Chapter 2

IoT Architecture

Outline of IoT Architecture:

- State of the Art Introduction,
- Architecture Reference Model, Introduction, Reference model and architecture,
- IoT reference model, IoT Reference Architecture, Introduction,
- Functional view, Information view,
- Deployment and operational view,
- Other relevant architectural views

2.1 State of the Art

Architecture Reference Model (ARM) describes a combination of a reference model and a reference architecture. A reference model is a model that describes the main conceptual entities and how they are related to each other, while the reference architecture aims at describing the main functional components of a system as well as how the system works, how the system is deployed, what information the system processes, etc.

An ARM is a tool that establishes a common language across all the possible stakeholders of an M2M or IoT system. It can also serve as a starting point for creating concrete architectures of real systems based on stakeholder requirements, design constraints, and design principles.

2.1.1 **European Telecommunications Standards Institute M2M/oneM2M:**

- This architecture is a combination of both a functional and topological view showing some functional groups (FG) associated with physical infrastructure like M2M Devices, Gateways.
- There are two main domains, a network domain and a device and gateway domain.
- Access network is the boundary between device and gateway domain and functional groups.
- The Device and Gateway Domain contains the following functional/ topological entities:

1. **M2M Device:**
    - An M2M Device contains M2M Applications and M2M Services. E.g.device with a temperature sensor.
    - The M2M Device performs registration, authentication, authorization, management, and provision to the Network Domain.
2. **M2M Area Network:**

- This is a local area network (LAN) or a Personal Area Network (PAN) and provides connectivity between M2M Devices and M2M Gateways. Typical networking technologies are IEEE 802.15.1 (Bluetooth), IEEE 802.15.4

3. **M2M Gateway**:
   - The device that provides connectivity for M2M Devices in an M2M Area Network towards the Network Domain.
   - The M2M Gateway contains M2M Applications and M2M Service Capabilities.
   - The M2M Gateway may also provide services to other legacy devices that are not visible to the Network Domain.

The Network Domain contains the following functional/topological entities:

1. **Access Network:**

This is the network that allows the devices in the Device and Gateway Domain to communicate with the Core Network.

2. **Core Network**:

It provides the following functions:

- IP connectivity.
- Service and Network control.
- Interconnection with other networks.
- Roaming.

3. **M2M Service Capabilities:**

These are functions exposed to different M2M Applications through a set of open interfaces.

4. **M2M Applications:**

These are the specific M2M applications (e.g. smart metering) that utilize the M2M Service Capabilities through the open interfaces.

5. **Network Management Functions:**

These are all the necessary functions to manage the Access and Core Network (e.g. Provisioning, Fault Management, etc.).

6. **M2M Management Functions:**

These are the necessary functions required to manage the M2M Service Capabilities on the Network
Domain while the management of an M2M Device or Gateway is performed by specific M2M Service Capabilities. **M2M Service Bootstrap Function (MSBF):** The MSBF facilitates the bootstrapping of permanent M2M service layer security credentials in the M2M Device or Gateway and the M2M Service Capabilities in the Network Domain. **M2M Authentication Server (MAS):** This is the safe execution environment where permanent security credentials such as the M2M Root Key are stored.

2.1.2 International Telecommunication Union _ Telecommunication sector view

- The Telecommunication sector of the International Telecommunication Union (ITU-T) is doing research on IoT standardization since 2005 with the Joint Coordination Activity on Network Aspects of Identification Systems (JCA-NID),
- The ITU-T IoT domain model includes a set of physical devices that connect directly or through gateway devices to a communication network that allows them to exchange information with other devices, services, and applications.
- Application Layer of ITU-T IoT model is the host of specific IoT applications (e.g. remote patient monitoring).
- The Service & Application Support Layer consists of generic service capabilities used by all IoT applications, such as data processing and data storage, and specific service capabilities tailored to specific application domains, such as e-health or telematics.
- The Network Layer provides networking capabilities such as Mobility Management, Authentication, Authorization, and Accounting (AAA), and Transport Capabilities such as connectivity for IoT service data.
- The Device Layer includes Device Capabilities and Gateway Capabilities. The Device Capabilities include direct device interaction with the communication network and Network Layer Capabilities.
- The Gateway Device Capabilities include multiple protocol support and protocol to bridge the Network Layer capabilities and the device communication capabilities.
- In Management Capabilities, the typical FCAPS (Fault, Configuration, Accounting, Performance, Security) model of capabilities as well as device management (e.g. device) network topology management (e.g. for local and short range networks), and traffic management capabilities are included.
- In Security Capabilities, grouping of different Security Capabilities which are required by other layers are represented. The capabilities are included like message integrity/confidentiality support.

