Smart Objects: The Things in IOT

Sensors, Actuators and Smart objects

1.1 What is a sensor?

Sensor is a device which is able to detect changes in an environment and respond to environment.

Sensor converts physical attribute to electrical signal which is then converted to digital or analog signal which then sent to another device for transformation into useful data that can be consumed by intelligent devices or humans. example: light sensors, thermostat sensor, pressure sensors etc.

There are several ways to group and cluster sensors into different categories include the following:

Active or passive: Sensors can be categorized based on whether they produce an energy output and typically require an external power supply (active) or whether they simply receive energy and typically require no external power supply (passive).

Invasive or non-invasive: Sensors can be categorized based on whether a sensor is part of the environment it is measuring (invasive) or external to it (non-invasive).

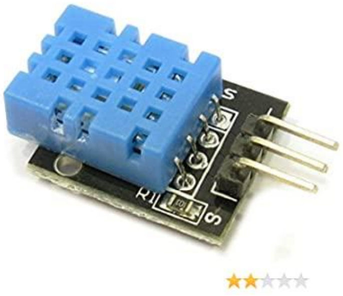
Contact or no-contact: Sensors can be categorized based on whether they require physical contact with what they are measuring (contact) or not (no-contact).

Absolute or relative: Sensors can be categorized based on whether they measure on an absolute scale (absolute) or based on a difference with a fixed or variable reference value (relative).

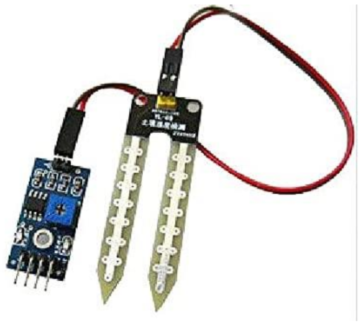
Area of application: Sensors can be categorized based on the specific industry or vertical where they are being used.

How sensors measure: Sensors can be categorized based on the physical mechanism used to measure sensory input (for example, thermoelectric, electrochemical, piezoresistive, optic, electric, fluid mechanic, photo elastic).

What sensors measure: Sensors can be categorized based on their applications or what physical variables they measure.



Temperature and humidity sensor

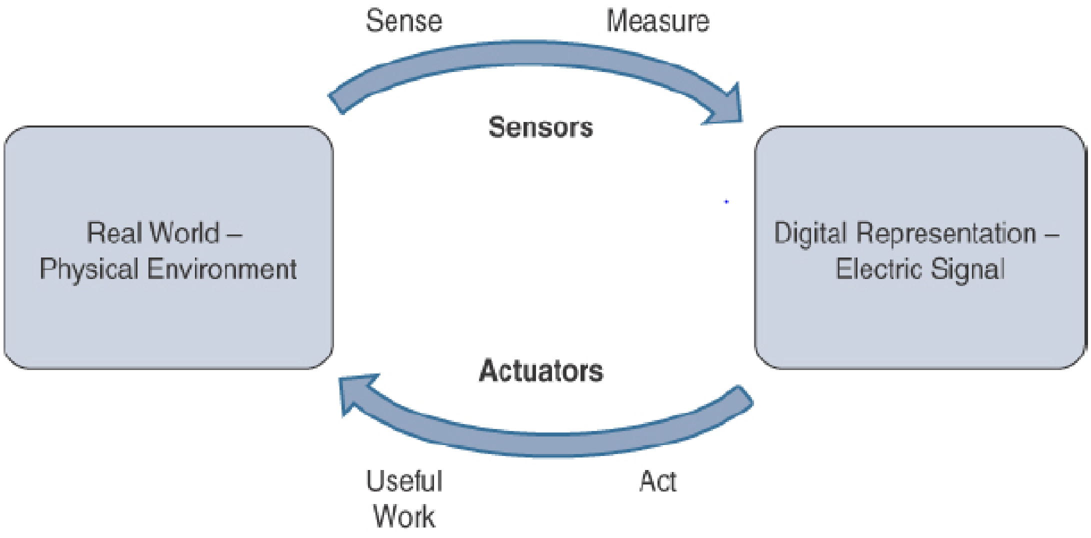


Soil Moisture Sensor

1.2 Difference between sensor and actuator?

Sensors convert physical attribute to electrical signal like it sense temperature by measuring heat and converts into electrical signal.

Where an **actuator** converts electrical signal into physical action by keeping some temperature value as threshold, if temperature value goes beyond threshold then motor should on, here motor becomes actuator.
They are devices which transform an input signal (mainly an electrical signal) into some form of motion.



Actuators are classified based on

Type of motion

Actuators can be classified based on the type of motion they produce (for example, linear, rotary, one/two/three-axes).

Power

Actuators can be classified based on their power output (for example, high power, low power, micro power)

Binary or continuous

Actuators can be classified based on the number of stable-state outputs

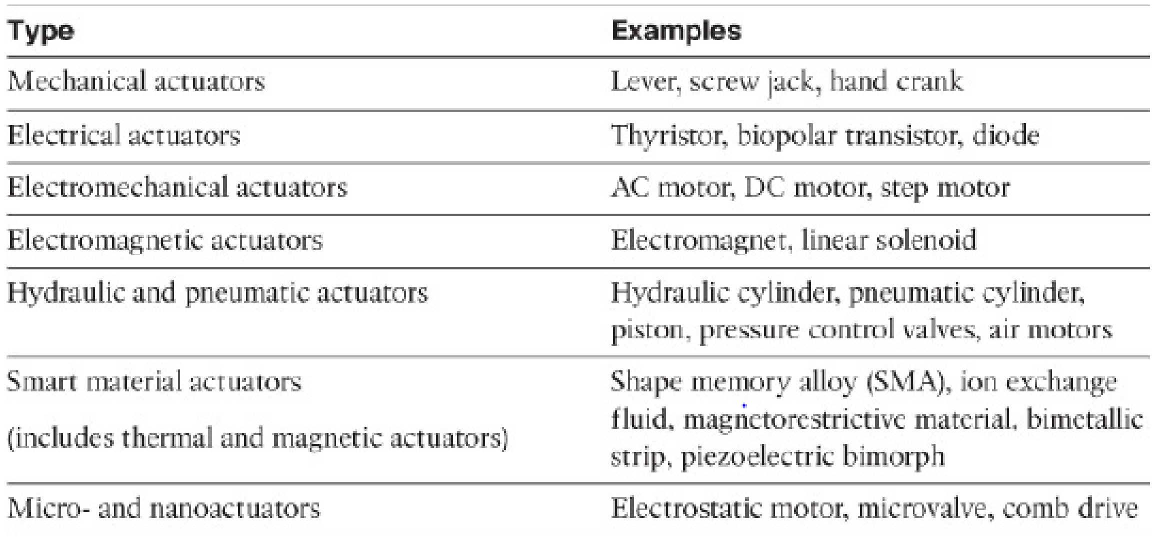
Area of application

Actuators can be classified based on the specific industry or vertical where they are used.

