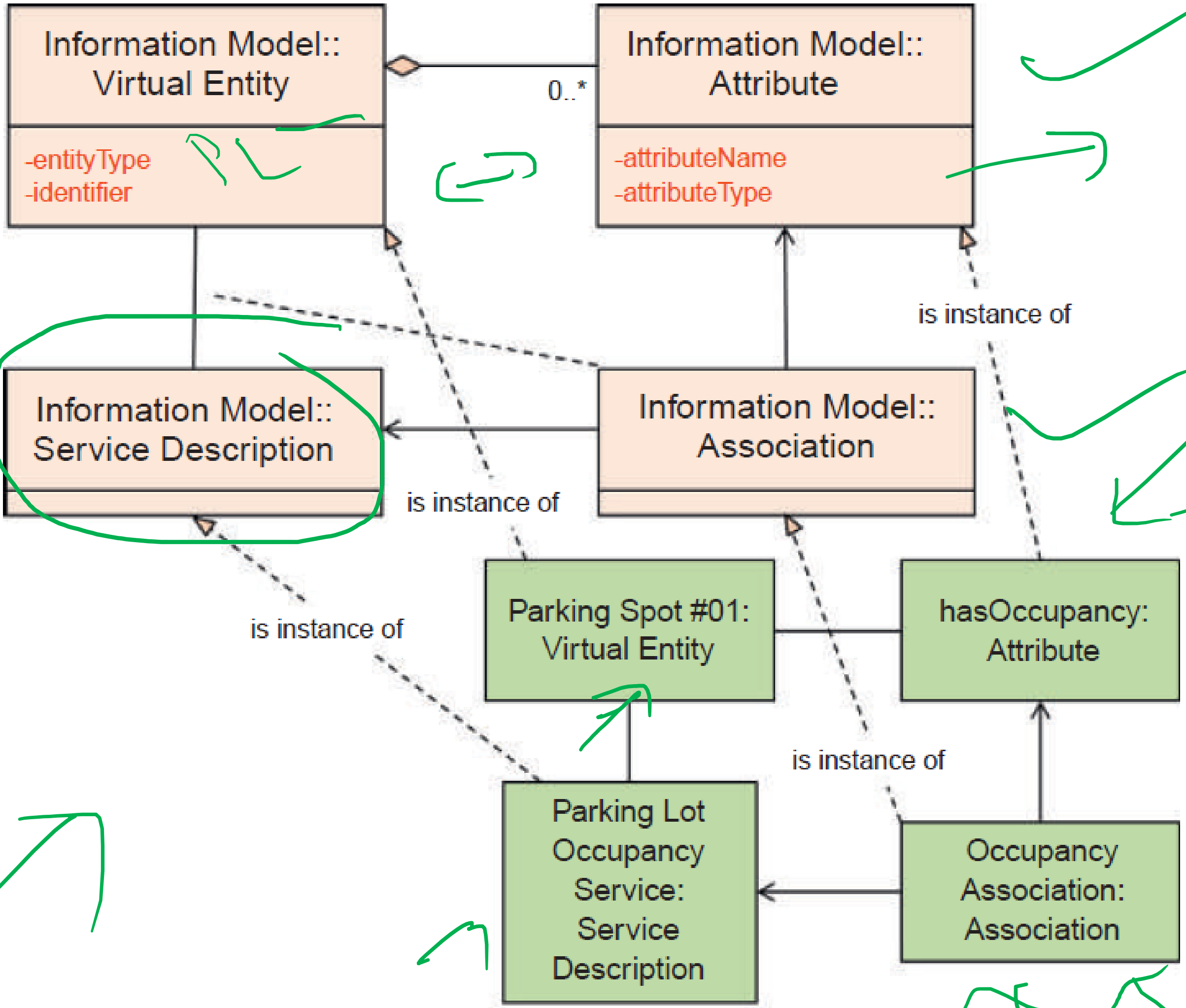


FIGURE 7.9

IoT Information Model example.

Information model (contd..)

