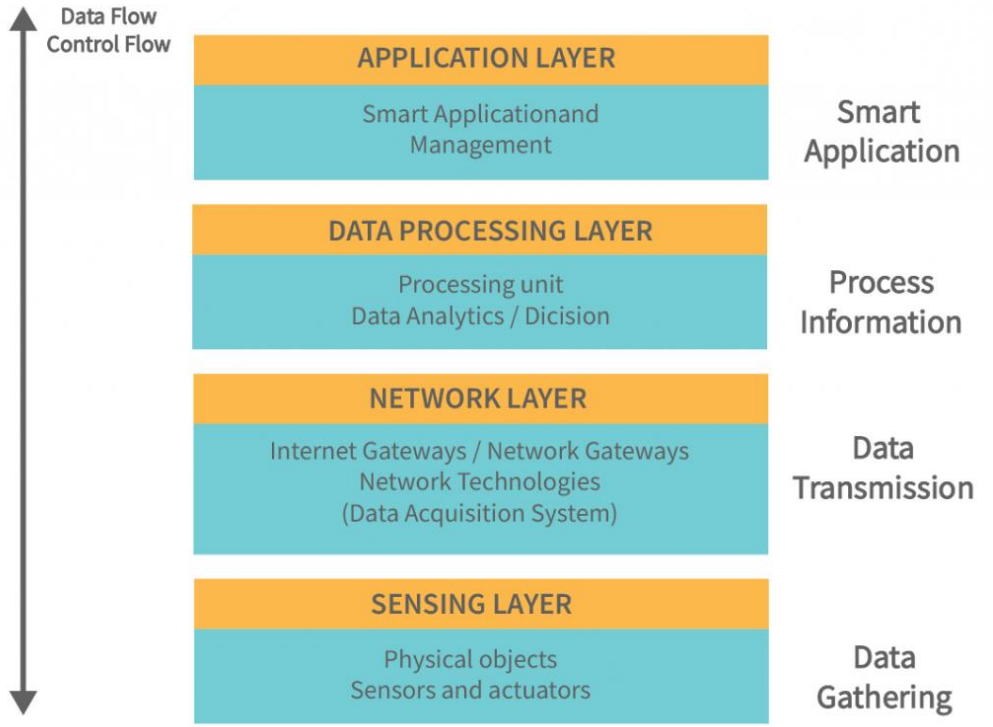
UNIT-2

IOT ARCHITECTURE & CHALLENGES

2.1 ARCHITECTURE

Depending upon different application areas of Internet of Things, it works accordingly as per it has been designed/developed. But it does not have any standard defined architecture of working which is strictly followed universally. The architecture of IoT depends upon its functionality and implementation in different sectors. A four-layer architecture is the standard and most widely accepted format.



1) Sensing Layer

The sensing layer is the first layer of the IoT architecture and is responsible for collecting data from different sources.

This layer includes sensors and actuators that are placed in the environment to gather information about temperature, humidity, light, sound and other physical parameters.

These devices are connected to the network layer through wired or wireless communication protocols.

2) Network Layer

The network layer of an IoT architecture is responsible for providing communication and connectivity between devices in the IoT system.

It includes protocols and technologies that enable devices to connect and communicate with each other and with the wider internet.

Examples of network technologies that are commonly used in IoT include Wi-Fi, Bluetooth, Zigbee and cellular networks such as 4G and 5G.

Additionally, the network layer may include gateways and routers that act as intermediaries between devices and the wider internet and may also include security features such as encryption and authentication to protect against unauthorized access.

3) Data Processing Layer

The data processing layer of IoT architecture refers to the software and hardware components that are responsible for collecting, analysing and interpreting data from IoT devices.

This layer is responsible for receiving raw data from the devices, processing it and making it available for further analysis or action.

The data processing layer includes a variety of technologies and tools such as data management systems, analytics platforms and machine learning algorithms. These tools are used to extract meaningful insights from the data and make decisions based on that data.

Example of a technology used in the data processing layer is a data lake, which is a centralized repository for storing raw data from IoT devices.