Type of energy

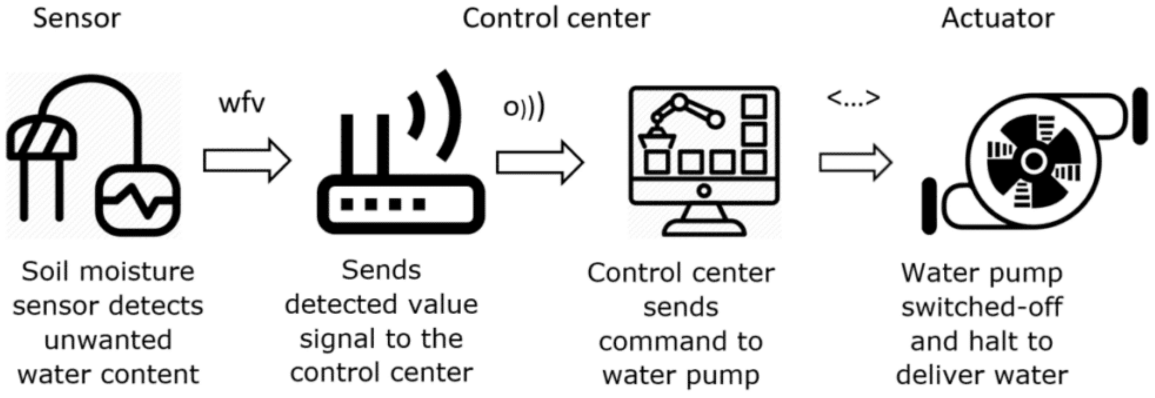
Actuators can be classified based on their energy type.

Below table shows classification of actuators base on energy



Type	Examples
Mechanical actuators	Lever, screw jack, hand crank
Electrical actuators	Thyristor, biopolar transistor, diode
Electromechanical actuators	AC motor, DC motor, step motor
Electromagnetic actuators	Electromagnet, linear solenoid
Hydraulic and pneumatic actuators	Hydraulic cylinder, pneumatic cylinder, piston, pressure control valves, air motors
Smart material actuators (includes thermal and magnetic actuators)	Shape memory alloy (SMA), ion exchange fluid, magnetorestrictive material, bimetallic strip, piezoelectric bimorph
Micro- and nanoactuators	Electrostatic motor, microvalve, comb drive

While sensors give the data, actuators give the activity. The most fascinating use cases for IoT are those where sensors and actuators cooperate in an insightful, key, and correlative style. This incredible mix can be utilized to take care of ordinary issues by just lifting the information that sensors give to noteworthy understanding that can be acted on by work-delivering actuators.


Sensor
Control center
Actuator
wfv
o)))
<...>
Soil moisture sensor detects unwanted water content
Sends detected value signal to the control center
Control center sends command to water pump
Water pump switched-off and halt to deliver water

Example:

Soil moisture sensor detects moisture content in soil and sensor to microcontroller which is control center where threshold value is set. If soil moisture value is above the threshold value then automatically water pump should on .

Here water pump is **Actuator**

MEMS-Micro-Electro-Mechanical system

- MEMS are micro machines can integrate and combine electric and mechanical elements such as machines.
- **It is a technology used to do small integrated devices or systems that is a combination of mechanical and electrical components**.
- One of the keys to this technology is **a microfabrication technique that is similar to what is used for microelectronic integrated circuits.**
- This approach allows mass production at very low costs. The combination of tiny size, low cost, and the ability to produce makes MEMS an attractive option for a huge number of IoT applications.
- Smart phones also use MEMS technologies for things like accelerometers and gyroscopes. In fact, automobiles were among the first to commercially introduce MEMS into the mass market, with airbag accelerometers.
- Size can vary from a few micrometers to millimeters.
- MEMS uses batch processing technologies.
- Sensors such as MEMS accelerators, MEMS pressure sensors,tilt sensor and other types of sensors
- Actuators such as MEMS switches, micro pumps, micro-levels and micro-grippers.

1.3 Smart objects:

- A smart object is an object that enhances the interaction with not only people but also with other smart objects.
- Also known as smart connected products or smart connected things (SCoT), they are products, assets and other things embedded with processors, sensors,

software and connectivity that allow data to be exchanged between the product and its environment, manufacturer, operator/user, and other products and systems.(wiki)

- Connectivity also enables some capabilities of the product to exist outside the physical device, in what is known as the product cloud.
- The data collected from these products can be then analyzed to inform decision-making, enable operational efficiencies, and continuously improve the performance of the product.

Smart Objects: A Definition

Historically, the definition of a smart object has been a bit nebulous because
of the different interpretations of the term by varying sources.

To add to the overall confusion, the term smart object, despite some semantic differences, is often used interchangeably with terms such **as smart sensor, smart device, IoT device, intelligent device, thing, smart thing, intelligent node, intelligent thing, ubiquitous thing, and intelligent product.**

A smart object is a device that should have following characteristics:

1.Processing unit:

A smart object must have processing unit for getting data, processing and analyzing sensing information received by the sensor(s), coordinating control signals to any actuators, and controlling a variety of functions on the smart object, including the communication and power systems.

The most common is a microcontroller because of its small form factor, flexibility, programming simplicity, ubiquity, low power consumption, and low cost.

2.Sensors and actuators:

A smart object is capable of interacting with the physical world through sensors and actuators. Depending on application one or more sensors and actuators are used.