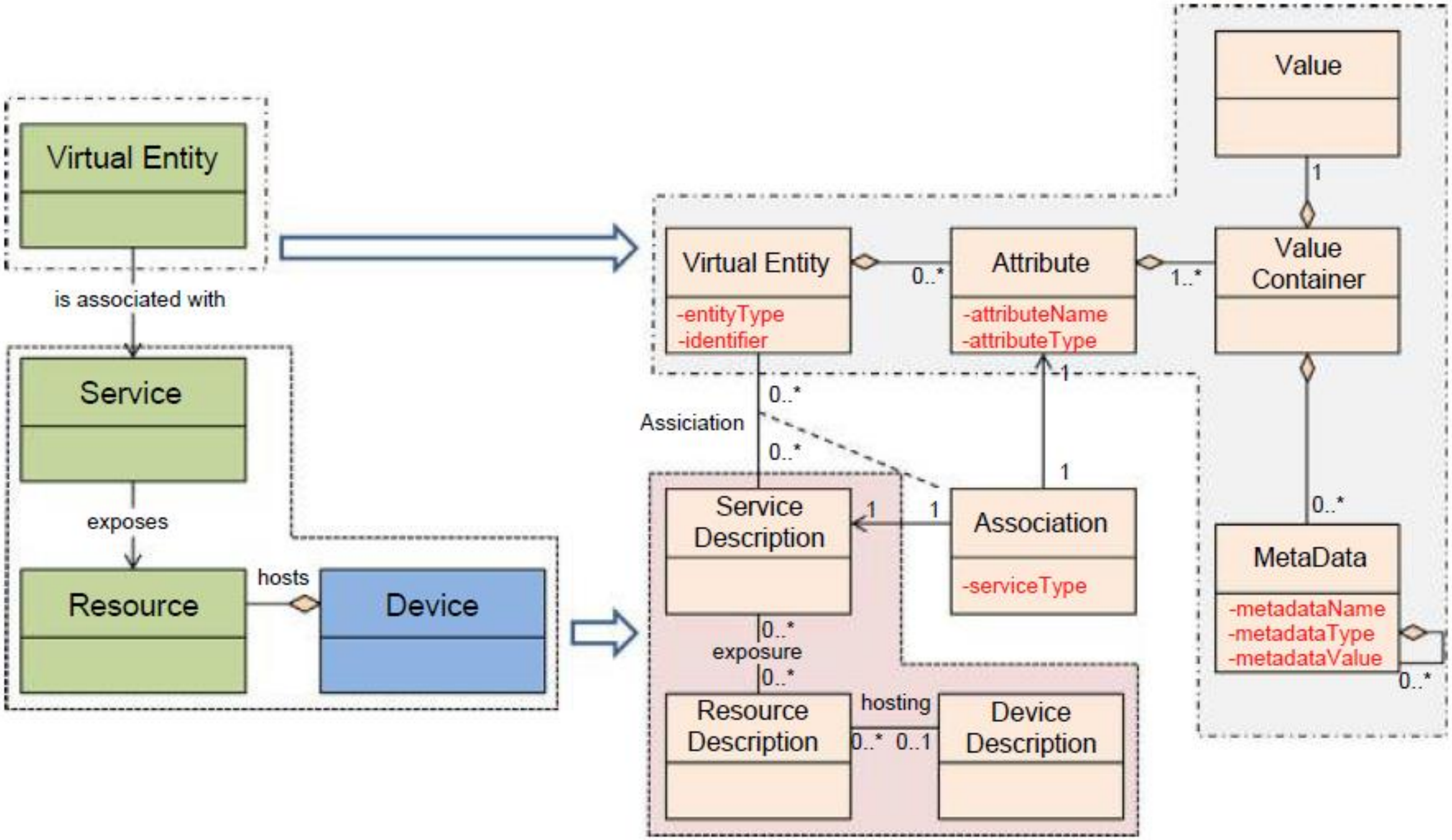


FIGURE 7.10

Relationship between core concepts of IoT Domain Model and IoT Information Model.

Information model (contd..)

Attributes or properties that could exist in a Virtual Entity description:

1. **Location and its temporal information are important:** These properties are extremely important when the interested Physical Entities are mobile (e.g. a moving car). A mobile Physical Entity affects the associations between Attributes and related Services, e.g. a person moving close to a camera (sensor) is associated with the Device, Resource, and Services offered by the camera for as long as she stays within the field of view of the camera.
2. **Even non-moving Virtual Entities contain properties that are dynamic with time**, and therefore their temporal variations need to be modeled and captured by an information model.
3. **Information such as ownership is also important in commercial settings** because it may determine access control rules or liability issues. It is important to note that the Attribute class is general enough to capture all the interesting properties of a Physical Entity, and thus provides an extensible model whose details can only be specified by the specific actual system in mind.