2.1.3 Internet Engineering Task Force architecture fragments:

- IETF depends on three concepts: 6LoWPAN (IPv6 over Low-power WPAN), CoRE (Constrained RESTful Environments), and ROLL (Routing Over Low power and Lossy networks).
- Application Support Layer in IETF implements Constrained Application Protocol (CoAP), which provides reliability and RESTful operation support to applications.

- The IETF CoRE working group has a draft specification for a Resource Directory. A Resource Directory is a CoAP server resource (/rd) that maintains a list of resources, their corresponding server contact information (e.g. IP addresses or fully qualified domain name, or FQDN), their type, its interface.

- A Mirror Server is a mechanism for CoAP Server resource presentations. A Mirror Server is a CoAP Server resource (/ms) that maintains a list of resources and their cached representations.

- A CoAP Server registers its resources to the Mirror Server, and upon registration a new mirror server resource is created on the Mirror Server with a container (mirror representation) for the original server representation. The original CoAP Server updates the mirror representation either periodically or when the representation changes.
- A CoAP Client retrieves the mirror representation and receives the latest updated representation from the original CoAP Server.
- The Mirror Server is useful when the CoAP Server is not always available for direct access.

2.1.4 Open Geospatial Consortium architecture

The Open Geospatial Consortium (OGC 2013) is an international industry consortium of a few hundred companies, government agencies, and universities that develops publicly available standards that provide geographical information support to the Web, and wireless and location-based services. The architecture of such system provides following functionalities:

- Designing sensor systems and observations that meet an application's criteria.
- Designing a sensor's capabilities and quality of measurements.
- Retrieval of real-time or time-series observations
- Subscription to, and publishing of, alerts to be issued by sensors or sensor services based upon certain criteria.
- SensorML and Transducer Model Language (TML), which include XML schema for describing sensor and actuator systems; e,g a system that contains a temperature sensor measuring temperature in Celsius, which also involves a process for converting this measurement to a measurement with Fahrenheit units.
- Observations and Measurements (O&M), which is an XML schema for describing the observations and measurements for a sensor
- Sensor Observation Service (SOS), which is a service for requesting, filtering, and retrieving observations and sensor system information.
- Sensor Planning Service (SPS), which is a service for applications requesting a user-defined sensor observations and measurements acquisition. This is the intermediary between the application and a sensor collection system.

Working of the system:

- A registry (CAT) maintains the descriptions of the existing OGC services, like Sensor Observation and Sensor Planning Services.
- Upon installation the sensor system, it 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 (Sensor Observation Service) and the SOS registers to the Catalog.
- A client application #1 requests from the Sensor Planning Service that the Sensor System to sense after every 10 seconds and publish the measurements using O&M.
- Another client application #2 looks up the Catalog, 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 are dispatched to the client application #2.

This architecture is more information-centric than communication-centric. The main objective of the OGC standards is to enable data, information, and service interoperability.

2.2 Architecture Reference Model:

An ARM consists of two main parts: a Reference model and a Reference Architecture. A real system may not have all the modeled entities or architectural elements described in this architecture, or it can contain other non-IoT-related entities.

Reference Model:

- A reference model describes the domain using a number of sub-models. The domain model of an architecture model captures the main concepts or entities in the domain in application.
- The domain model adds descriptions about the relationship between the concepts. These concepts and relationships serve the basis for the development of an information model because a working system needs to capture and process information about its main entities and their interactions.
- The functional model describes working system that captures and operates on the domain and information model of the same application.
- The communicating entities of communication model capture the communication or interactions of these entities.

Reference Architecture:

- A System Architecture is a communication tool for different stakeholders of the system like Developers, system managers, partners, suppliers, and customers who have different views of a single system based on their requirements and their specific interactions with the system. As a result, describing an architecture for M2M and IoT systems involves the presentation of the multiple views of the systems in order to satisfy the different stakeholders.
- The high-level abstraction is called Reference Architecture as it serves as a reference for generating concrete architectures and actual systems. A Reference Architecture captures the essential parts of an architecture, such

as design principles, guidelines, and required parts (such as entities), to monitor and interact with the physical world.