3.Communication device:

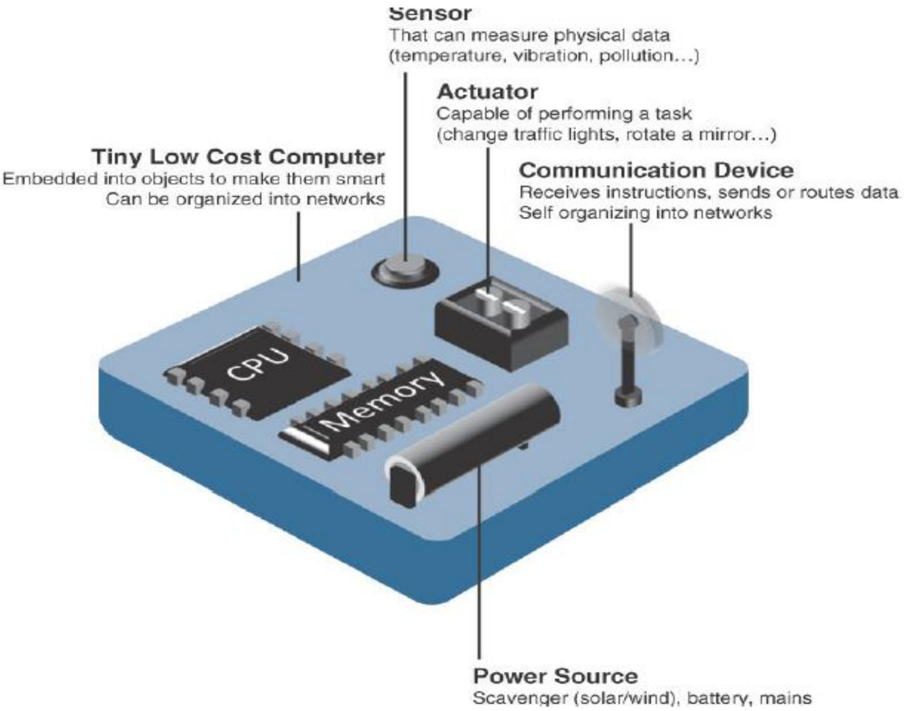
To talk with one smart object to another smart objects and to connect to network communication device is required. Connection between the devices can be either wired or wireless. Overwhelmingly, in IoT networks smart objects are wirelessly interconnected for a number of reasons, including cost, limited infrastructure availability, and ease of deployment. There are myriad different communication protocols for smart objects

4.Power source:

Power source is one of the most significant part . As with the other three smart object building blocks, the power requirements also vary greatly from application to application.

since power for smart objects are limited in power which needs to be run for long time and are not easily accessible.

This combination, especially when the smart object relies on battery power, implies that power efficiency, judicious power management, sleep modes, ultra-low power consumption hardware, and so on are critical design elements.


Sensor
That can measure physical data
(temperature, vibration, pollution…)
Actuator
Capable of performing a task
(change traffic lights, rotate a mirror…)
Tiny Low Cost Computer
Embedded into objects to make them smart
Can be organized into networks
Communication Device
Receives instructions, sends or routes data
Self organizing into networks
CPU
Memory
Power Source
Scavenger (solar/wind), battery, mains

Characteristics of a smart object

Trends in smart objects:

Following are trends impacting Smart object

Size is decreasing: Some smart objects are very small. This reduced size makes smart objects easier to embed in day to day objects.

Power consumption is decreasing: Many sensors consume less power that can be battery powered. Some battery-powered sensors last 10 or more years without battery replacement.

Processing power is increasing: Processors are continually getting more powerful and smaller. This is a key advancement for smart objects, as they become increasingly complex and connected.

1.4 M2M

Machine -to-Machine (M2M) refers to the communication or exchange of data between to two or more machines without human interfacing or interaction. The communication in M2M may be wired or wireless systems. The M2M uses a device such as sensor, RFID, meter, etc. to capture an ‘events’ like temperature, inventory level, etc., which is relayed through a network i.e., wireless, wired or hybrid to an application (software program), that translates the captured event into meaningful information. Unlike SCADA or other remote monitoring tools, M2M systems often use public networks and access methods -- for example, cellular or Ethernet -- to make it more cost-effective.

Machine-to-Machine (M2M)

- An M2M area network comprises of machines (or M2M nodes) which have embedded hardware modules for sensing, actuation, and communication.
- Various communication protocols can be used for M2M local area networks such as ZigBee, Bluetooth, ModBus, M-Bus, Wireless M-Bus, Power Line Communication (PLC), 6LoWPAN, IEEE 802.15.4, etc.
- The communication network provides connectivity to remote M2M area networks.
- The communication network can use either wired or wireless networks (IP- based).
- While the M2M area networks use either proprietary or non-IP based communication protocols, the communication network uses IP-based networks.

M2M System Architecture

The main components of M2M system are:

1. M2M area networks
2. Communication networks
3. Application domains
4. M2M gateways.

M2M area networks

- M2M network area consists of machines or M2M nodes which communicate with each other. The M2M nodes embedded with hardware modules such as sensors, actuators and communication devices.
- M2M uses communication protocol such as ZigBee, Bluetooth, Power line communication (PLC) etc. These communication protocols provide connectivity between M2M nodes within M2M area network.
- The M2M nodes communicate with in one network it can ’ t communicate with external network node. To enable the communication between remote M2M area networks, M2M gateways are used.

M2M Gateways

- The Gateway module provides control and localization services for data collection. The gateways also double up in concentrating traffic to the operator score. It supports Bluetooth, ZigBee, GPRS capabilities.

- M2M communication network serves as infrastructure for realizing communication between M2M gateway and M2M end user application or server. For this cellular network (GSM /CDMA), Wire line network and communication satellites may be used.

Communication networks

- The communication network provides the connectivity between M2M nodes and M2M applications.
- It uses wired or wireless network such as LAN, LTE, WiMAX, satellite communication etc.

Application domains

- It contains the middleware layer where data goes through various application services and is used by the specific business-processing engines.
- M2M applications will be based on the infrastructural assets that are provided by the operator. Applications may either target at end users, such as user of a specific M2M solution, or at other application providers to offer more refined building blocks by which they can build more sophisticated M2M solutions and services.

Difference between IoT and M2M

Communication Protocols

- M2M and IoT can differ in how the communication between the machines or devices happens.
- M2M uses either proprietary or non-IP based communication protocols for communication within the M2M area networks.

Machines in M2M vs Things in IoT

- The "Things" in IoT refers to physical objects that have unique identifiers and can sense and communicate with their external environment (and user applications) or their internal physical states.
- M2M systems, in contrast to IoT, typically have homogeneous machine types within an M2M area network.