Information model (contd..)

The Service Description contains the following :

1. Service type
2. Service area
3. Service schedule
4. Associated resources
5. Metadata or semantic information

Information model (contd..)

A Resource description contains the following information:

1. Resource name and identifier for facilitating resource discovery.
2. Resource type, which specifies if the resource is
   - (a) a sensor resource, which provides sensor readings;
   - (b) an actuator resource, which provides actuation capabilities (to affect the physical world) and actuator state;
   - (c) a processor resource, which provides processing of sensor data and output of processed data;
   - (d) a storage resource, which provides storage of data about a Physical Entity;
   - (e) a tag resource, which provides identification data for Physical Entities.
3. Free text attributes or tags used for capturing typical manual input such as “fire alarm, ceiling.”
4. Indicator of whether the resource is an on-Device resource or network resource.
5. Location information about the Device that hosts this resource in case of an on-Device resource.
6. Associated Service information.
7. Associated Device description information.

Functional model

- The IoT Functional Model aims at describing mainly the Functional Groups (FG) and their interaction with the Architecture Reference Model (ARM).
- 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.

Functional model (contd..)

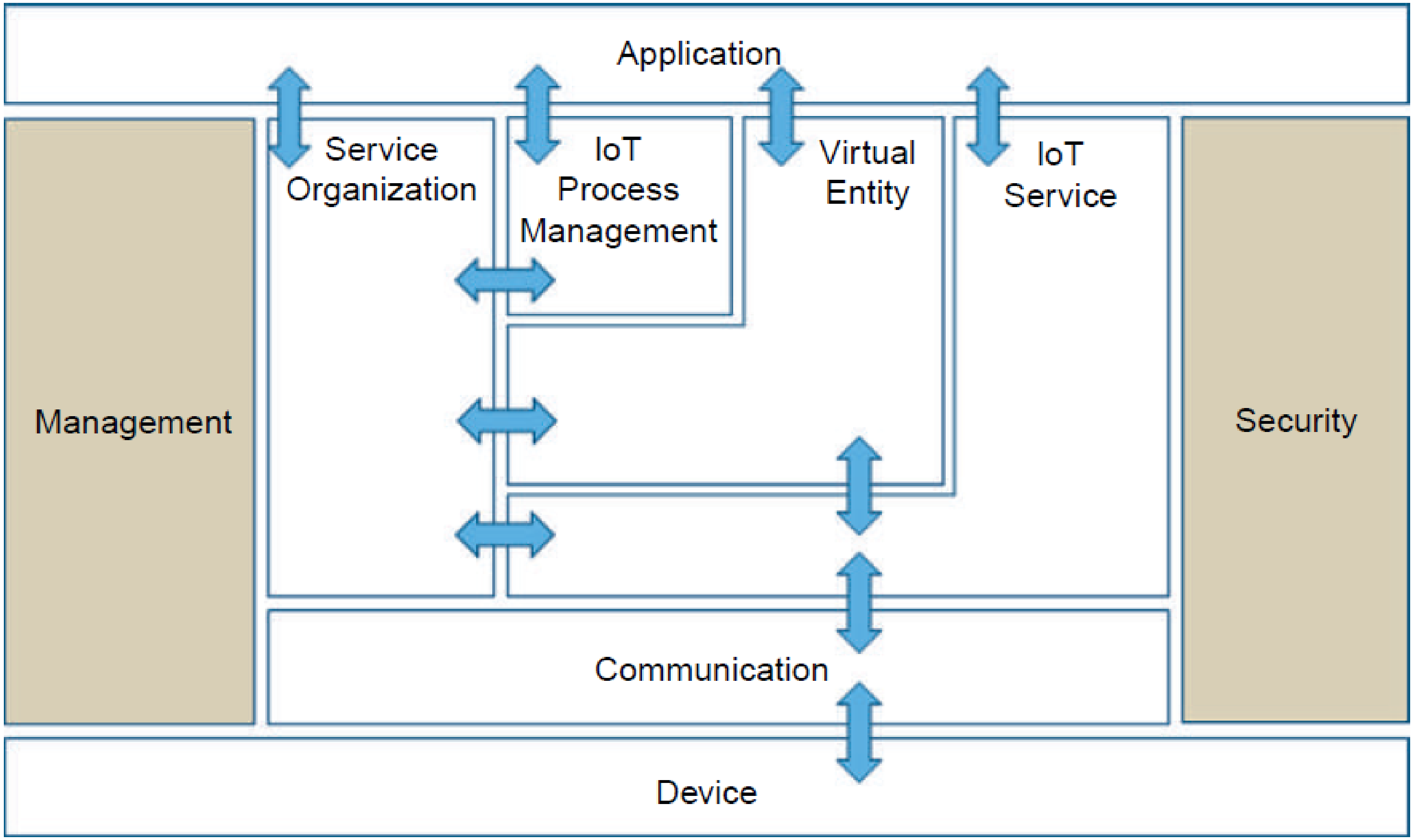


FIGURE 7.11

IoT-A Functional Model.

Functional model (contd..)

Functional Groups

Device Functional Group

- Includes all functionalities that have to be provided to all physical devices.
- Like sensing, actuation, processing, storage, identification components

Communication Functional Group

- All possible communication mechanisms like wireless or wired communication.

Functional model (contd..)

Functional Groups

IoT Service Functional Group

- Corresponds mainly to the Service class from the IoT Domain Model, and contains single IoT Services exposed by Resources hosted on Devices or in the Network

Virtual Entity Functional Group

- The Virtual Entity FG corresponds to the Virtual Entity class in the IoT Domain Model, and contains the necessary functionality to manage associations between Virtual Entities with themselves as well as associations between Virtual Entities and related IoT Services, i.e. the Association objects for the IoT Information Model.

Functional model (contd..)

Functional Groups

IoT Service Organization functional group

- A service hub between several other functional groups such as the IoT Process Management FG.
- contains functions for discovery, composition, and choreography of services

IoT Process Management functional group

The IoT Process Management FG is a collection of functionalities that allows smooth integration of IoT-related services (IoT Services, Virtual Entity Services, Composed Services) with the Enterprise (Business) Processes.

Functional model (contd..)

Functional Groups

Management functional group

Includes the necessary functions for enabling fault and performance monitoring of the system, configuration for enabling the system to be flexible to changing User demands, and accounting for enabling subsequent billing for the usage of the system.