4) Application Layer

The application layer of IoT architecture is the topmost layer that interacts directly with the end-user. It is responsible for providing user-friendly interfaces and functionalities that enable users to access and control IoT devices.

This layer includes various software and applications such as mobile apps, web portals and other user interfaces that are designed to interact with the underlying IoT infrastructure.

It also includes middleware services that allow different IoT devices and systems to communicate and share data seamlessly.

The application layer also includes analytics and processing capabilities that allow data to be analysed and transformed into meaningful insights. This can include machine learning algorithms, data visualization tools and other advanced analytics capabilities.

2.2 IOT REFERENCE MODEL & ARCHITECTURE

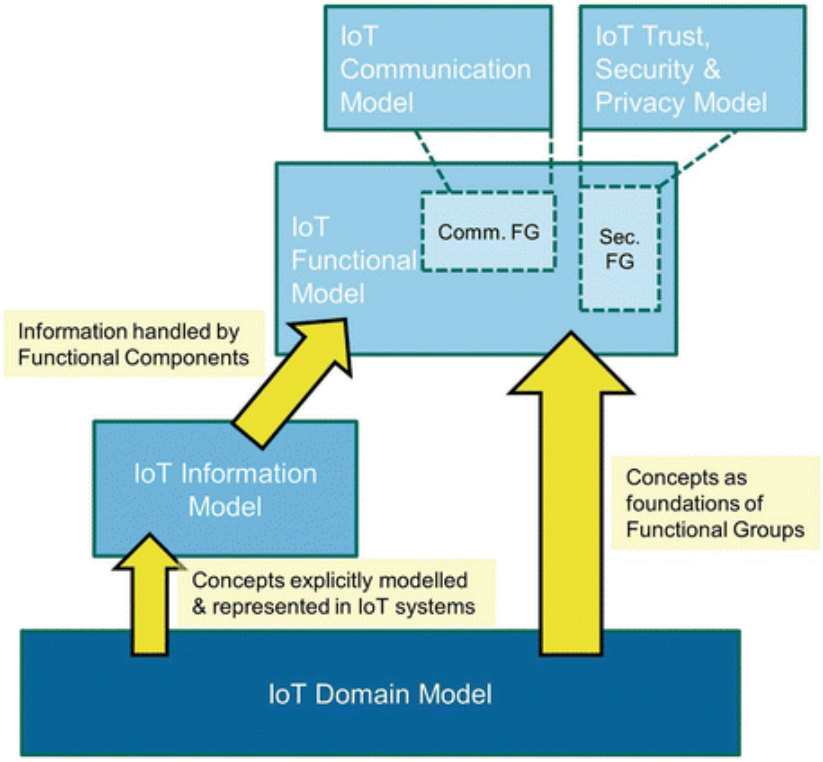
An ARM (Architectural Reference Model) consists of two main parts:

1) **Reference Model**
2) **Reference Architecture.**

Reference Model describes the domain using a number of sub-models.

Reference Architecture is used to describe essential building blocks and identify design choices to deal with confliction requirements.

Reference Model:



The Reference Model contains:

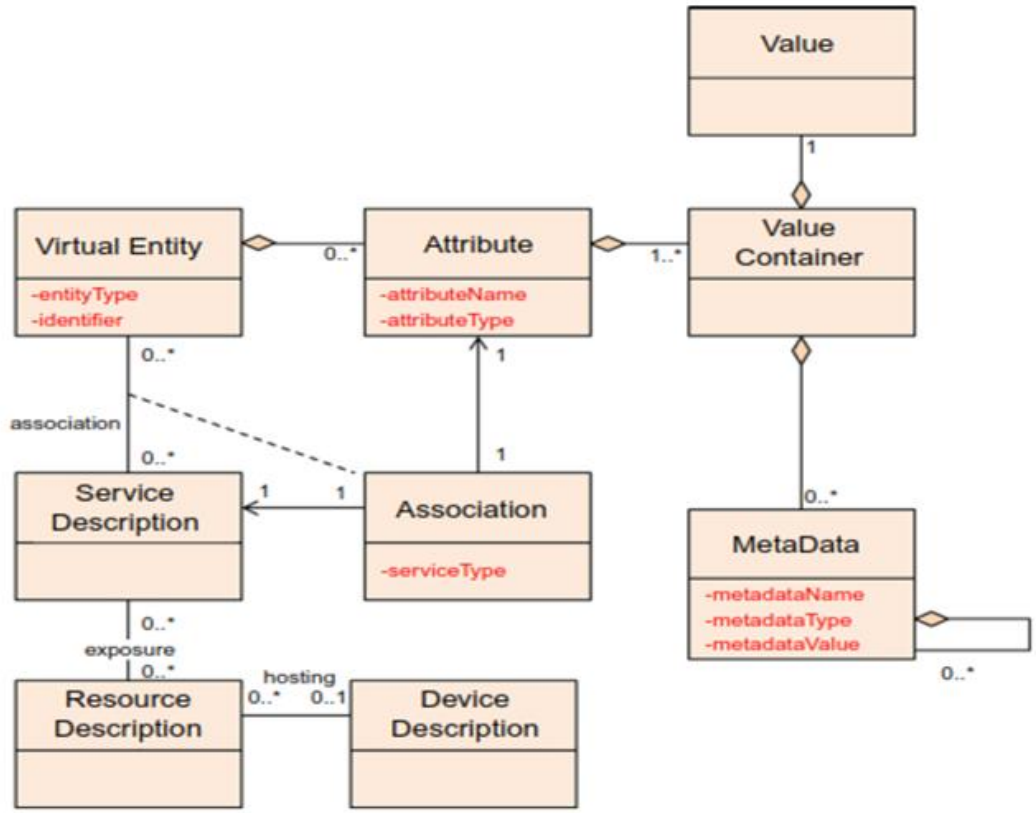
1) **Domain Model**
2) **Informational Model**
3) **Functional Model**
4) **Communication Model**
5) **Trust, Security and Privacy Model**

Domain Model:

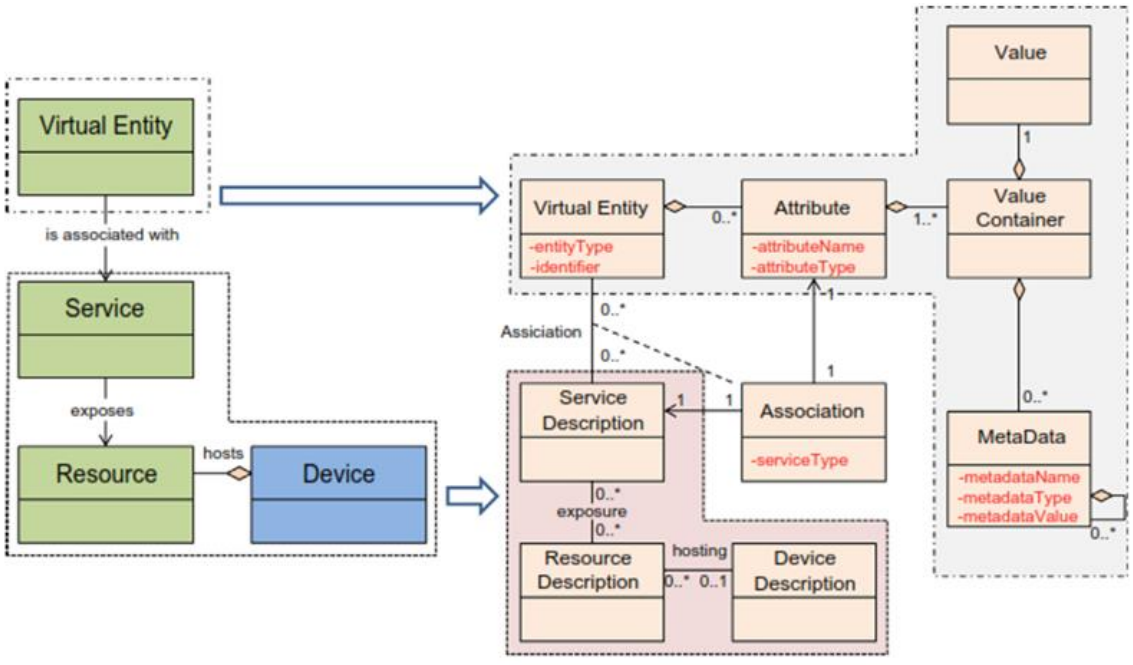
It defines the main concepts of the IoT and the relation between these concepts.