Hardware vs Software Emphasis

- While the emphasis of M2M is more on hardware with embedded modules, the emphasis of IoT is more on software.
- Data Collection & Analysis
- M2M data is collected in point solutions and often in on-premises storage infrastructure.
- In contrast to M2M, the data in IoT is collected in the cloud (can be public, private or hybrid cloud).

Applications

- M2M data is collected in point solutions and can be accessed by on-premises applications such as diagnosis applications, service management applications, and on- premisis enterprise applications.
- IoT data is collected in the cloud and can be accessed by cloud applications such as analytics applications, enterprise applications, remote diagnosis and management applications, etc.

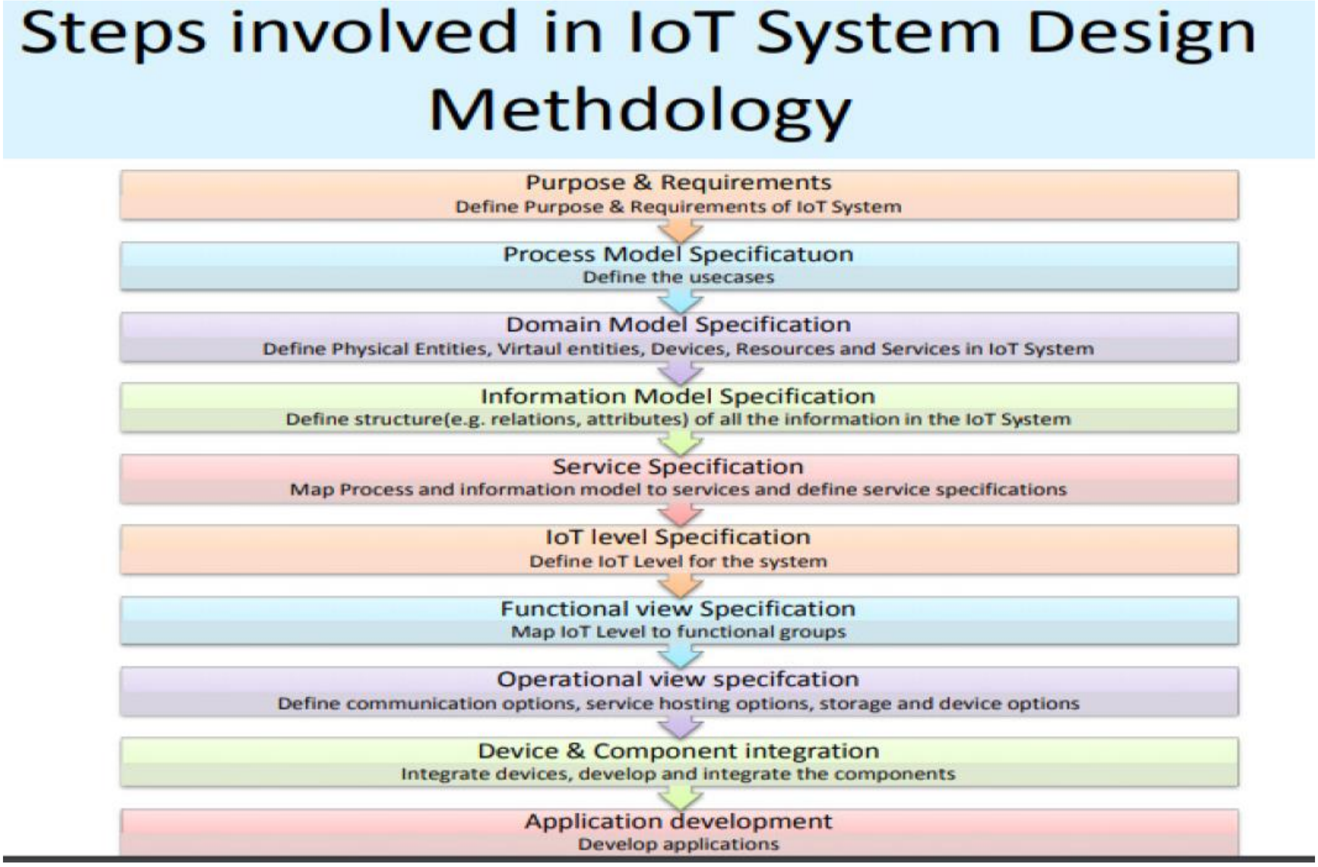
- M2M and IoT both are used when electronic devices are connected and share data with each other.

There are some differences between IoT and M2M based on technologies, system architectures and types of applications

Difference between IoT and M2M	
IoT	**M2M**
Digital connectivity among various devices to communicate	Devices connected to the formwork cycle using various machines and devices
Random work, work instructions are given among devices	The action triggered events among devices
Interchange of data is huge as it involves devices, machines, people, things, etc	Cloud computing helps in interacting/data exchange
Sensor integrated devices to enable IoT connectivity	Wired, wireless, cellular, etc
Two-way communication offers an option to all the devices	Mostly one way, based on triggered actions
Solution managing all the connections offer unlimited integration	Requires particular communication rules, resulting in minimal integration
Need internet for most of the cases	No need to rely on the internet

IoT Design Methodology that includes:

- Purpose & Requirements Specification
- Process Specification
- Domain Model Specification
- Information Model Specification
- Service Specifications
- IoT Level Specification
- Functional View Specification
- Operational View Specification
- Device & Component Integration
- Application Development



Steps involved in IoT System Design Methdology

Purpose & Requirements
Define Purpose & Requirements of IoT System

Process Model Specificatuon
Define the usecases

Domain Model Specification
Define Physical Entities, Virtaul entities, Devices, Resources and Services in IoT System

Information Model Specification
Define structure(e.g. relations, attributes) of all the information in the IoT System

Service Specification
Map Process and information model to services and define service specifications

IoT level Specification
Define IoT Level for the system

Functional view Specification
Map IoT Level to functional groups

Operational view specifcation
Define communication options, service hosting options, storage and device options

Device & Component integration
Integrate devices, develop and integrate the components

Application development
Develop applications

Advantages of Using Design methodology

- Reducing the design, testing and maintenance time
- Provide better interoperability
- Reduce the complexity