Security functional group

The Security FG contains the functional components that ensure the secure operation of the system as well as the management of privacy.

Functional model (contd..)

Functional Groups

Application functional group

The Application FG is just a placeholder that represents all the needed logic for creating an IoT application.

Communication model

- The communication model for an IoT Reference Model consists of the identification of the endpoints of interactions, traffic patterns (e.g. unicast vs. multicast), and general properties of the underlying technologies used for enabling such interactions.
- The potential communicating endpoints or entities are the Users, Resources, and Devices from the IoT Domain Model.
- Users include Human Users and Active Digital Artifacts (Services, internal system components, external applications).
- Devices with a Human_Machine Interface mediate the interactions between a Human User and the physical world (e.g. keyboards, mice, pens, touch screens, buttons, microphones, cameras, eye tracking, and brain wave interfaces, etc.), and therefore the Human User is not a communication model endpoint.

Communication model (contd..)

- The User interactions include the User-to-Service and Service-to-Service interactions as well as the Service_Resource_Device interactions.
- The User-to-Service and Service-to-Service communication is typically based on Internet protocols and one or both Services are hosted in Service-to-Service interactions on constrained/low-end Devices such as embedded systems.
- The communication model for these interactions includes several types of gateways (e.g. network, application layer gateways) to bridge between two or more disparate communication technologies.
- The Device to host Resources or Services results in moving the corresponding Resources and/or Services out of the Device and into more powerful Devices or machines in the cloud.
- Then the Resource-to-Device or the Service-to-Resource communication needs to involve multiple types of communication stacks.

Safety, privacy, trust, security model

- An IoT system enables interactions between Human Users and Active Digital Artifacts (Machine Users) with the physical environment.
- The fact that Human Users are part of the system that could potentially harm humans if malfunctioning, or expose private information, motivates the Safety and Privacy needs for the IoT Reference Model and Architecture.

Security model (contd..)

Safety

- System safety is highly application- or application domain- specific, and is typically closely related to an IoT system that includes actuators that could potentially harm animate objects (humans, animals).
- Critical infrastructure protection is also related to safety because the loss of such infrastructure due to a malicious user attack could be detrimental to humans,
- Example: attacks to a Smart Grid could result in damages ranging from simple loss of electricity in a home to electricity loss in a hospital.
- A system designer of such critical systems typically follows an iterative process with two steps:

  (a) identification of hazards followed

  (b) the mitigation plan.