Information Model:

It models Domain Model concepts that are to be explicitly represented and manipulated in the digital world. In addition, the IM explicitly models relations between these concepts. The Information Model is a meta model that provides a structure for the information. This structure provides the basis for defining the functional interfaces.



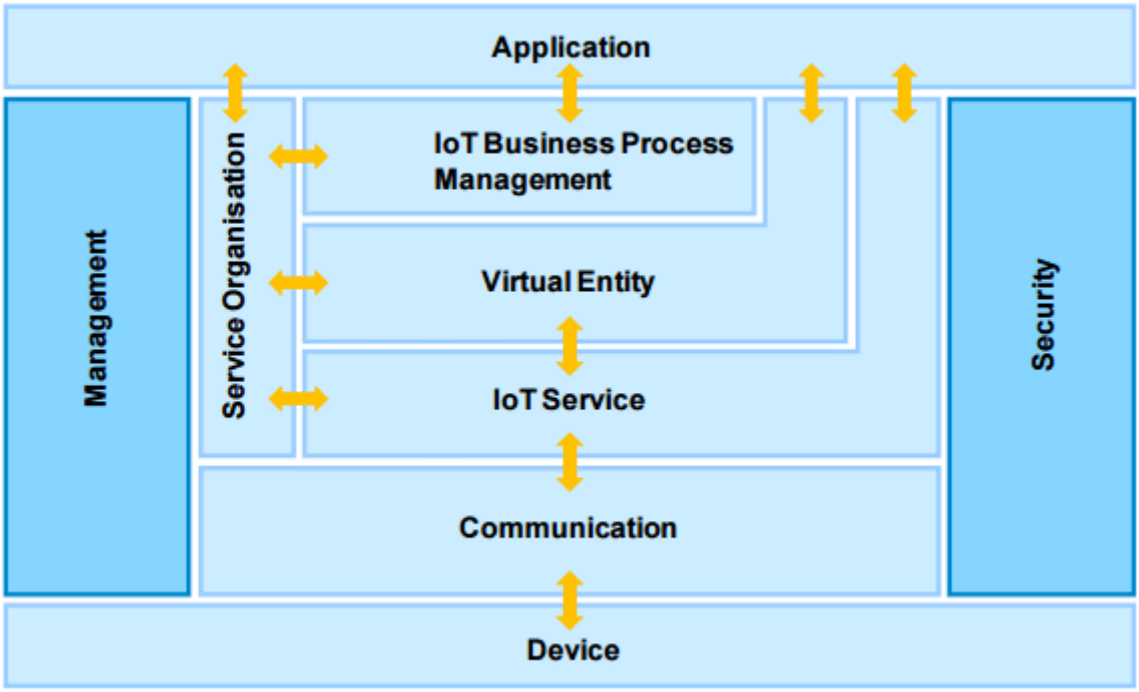
(Figure: High Level IoT Information Model)



(Figure: Relationship between core concepts of IoT Domain Model and IoT Information Model)

Functional Model:

It is derived from internal and external requirements. It is closely related to the Functional View of the Reference Architecture. 7 Functional Groups are strongly related to Domain model, Communication Model, Security Model.



(Figure: Functional Model)

Communication Model:

The IoT Communication Model introduces concepts for handling the complexity of communication in heterogeneous IoT environments. Communication also constitutes one FG in the IoT Functional Model.