Step 1 : Purpose and Requirement Specification
Defines

- System purpose
- behavior and
- Requirements (such as data collection requirements, data analysis requirement, system management requirements, data privacy and security requirements, User interfaces requirements)
- **Purpose:** An automated irrigation mechanism which turns the pumping motor ON and OFF on detecting the moisture content of the earth without the intervention of human
- **Behavior:** System should monitor the amount of soil moisture content in soil. In case the soil moisture of the soil deviates from the specified range, the watering system is turned ON/OFF. In case of dry soil, it will activate the irrigation system, pumping water for watering the plants.
- **System Management Requirements:** system should remotely provide monitoring and control functions
- **Data Analysis Requirements:** system should perform local analysis of data

- **Application Deployment Requirement:** Deployed locally on device but acts remotely without manual intervention.
- **Security:** Authentication to Use the system must be available

Step 1: Purpose & Requirements Specification

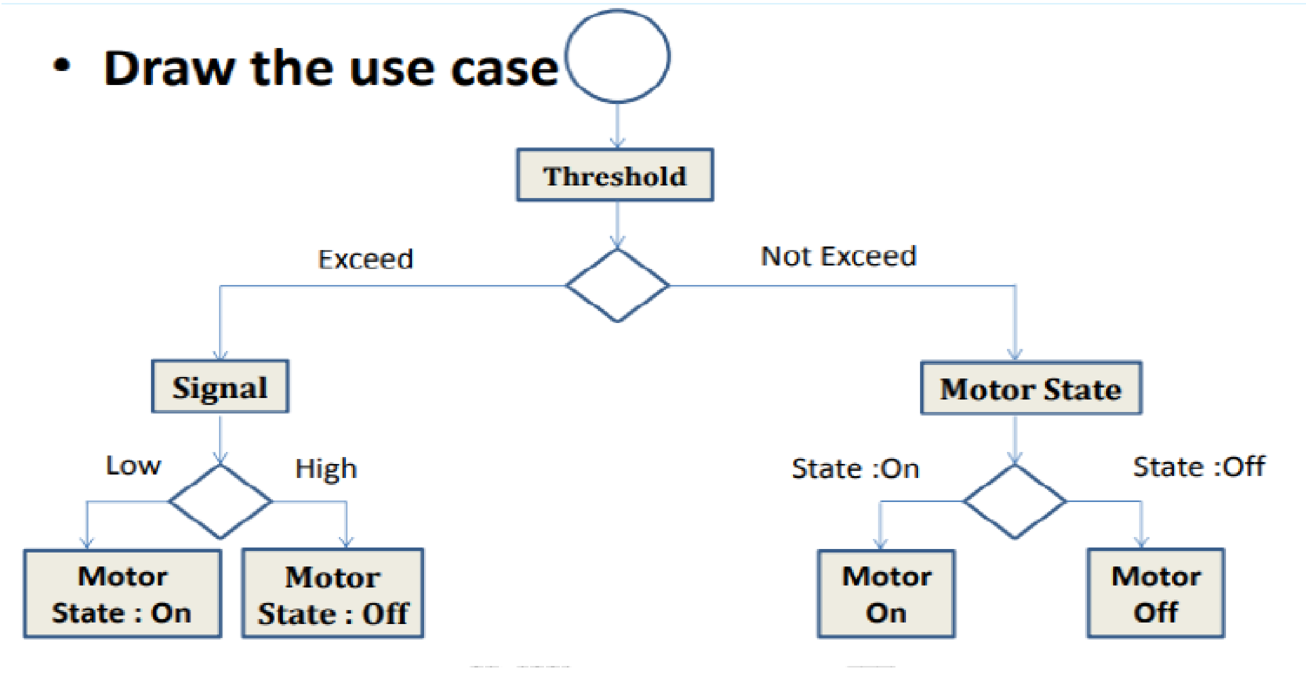
- The first step in IoT system design methodology is to define the purpose and requirements of the system. In this step, the system purpose, behavior and requirements (such as data collection requirements, data analysis requirements, system management requirements, data privacy and security requirements, user interface requirements, ...) are captured

Step 2 : Process Specification

- Define the process with the help of use cases
- The use cases are formally described based on Purpose & requirement specification In this
- use case: ‐ Circle denotes a state or an attribute

Step 2: Process Specification

- The second step in the IoT design methodology is to define the process specification. In this step, the use cases of the IoT system are formally described based on and derived from the purpose and requirement specifications.



- Draw the use case

Step 3 : Domain Model Specification

- Describes the main concepts, entities, and objects in the domain of IoT system to be designed
- It defines the attributes of the objects and relationships between them
- Entities, Objects and Concepts include the following: Physical entity, Virtual entity, Device, Resource, Service

- Physical Entity:
  ⁻ Discreet identifiable entity in physical environment
  ⁻ For e.g. Pump, motor, LCD
  ⁻ The IoT System provides the information about the physical entity (using sensors) or performs actuation upon the Physical entity (like switching a motor on etc.)
  ⁻ In smart irrigation example, there are three Physical entities involved:
    - Soil (whose moisture content is to be monitored)
    - Motor (To be controlled)
    - Pump (To be controlled)

- Virtual Entity:
  ⁻ Representation of physical entity in digital world
  ⁻ For each physical entity there is a virtual entity

- Device:
  ⁻ Medium for interactions between Physical and Virtual Entities.
  ⁻ Devices (Sensors) are used to gather information from the physical entities
  ⁻ Devices are used to identify Physical entities (Using Tags)
  ⁻ In Smart Irrigation System, device is soil moisture sensor and buzzer as well as the actuator (relay switch) attached to it.

- In smart irrigation system there are three services:
  ⁻ A service that sets the signal to low/ high depending upon the threshold value
  ⁻ A service that sets the motor state on/off
  ⁻ A controller service that runs and monitors the threshold value of the moisture and switches the state of motor on/off depending upon it. When threshold value is not crossed the controller retrieves the motor status from database and switches the motor on/off.

Step 3: Domain Model Specification

- The third step in the IoT design methodology is to define the Domain Model. The domain model describes the main concepts, entities and objects in the domain of IoT system to be designed. Domain model defines the attributes of the objects and relationships between objects. Domain model provides an abstract representation of the concepts, objects and entities in the IoT domain, independent of any specific technology or platform. With the domain model, the IoT system designers can get an understanding of the IoT domain for which the system is to be designed.