Security model (contd..)

Privacy

User privacy is of utmost importance for an IoT system.
The IoT-A Privacy Model depends on the following functional components:

a. **Identity Management-** is the derivation of several identities of different types for the same architectural entity with the objective to protect the original User identity for anonymization purposes.
b. **Authentication-** allows the verification of the identity of a User whether this is the original or some derived identity
c. **Authorization-** is the function that asserts and enforces access rights when Users (Services, Human Users) interact with Services, Resources, and Devices.
d. **Trust & Reputation-** maintain the static or dynamic trust relationships between interacting entities.

Security model (contd..)

Trust

According to the Internet Engineering Task Force (IETF) Internet Security Glossary (Shirey 2007), "***Generally, an entity is said to 'trust' a second entity when the first entity makes the assumption that the second entity will behave exactly as the first entity expects***." This definition includes an "***expectation***" which is difficult to capture in a technical context.

Security model (contd..)

Trust Model of IoT-A are:

- Trust Model Domains
- Trust Evaluation Mechanism
- Trust behaviour Policies
- Trust Anchor
- Federation of Trust

IoT Architecture Reference Model (ARM)

- The foundation of an IoT Reference Architecture description is an IoT reference model.
- A System Architecture is a communication tool for different stakeholders of the system.
- Developers, component and system managers, partners, suppliers, and customers have different views of a single system based on their requirements and their specific interactions with the system.
- The high-level abstraction is called Reference Architecture as it serves as a reference for generating concrete architectures and actual systems,

Three-layer Architecture of IoT

(i) The *perception layer* is the physical layer, which has sensors for sensing and gathering information about the environment. It senses some physical parameters or identifies other smart objects in the environment.

(ii) The *network layer* is responsible for connecting to other smart things, network devices, and servers. Its features are also used for transmitting and processing sensor data.

(iii) The *application layer* is responsible for delivering application-specific services to the user. It defines various applications in which the Internet of Things can be deployed, for example, smart homes, smart cities, and smart health.