- Concrete architectures are instantiations of abstract and high-level Reference Architectures. A concrete architecture can be mapped into real world components by designing, building, engineering, and testing the different components of the actual system.
- The process of generating concrete architecture from reference architecture is iterative so that valuable feedback of stakeholders will be considered.

The figure (slide no 16) shows how to create IoT ARM based on reference architecture and how to use it to create actual IoT application.

1. **Creation of ARM:**

- The stakeholders and business scenarios are identified for specific application.
- Application specific requirements are defined.
- The requirements are extrapolated into unified requirements to implement them in the system.

2. **Application of ARM for IoT application:**

- Based on requirements and scenarios, IoT reference model and architecture is built.
- Application specific requirements and IoT reference model and architecture are used to build domain specific architecture.

2.3 IoT reference model

2.3.1 Domain Model:

A domain model defines the main concepts of a specific area of interest, here IoT. These concepts are expected to remain unchanged over the course of time, even if the details of an ARM may undergo continuous transformation or evolution over 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 and between people who work across different domains.

The domain model can be represented using UML notations.

The IoT is a support infrastructure for enabling objects and places in the physical world to have a corresponding representation in the digital world.

Example for Physical vs. Virtual World:

- Imagine the application of 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. The parking lot also includes an electronic road sign on the side of the street that shows in real-time the number of empty spots. Frequent customers also download a smart phone application that informs them about the availability of a parking spot before they even drive on the street where the parking lot is located.
- In order to realize such a service, the relevant physical objects as well as their properties need to be captured and translated to digital objects such as

variables, counters, or database objects so that software can operate on these objects and achieve the desired effect, i.e. detecting when someone parks without paying, informing drivers about the availability of parking spots, producing statistics about the average occupancy levels of the parking lot, etc. For these purposes, the parking lot as a place is instrumented with parking spot sensors (e.g. loops), and for each sensor, a digital representation is created (Parking spot #1_#16). In the digital world, a parking spot is a variable with a binary value ("available" or "occupied"). The parking lot payment station also needs to be represented in the digital world in order to check if a recently parked car owner actually paid the parking fee. Finally, the availability sign is represented to the digital world in order to allow notification to drivers that an empty lot is full for maintenance purposes, or even to allow maintenance personnel to detect when the sign is malfunctioning.

- The domain model captures the interaction between physical world and IoT. The most fundamental interaction is between a human or an application with the physical world object. Therefore, a User and a Physical Entity are two concepts that belong to the domain model. The physical interaction is the result of the intention of the human to achieve a certain goal (e.g. park the car). User can interact with the physical environment through service or application. This application is also a User in the domain model. A Physical Entity, as the model shows, can contain other physical entities. The objects, places, and things are Physical Entities as the Assets from asset layer.
- A Physical Entity is represented in the digital world as a Virtual Entity. A Virtual Entity can be a database entry, a geographical model, an image, or any other Digital Artifact.
- One Physical Entity can be represented by multiple Virtual Entities, each with a different purpose, e.g. a database entry of a parking spot denoting the spot availability, and an (empty/full) image of a parking spot on the monitor of the parking lot management system. Each Virtual Entity also has a unique identifier for making it addressable among other Digital Artifacts. A Virtual Entity representation contains several attributes that correspond to the Physical Entity current state (e.g. the parking spot availability). The Virtual Entity and the Physical entity synchronize whenever a User operates on one or the other. For example, a remotely controlled light (Physical Entity) represented by a memory location (Virtual Entity) in an application could be switched on/off by the User by changing the Virtual Entity representation, or in other words writing a value in the corresponding memory location. In this case, the real light should be turned on/off (Virtual to Physical Entity synchronization).
- The physical entities are instrumented with certain kinds of Devices. Three kinds of Device types are the most important:

1. **Sensors:**
   These are simple or complex Devices that involve a transducer that converts physical properties such as temperature into electrical signals. These Devices

perform the necessary conversion of analog electrical signals into digital signals. e.g. a voltage level to a 16-bit number,

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 possess capabilities like communication capabilities, storage of intermediate commands, processing, and conversion of digital signals to analog electrical signals.
3. **Tags:** Tags identify the Physical Entity that they are attached to. E.g. a Radio Frequency Identification (RFID) tag, barcode or Quick Response (QR) code.