Step 4 : Information Model Specification

- Defines the structure of all the information in the IoT system (such as attributes, relations etc.)
- It does not describe the specifics of how the information is represented or stored.
- This adds more information to the Virtual entities by defining their attributes and relations
- I: e, Draw Class diagram

Step 4: Information Model Specification

- The fourth step in the IoT design methodology is to define the Information Model. Information Model defines the structure of all the information in the IoT system, for example, attributes of Virtual Entities, relations, etc. Information model does not describe the specifics of how the information is represented or stored. To define the information model, we first list the Virtual Entities defined in the Domain Model. Information model adds more details to the Virtual Entities by defining their attributes and relations.

Step 5 : Service Specification

- Define the services in IoT System, service types, service inputs/outputs, service endpoints, service schedules, service preconditions and service effects
- Services can be controller service, Threshold service, state service for smart irrigation system
- These services either change the state/attribute values or retrieve the current vlues.
- For eg.

  ⁻ Threshold service sets signal to high or low depending upon the soil moisture value.

  ⁻ State service sets the motor state: on or off

  ⁻ Controller service monitors the threshold value as well as the motor state and switches the motor on/off and updates the status in the database

Step 5: Service Specifications

- The fifth step in the IoT design methodology is to define the service specifications. Service specifications define the services in the IoT system, service types, service inputs/output, service endpoints, service schedules, service preconditions and service effects.

Step 6 : IoT Level Specification

- Decide the deployment level of IoT System. Here I am using Deployment Level 1.

Step 6: IoT Level Specification

- The sixth step in the IoT design methodology is to define the IoT level for the system. In Chapter-1, we defined five IoT deployment levels.

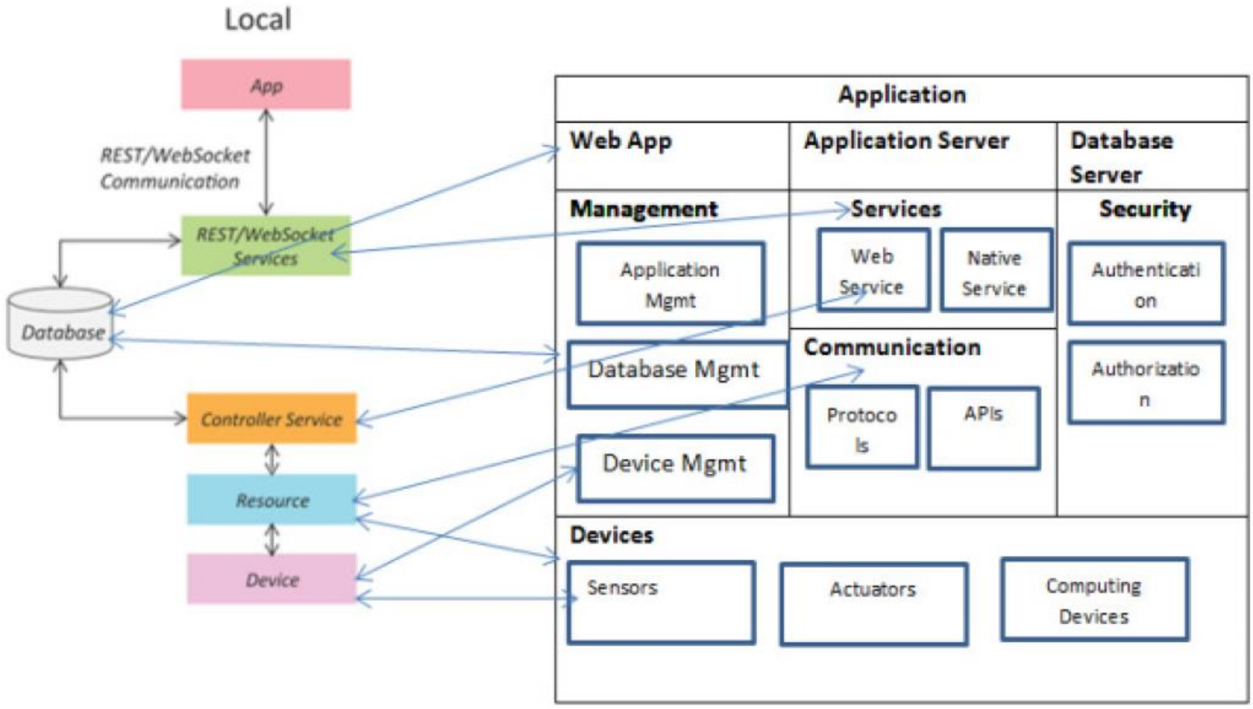
Step 7 : Functional View Specification

- Define the functions of IoT System grouped into various functional groups.
- These functional groups provide functionalities for interacting with the concepts defined in Domain model specification.

Step 7: Functional View Specification

- The seventh step in the IoT design methodology is to define the Functional View. The Functional View (FV) defines the functions of the IoT systems grouped into various Functional

Groups (FGs). Each Functional Group either provides functionalities for interacting with instances of concepts defined in the Domain Model or provides information related to these concepts.



Step 8 : Operational View Specification

- Define the Operations/options related to IoT System development
- Such as Device options, Storage options, Application hosting option