Privacy Model:

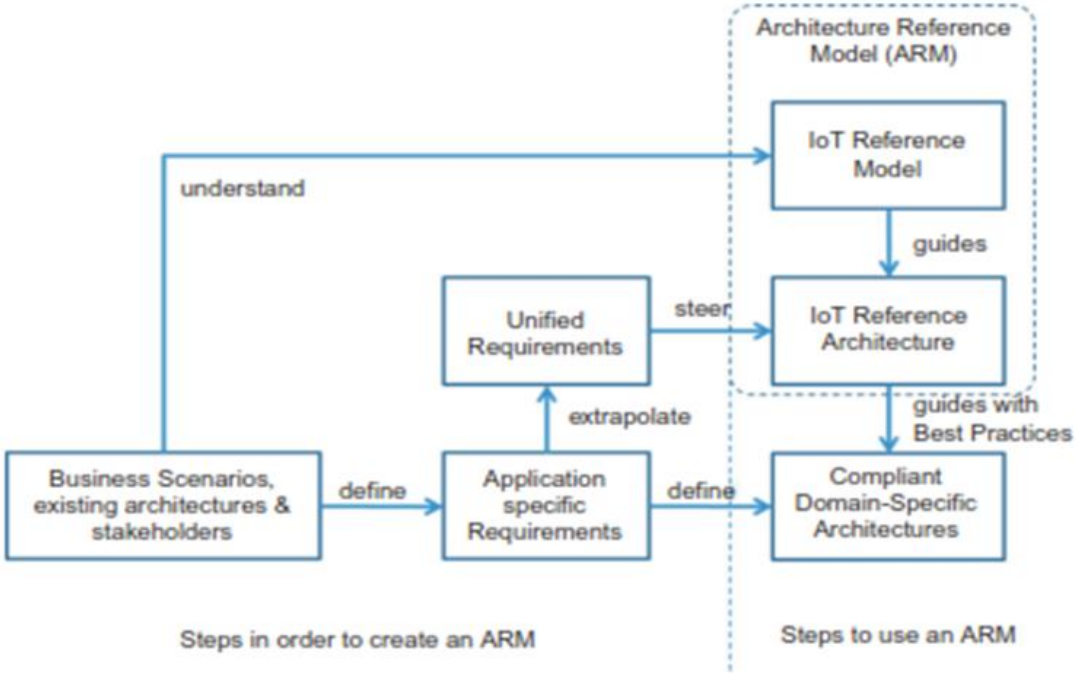
Because interactions with the physical world may often include humans, protecting the user privacy is of utmost importance for an IoT system. The Privacy Model depends on the functional components such as Identity Management, Authentication, Authorisation and Trust & Reputation.

Trust Model:

Generally, an entity is said to 'trust' a second entity when the first entity assumes that the second entity will behave exactly as the first entity expects.

Security Model:

The Security Model of IoT consists of communication security that focuses mostly on the confidentiality and integrity protection of interacting entities and functional components such as Identity Management, Authentication, Authorisation and Trust & Reputation.



(Figure: IOT Reference Architecture and Reference Model Dependency)