IoT model offers following IoT services like:

1. **Resource-Level Services:** They provide the functionality of a Device through on-Device resources. Network Resources also provide services like location of the resource, transferring data packages etc.
2. **Virtual Entity-Level Services:** They 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: These services** are the combination of Resource-Level and Virtual Entity-Level services.

2.3.2 Information Model:

- Virtual entity is an important thing in IoT model which represents "thing" for the application.
- The IoT Information Model is represented using Unified Modeling Language (UML) diagrams. Association class is used to describe information about the specific association between a Virtual Entity and a related Service.
- IoT devices, real implementation entities will not be described in information model.
- The IoT Information Model describes Virtual Entities and their attributes. The attributes which have one or more values are shown by metadata.
- A Virtual Entity is associated with Resources to provide Services about the specific Virtual Entity. This association between a Virtual Entity and its Services is captured in the Information Model with the explicit class called Association.
- An example of an instantiation of the high-level information model for parking lot example is shown in Figure (slide no 26). Only one corresponding virtual entity is described in the figure as one parking spot. This Virtual Entity is described with one Attribute (among others) called hasOccupancy. This Attribute is associated with the Parking Lot Occupancy Service Description through the Occupancy Association. The Occupancy Association is the explicit

expression of the association (line) between the Parking Spot #1 Virtual Entity and the Parking Lot Occupancy Service. Please note that the dashed arrows with hollow arrowheads represent the relationship "is instance of" for the information model.

- **Relating Information Model and Domain Model: (slide no 27)** The Information Model captures the Virtual Entity in the Domain Model with several associated classes (Virtual Entity, Attribute, Value, MetaData, Value Container) to describe the description of a Virtual Entity and its context. The Device, Resource, and Service in the IoT Domain Model are also captured in the IoT Information Model as service description, resource description and device description classes.
- The attributes of virtual entity may contain following types of information:
    - Location and its temporal information
    - Temporal information
    - Ownership useful for commercial purpose
- The Services in the IoT Domain Model are mapped to the Service Description in the IoT Information Model. It contains information like:
    - Service Type e.g. web service, validations
    - Service area and service schedule e.g. geographical area or observation area
    - Associated resources
    - Metadata: Information required for execution of the service
- The IoT Information Model also contains Resource descriptions as follows:
    - Resource name and identifier
    - 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; and (e) a tag resource, which provides identification data for Physical Entities.
    - Indicator of whether the resource is an on-Device resource or network resource.
    - Location information about the Device that hosts this resource in case of an on-Device resource.
    - Associated Service information.
    - Associated Device description information.

2.3.3 Functional Model:

The IoT Functional Model describes the Functional Groups (FG) of IoT application and their interaction with the ARM. The Application, Virtual Entity, IoT Service, and Device FGs are from the User, Virtual Entity, Resource, Service, and Device classes from the IoT Domain Model. The Communication FG supports communicating Devices and digital artifacts.

- **Device functional group:** The Device FG contains all the possible functionality hosted by the physical Devices that are used for instrumenting the Physical Entities. This Device functionality includes sensing, actuation, processing, storage, and identification components, the sophistication of which depends on the Device capabilities.
- **Communication functional group:** The Communication FG abstracts all the possible communication mechanisms used by the relevant Devices in an actual system in order to transfer information to the digital world components or other Devices. Examples of such functions include wired bus or wireless mesh technologies through which sensor Devices are connected to Internet Gateway Devices.
- **IoT Service functional group:** The IoT Service FG corresponds mainly to the Service class from the IoT Domain Model, and contains 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.
- **IoT Service Organization functional group: The** purpose of the IoT Service Organization FG is to host all functional components that support the composition and implementation of IoT and Virtual Entity services. E.g. Simple IoT or Virtual Entity Services can be composed to create more complex services, a control loop with one Sensor Service and one Actuator service with the objective to control the temperature in a building.
- **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.
- **Management functional group:** The Management FG 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. Support functions such as management of ownership, administrative domain, rules and rights of functional components, and information stores are also included in the Management FG.
- **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. The Security FG contains components for Authentication of Users (Applications, Humans), Authorization of access to Services by Users, secure communication (ensuring integrity and confidentiality of messages) between entities of the system such as Devices,

Services, Applications, and last but not least, assurance of privacy of sensitive information relating to Human Users.