Step 8: Operational View Specification

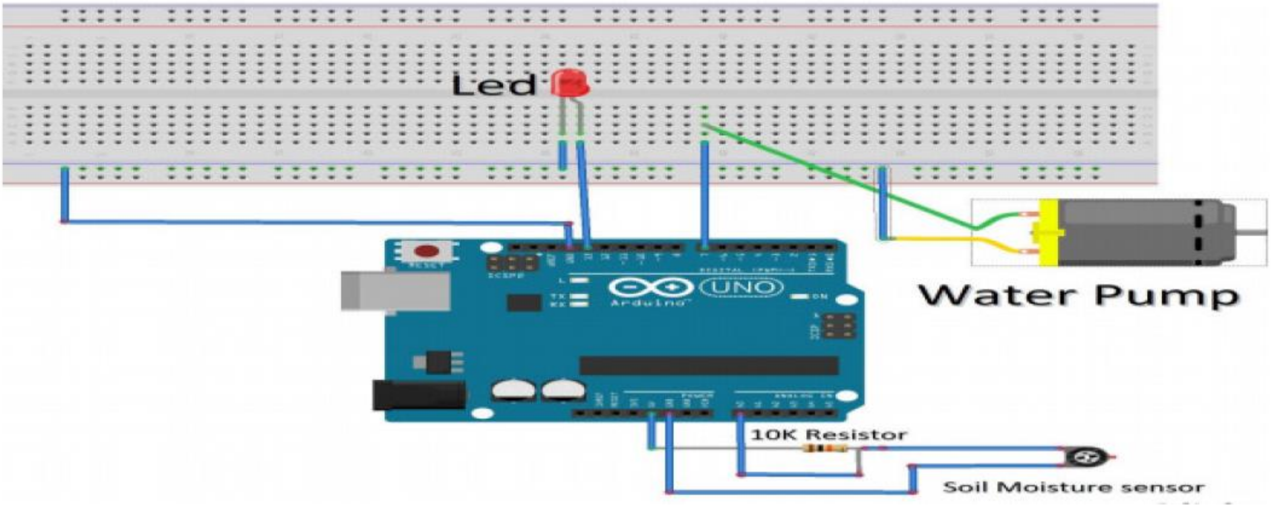
- The eighth step in the IoT design methodology is to define the Operational View Specifications. In this step, various options pertaining to the IoT system deployment and operation are defined, such as, service hosting options, storage options, device options, application hosting options, etc

Step 9 : Device and Component Integration

- Integrates the devices and components and draw a schematic diagram showing the same

Step 9: Device & Component Integration

- The ninth step in the IoT design methodology is the integration of the devices and components



Step 10 : Application development

- GUI / Screenshot of IoT Application

Step 10: Application Development

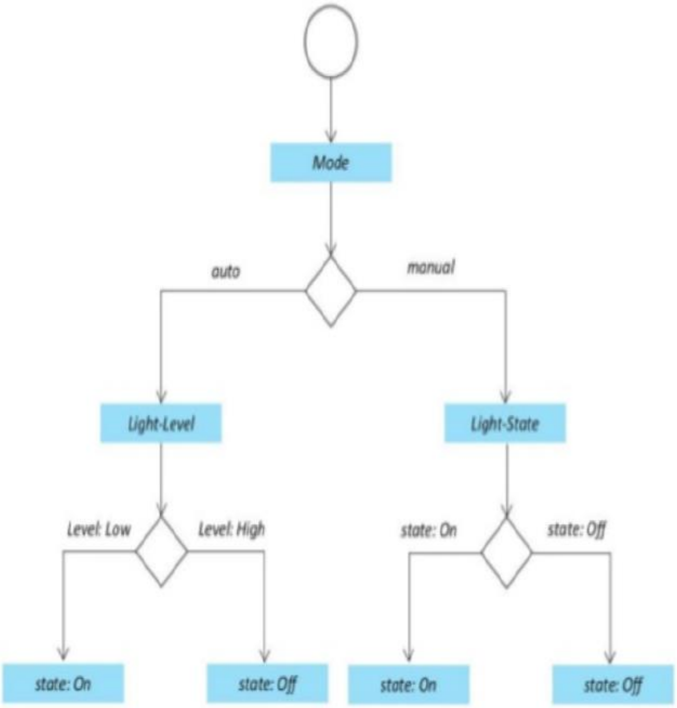
- The final step in the IoT design methodology is to develop the IoT application.

Home Automation Case Study

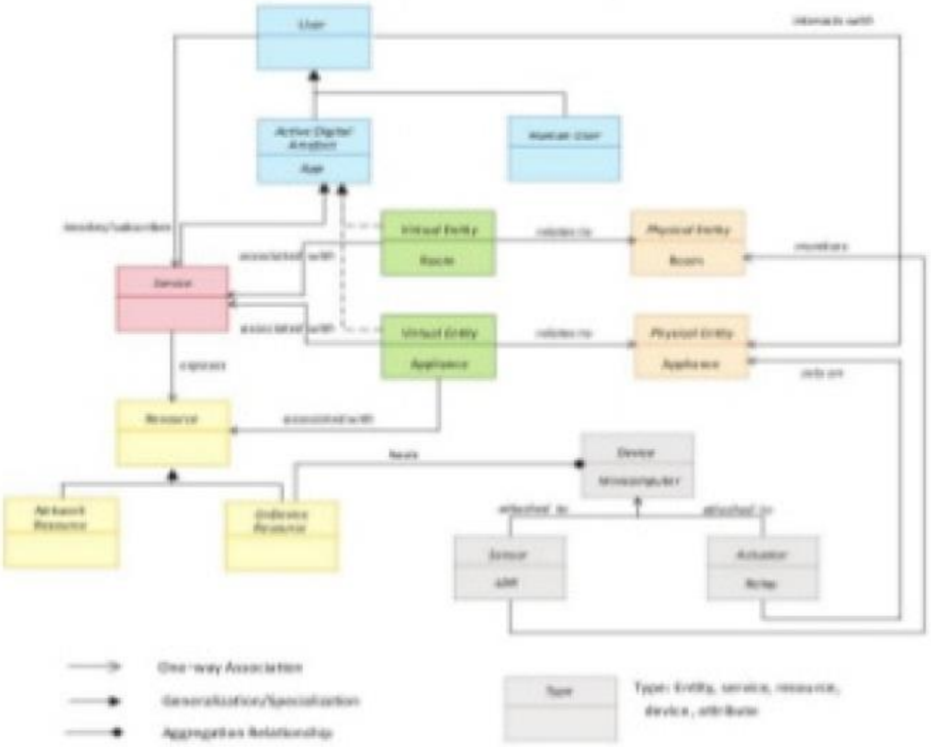
Step:1 - Purpose & Requirements

- Applying this to our example of a smart home automation system, the purpose and requirements for the system may be described as follows:
- **Purpose:** A home automation system that allows controlling of the lights in a home remotely using a web application.
- **Behavior:** The home automation system should have auto and manual modes. In auto mode, the system measures the light level in the room and switches on the light when it gets dark. In manual mode, the system provides the option of manually and remotely switching on/off the light.
- **System Management Requirement:** The system should provide remote monitoring and control functions.
- **Data Analysis Requirement:** The system should perform local analysis of the data.
- **Application Deployment Requirement**: The application should be deployed locally on the device but should be accessible remotely.
- **Security Requirement:** The system should have basic user authentication capability.