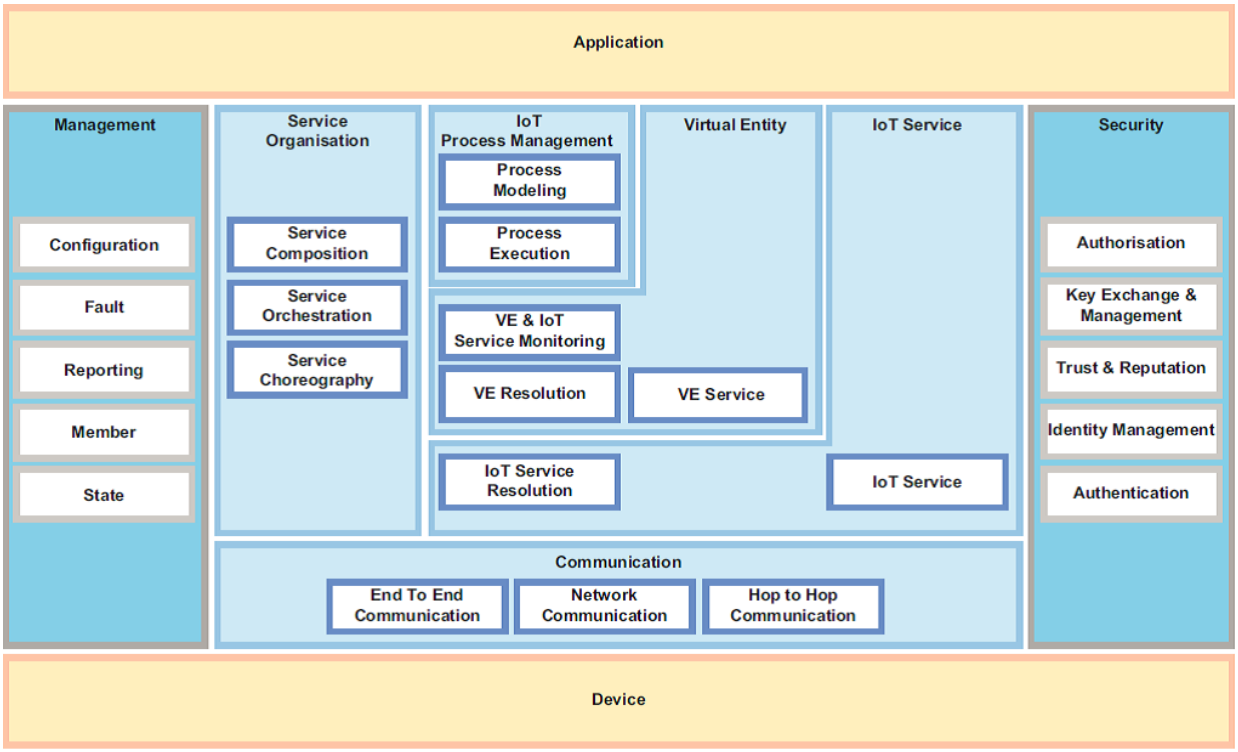
2.3 FUNCTIONAL, INFORMATION, DEPLOYMENT & OPERATIONAL VIEW

IoT Reference Architecture is presented as a set of architectural views which are listed below:

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

Views are useful for reducing complexity of Reference Architecture.

Functional View:



- **Device and Application Functional Group**
  The device functional group includes sensing, actuators, processing, storage and identification components, the sophistication of which depends on the device's capabilities. Application function group contains the standalone application.
- **Communication Functional Group**
  It contains the components for end to end communication, network communication and hop-by-hop communication.
- **IoT Service Functional Group**
  It 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 functional group contains functions that support the interactions between users and physical things through virtual entity service.
- **Process Management Functional Group**
  It is a collection of functionalities that allows smooth integration of IOT related Services with the business process.
- **Service Organization Functional Group**
  It acts as a communication hub between several other functional groups by composing and orchestrating services of different level of abstraction.
- **Security Functional Group**
  It is responsible for security and privacy matters in IOT-A compliant IOT systems.
- **Management Functional Group**
  It is responsible for composition and tracking of actions that involve in the other functional groups.

Information View:

Information view consists of the description of information handled by an IoT system and the way this information is handled in the system i.e. information lifecycle and flow (how information is created, processed and deleted) and information handling components.

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).
- 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:** It 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.

Deployment and Operational View:

Deployment view and Operational view are very important in addressing how the actual system can be realized by selecting technologies and making them communicate and operate in a comprehensive way.

This view is dependent on actual use case and requirements.

Let's take an example of the parking lot system

- There are two sensor nodes #1 and #2, each of which is connected to eight car presence sensors.
- They are also connected to the payment stations through wireless or wired communication
- The payment station acts both as a user interface for the device to pay and get a payment receipt as well as a communication gateway that connects the two sensor nodes and payment interface physical devices with the internet through WAN.
- The occupation sign also acts as a communication gateway for the actuator node, and we assume that because of the deployment, a direct connection to the payment station is not feasible.
- The physical gateway devices connect through WAN to the internet and towards a data centre where the parking lot management system software is hosted as one of the virtual machines on a platform as service configurations.
- The two main application connected to this management system are human user mobile phone applications and parking operation centre applications.