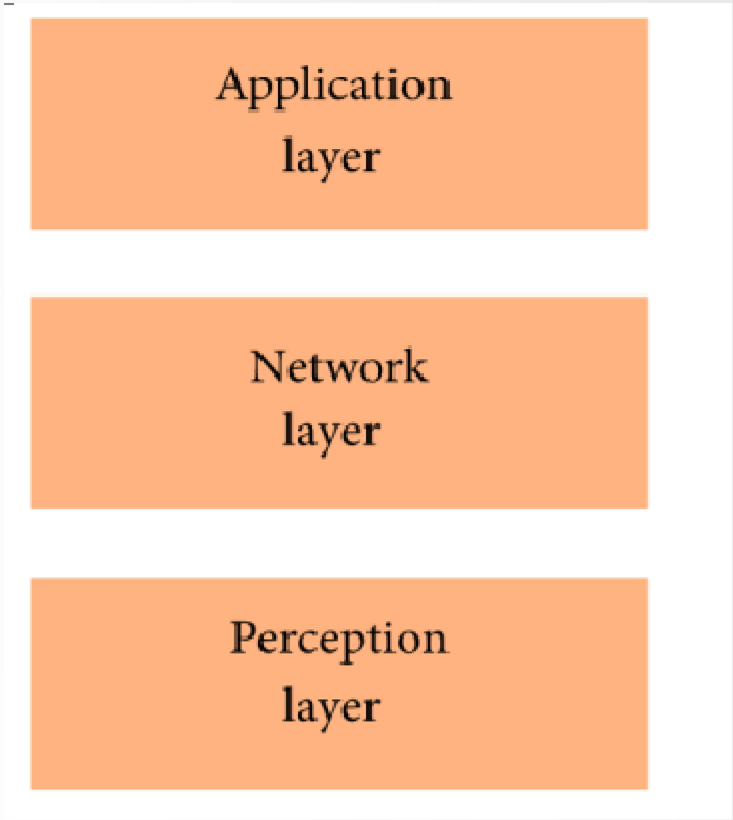


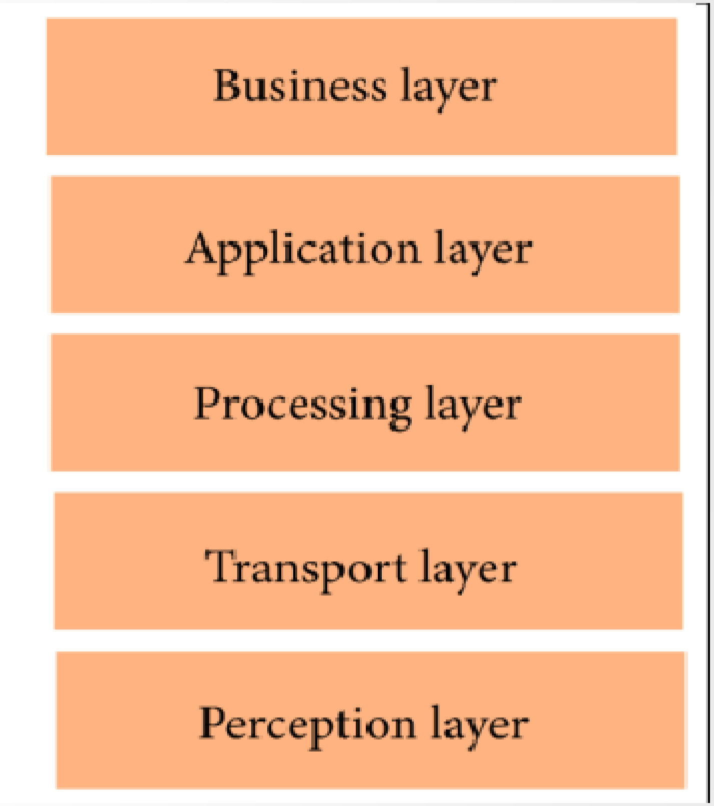
Figure: Three-layer Architecture of IoT

Five-layer Architecture of IoT

(i) The *transport layer* transfers the sensor data from the perception layer to the processing layer and vice versa through networks such as wireless, 3G, LAN, Bluetooth, RFID, and NFC.

(ii) The *processing layer* is also known as the middleware layer. It stores, analyzes, and processes huge amounts of data that comes from the transport layer. It can manage and provide a diverse set of services to the lower layers. It employs many technologies such as databases, cloud computing, and big data processing modules.

(iii) The *business layer* manages the whole IoT system, including applications, business and profit models, and users' privacy.