- **Application functional group:** The Application FG is just a placeholder that represents all the needed logic for creating an IoT application. The applications typically contain custom logic tailored to a specific domain such as a Smart Grid. An application can also be a part of a bigger ICT system that employs IoT services such as a supply chain system that uses RFID readers to track the movement of goods within a factory in order to update the Enterprise Resource Planning (ERP) system.

2.4 IoT Reference Architecture:

IoT 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. IoT reference architecture is typically presented as a series of views that address different stakeholder concerns.

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 (e.g. city authorities).

2.4.1 Functional view

It consists of the Functional Groups (FGs) presented earlier in the IoT Functional Model, each of which includes a set of Functional Components (FCs).

Device and Application functional group:

The Device and Application FGs are same as IoT Functional Model. For convenience 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.

The Network FC is responsible for message routing & forwarding and the necessary translations of various identifiers and 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. For a Sensor type of a Resource, the IoT Service FC includes Services that receive requests from a User and returns the Sensor Resource value in synchronous or asynchronous fashion.

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. The Virtual Entity Service FC enables the interaction between Users and Virtual Entities by means of reading and writing the Virtual Entity attributes.

IoT process management functional group:

The IoT Process Management FG aims at supporting the integration of business processes with IoT-related services. It consists of two FCs:
• The Process Modeling FC provides that right tools for modeling a business process that utilizes IoT-related services.
• The Process Execution FC contains the execution environment of the process models created by the Process Modelling FC and executes the created processes by utilizing the Service Organization FG in order to resolve high-level application requirements to specific IoT services.

Service Organization functional group:

The Service Organization FG acts as a coordinator between different Services offered by the system. It consists of the following FCs:

- The Service Composition FC manages the descriptions and execution environment of complex services consisting of simpler dependent services.
- The Service Orchestration FC resolves the requests coming from IoT Process Execution FC or User into the concrete IoT services that fulfill the requirements.
- The Service Choreography FC is a broker for facilitating communication among Services using the Publish/Subscribe pattern.

Security functional group:

The Security FG contains the necessary functions for ensuring the security and privacy of an IoT system. It consists of the following FCs:

- The Identity Management FC manages the different identities of the involved Services or Users in an IoT system
- The Authentication FC verifies the identity of a User.
- The Authorization FC manages and enforces access control policies.
- The Key Exchange & Management is used for setting up the necessary security keys between two communicating entities in an IoT system.
- The Trust & Reputation FC manages reputation scores of different interacting entities in an IoT system and calculates the service trust levels.

Management functional group:

The Management FG contains system-wide management functions that may use individual FC management interfaces. It is not responsible for the management of each component, rather for the management of the system as a whole. It consists of the following FCs:

- The Configuration FC maintains the configuration of the FCs and the Devices in an IoT system
- The Fault FC detects, logs, isolates, and corrects system-wide faults if possible.
- The Member FC manages membership information about the relevant entities in an IoT system.
- The State FC is similar to the Configuration FC, and collects and logs state information from the current FCs, which can be used for fault diagnosis, performance analysis and prediction, as well as billing purposes.
- The Reporting FC is responsible for producing compressed reports about the system state based on input from FCs.

2.4.2 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. It is represented by the information lifecycle and flow along with the information handling components.

Information description

The pieces of information handled by an IoT system is as follows:

- Virtual Entity context information
- IoT Service output
- Virtual Entity descriptions
- Associations between Virtual Entities and related IoT Services.
- IoT Service Descriptions, which contain associated Resources, interface descriptions, etc.
- Resource Descriptions, which contain the type of resource (e.g. sensor), identity, associated Services, and Devices.
- Device Descriptions such as device capabilities (e.g. sensors, radios).
- Descriptions of Composed Services, which contain the model of how a complex service is composed of simpler services.
- IoT Business Process Model, which describes the steps of a business process utilizing other IoT-related services (IoT, Virtual Entity, Composed Services).
- Security information such as keys, identity pools, policies, trust models, reputation scores, etc.
- Management information such as state information, fault/performance purposes, configuration snapshots, reports, membership information, etc.