2.4 SIZE & SPACE CONSIDERATIONS IN IOT

Size and Space matter a lot in IoT systems. Today's consumers want everything to be as compact and small as possible. When we consider size and space, there arise two perspectives:

1) **IoT Device Perspective**
2) **IoT Data Perspective**

From the IoT device perspective, its size should be as small as possible so that it occupies less space. Further, an IoT device may not be always used as a standalone single device. In most cases, it may be used as a constituent of some other device/system. In such situations also, it is desired to have a small-sized IoT device.

From the IoT data perspective, the space for IoT data storage should be large enough to accommodate the constant streaming of data. The IoT data will be very large over time. So, the space consideration for data storage should be given enough and equal priority in the overall IoT system. For example, IoT wearables are often small in size; however, they need to be equipped with high performance batteries to perform various types of tasks.

Rural or outdoor areas generally have a large range where the signals need to cover a longer distance to reach to the server or the wireless gateway. This kind of set up requires a GPS or a cellular interface. If the transmission distance is very large, it will require high frequencies as well as high power. If the location is a remote one, then the life of the battery becomes a very important factor.

The operational distance can also be affected by interference with some physical obstacles or with other RF devices. Manufacturers also have many options available in hardware and software relative to the network technology for IoT-based products.

Some IoT devices can be connected directly to the Internet. They may use Ethernet or Wi-Fi. There may be other products that use wireless technologies. But they all will be requiring a 'gateway' to convert their network technology to either Wi-Fi or Ethernet.

Some networks will require antennas and connectors. This requirement will also add to their size. It is a fact that our physical environment is very diverse. This diversity also leads to a similar diverse IoT-based application space, which includes tiny implantable heart rate monitors, oil reservoir diagnosis sensors, HVAC (Heating, Ventilation and Air Conditioning) sensors that have an extremely long lifetime, etc.

2.5 SCALABILITY AND COMPATIBILITY BETWEEN DIFFERENT SMART SENSORS

Introduction to Smart Sensor

A smart sensor is a device that takes input from the physical environment and uses built-in compute resources to perform predefined functions upon detection of specific input and then process data before passing it on.

The smart sensor is also a crucial and integral element in the IoT. IoT technology makes it possible to provide a unique identifier for almost anything and to transmit data from or about those things over the internet or a similar sensor network.

A smart sensor is generally made of a sensor, a microprocessor and wireless communication technology of some kind.

A smart sensor might also include several other components besides the primary sensor. These components can include transducers, amplifiers, excitation control, analog filters, analog-to-digital converters and compensation that provides a built-in correction of less-than-ideal measurements or output. A smart sensor also incorporates software-defined elements that provide functions such as data conversion, digital processing and communication to external devices.

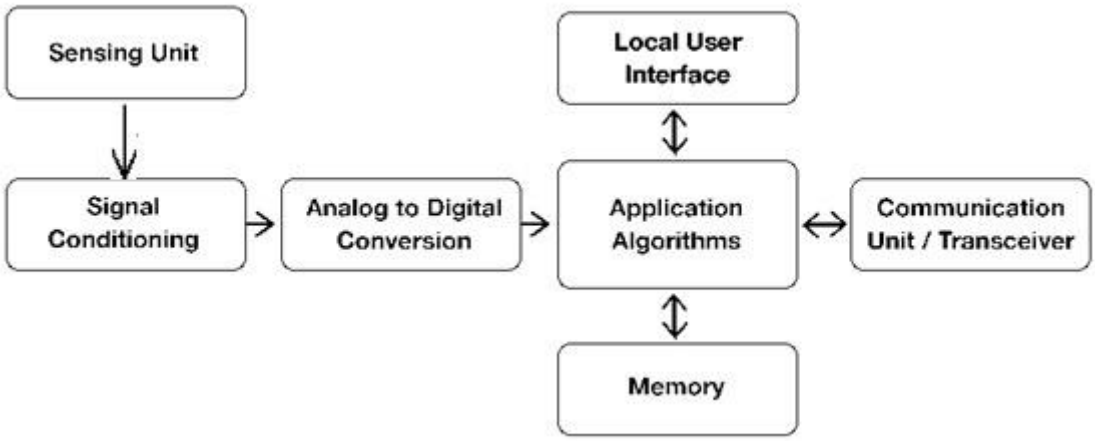
Smart sensors are used mainly for monitoring & control mechanisms in different environments like water level & food monitoring systems, smart grids, traffic monitoring & control, environmental monitoring, conserving energy in artificial lighting, monitoring of the remote system and fault diagnostics of equipment, transport & logistics, agriculture, telecommunications, industrial applications, animal tracking, etc.