Business layer

Application layer

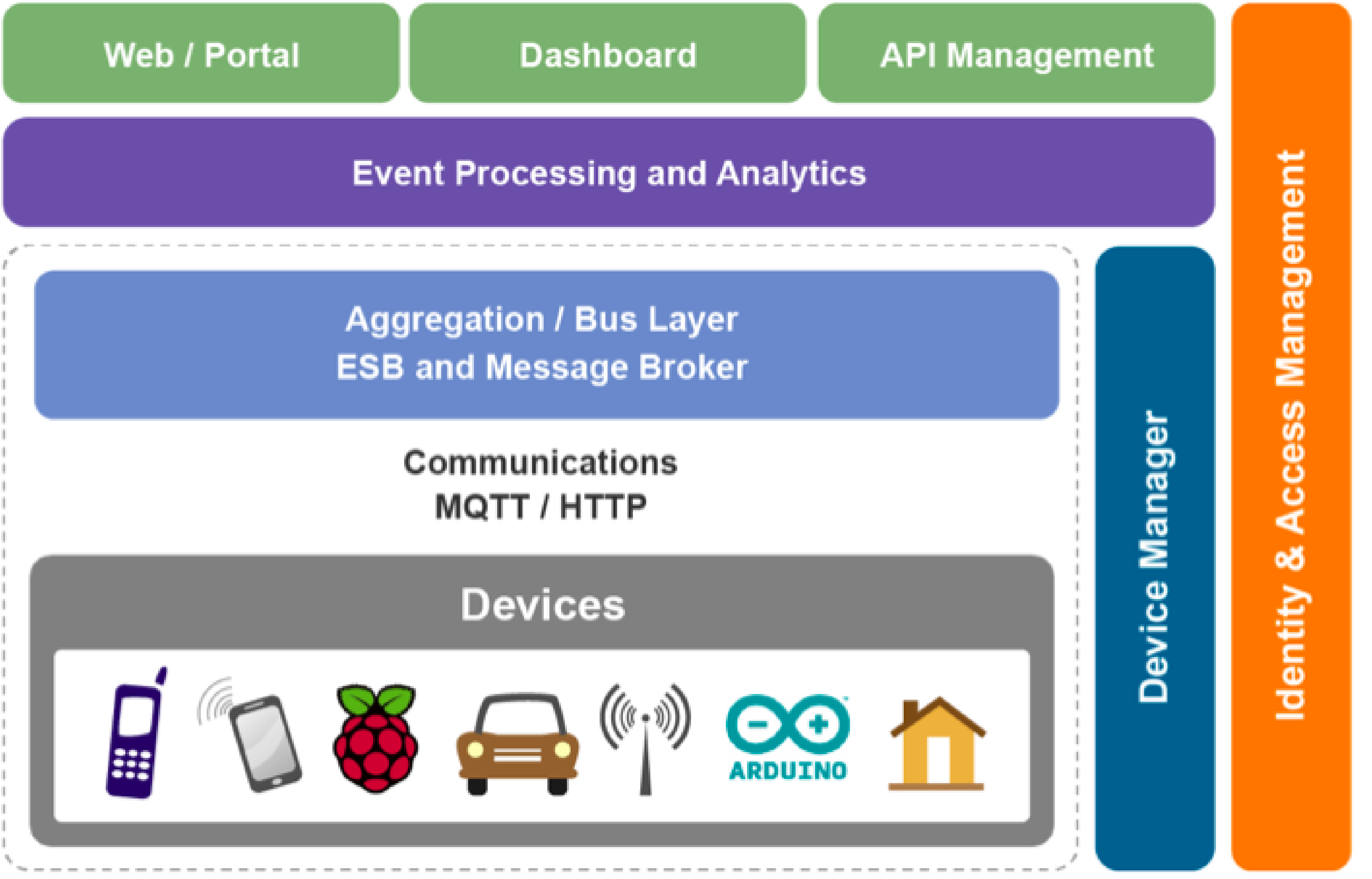
Processing layer

Transport layer

Perception layer

Figure: Five-layer Architecture of IoT

Reference Architecture of IoT



Reference Architecture of IoT (contd...)

The layers are :

- Client/external communications - Web/Portal, Dashboard, APIs
- Event processing and analytics (including data storage)
- Aggregation/bus layer – ESB and message broker
- Relevant transports - MQTT/HTTP/XMPP/CoAP/AMQP, etc.
- Devices

The cross-cutting layers are :

- Device manager
- Identity and access management

Reference Architecture of IoT (contd...)

The Device Layer

- Devices should possess a communication medium that attaches to the internet.
- Ex. Arduino with Ethernet shield, ZigBee device is ZigBee gateway
- Each device should have an identity called a Unique identifier (UUID) burnt into the device.

Communication Layer

- The communication layer supports the connectivity of the devices.
- Three potential protocols
  - HTTP/HTTPS
  - MQTT 3.1/3.1.1
  - CoAP

Reference Architecture of IoT (contd...)

Aggregation Layer

- This layer aggregates and brokers communications.
- Three abilities:
  - Support HTTP server and MQTT broker
  - Aggregate and combine data from different devices.
  - Bridge and transform between protocols

The Event Processing And Analytics Layer

- This layer takes the events from the bus and provides the ability to process and act upon these events.
- Store the data into a database.
- Approaches:
  - Highly scalable.
  - Map-reduce
  - Complex event

Reference Architecture of IoT (contd...)

Client/External Communications Layer

- Specifies how the devices have to communicate.
- Three approaches:
  - Create a web-based front end
  - Create a dashboard with analytics
  - Able to interact with systems outside the network

API management Layer

- provides a developer-focused portal where developers can find, explore, and subscribe to APIs.
- A gateway that manages access to the APIs.
- Gateway publishes data into the analytics layer where it is stored as well as processed to provide insights into how the APIs are used.

Reference Architecture of IoT (contd...)

Device Management

- A server-side system (the device manager) communicates with devices via various protocols and provides both individual and bulk control of devices.
- It also remotely manages software and applications deployed on the device.
- The device manager works in conjunction with the device management agents.

Identity and Access Management

Provides services like:
OAuth2 token issuing and validation.
identity services including SAML2 SSO and OpenID Connect.
Directory of users
Policy management for access control

Refer to..

Reference book-4: Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stamatis Karnouskos, “From Machine-to-Machine to the Internet of Things Introduction to a New Age of Intelligence”, Elsevier, 2014.

Chapters – 6 and 7

WHAT
NEXT