Some of the most commonly used smart sensors are listed as below:

1) Temperature Sensors
2) Pressure Sensors
3) Humidity Sensors
4) Infrared Sensors
5) Proximity Sensors
6) Water Quality Sensors
7) Chemical Sensors
8) Gas Sensors
9) Level Sensors
10) Smoke Sensors
11) Optical Sensors
12) Motion Detection Sensors
13) Image Sensors

Smart Sensor Block Diagram

The block diagram of the smart sensor is shown below. This block diagram includes different blocks like sensing unit, signal conditioning, analog to digital conversion, application algorithms, local user interface, memory and communication unit or transceiver.



- **Sensing Unit**

  This unit detects the changes in physical parameters & generates electrical signals equivalent to it.

- **Signal Conditioning Unit**

  The signal conditioning unit controls the signal to meet the necessities of next-level operations without losing data.

- **Analog to Digital Converter**

  ADC converts the signal from analog to digital format & sends it to the microprocessor.

- **Local User Interface**

  The local user interface or LUI is a panel-mounted device used to allow building operators to monitor & control system equipment.

- **Application Algorithm**

  The signals from smart sensors reach here & process the received data based on the application programs previously loaded here & generate output signals.

- **Memory**

  It is used to store media for saving received & processed data.

- **Communication Unit**

  The output signals from the application algorithm or microprocessor are transmitted to the main station through the communication unit. This unit also gets command requirements from the key station to execute specific tasks.

Advantages

1) These are small in size.
2) These sensors are very easy to use, design & maintain.
3) The performance level is higher.
4) Speed of communication & reliability is higher due to the direct conversion with the processor.
5) These sensors can perform self-calibration & self-assessments.
6) These sensors can notice issues like switch failures, open coils & sensor contamination.
7) These sensors optimize manufacturing processes easily that need changes.
8) They can store many systems' data.

Disadvantages

1) Smart sensors' reliability is one of the major drawbacks because if they are stolen or get damaged then they can affect a lot of systems badly.
2) It needs both sensors & actuators.
3) Sensor calibration has to be managed by an external processor.
4) High complexity in wired smart sensors, so the cost is also very high.

How are Smart Sensors different from Base Sensors?

Smart sensors include an embedded Digital Motion Processor (DMP), whereas base sensors don't include one. A DMP is a microprocessor that's integrated into the sensor. It lets the sensor perform onboard processing of the sensor data. This might mean normalizing the data, filtering noise from electrical signals or performing other types of signal conditioning. In any case, a smart sensor performs data conversion digital processing prior to any communication to external devices.

A base sensor is simply a sensor that isn't equipped with a DMP or other compute resources that would let it process data. Whereas a smart sensor produces output that is ready to use, a base sensor's output is raw and must typically be converted into a usable format.

Smart sensors are generally preferred over base sensors because they include native processing capabilities. Even so, there are situations where it might be more advantageous to use a base sensor. If an engineer is designing a device and needs complete control over sensor input, then a base sensor would be preferable. Base sensors also cost less than smart sensors because they contain fewer component.