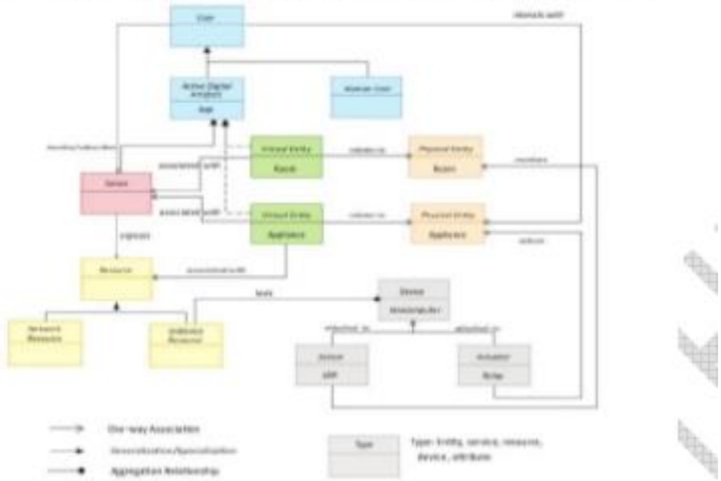
Step:2 - Process Specification



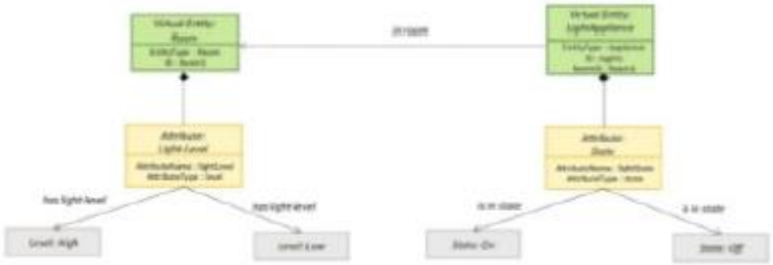
Step 3: Domain Model Specification



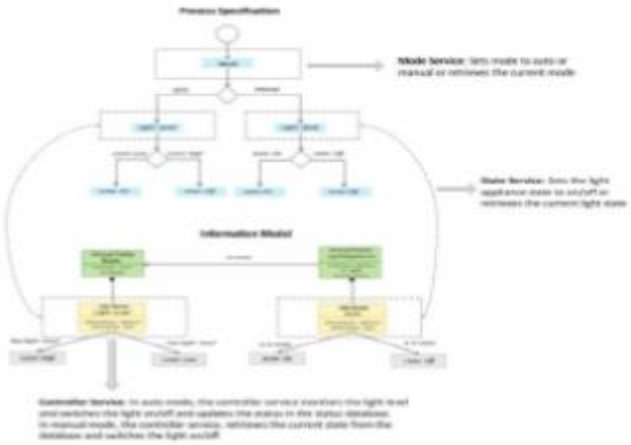
Step 3: Domain Model Specification



Step 4: Information Model Specification



Step 5: Service Specifications

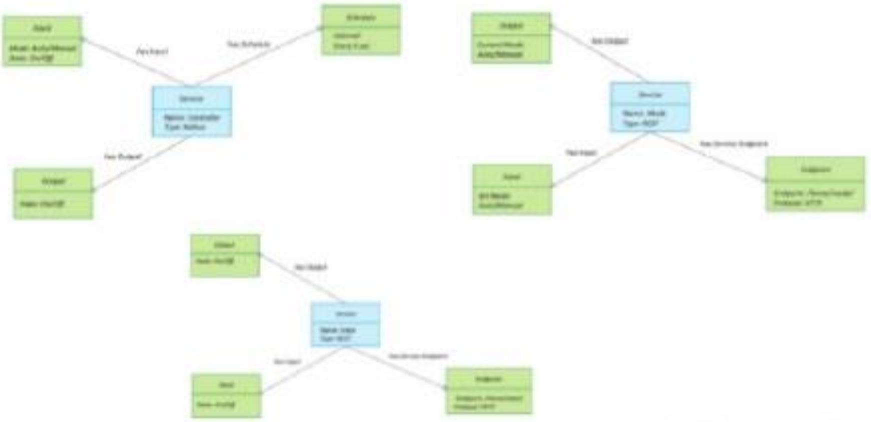


Mode Service: Sets mode to auto or manual or retrieves the current mode

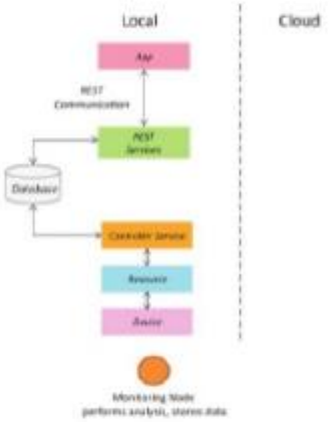
State Service: Sets the light appliance state to on/off or retrieves the current light state

Controller Service: In auto mode, the controller service monitors the light level and switches the light on/off and updates the status in the status database. In manual mode, the controller service, retrieves the current state from the database and switches the light on/off

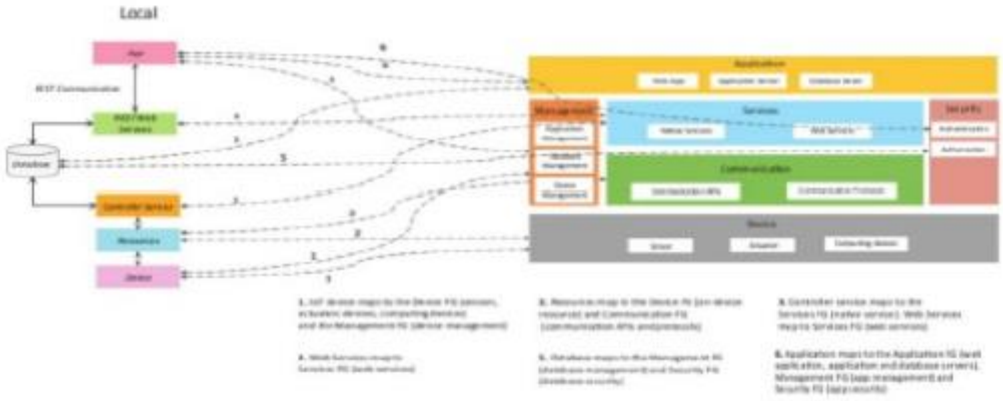
Step 5: Service Specifications



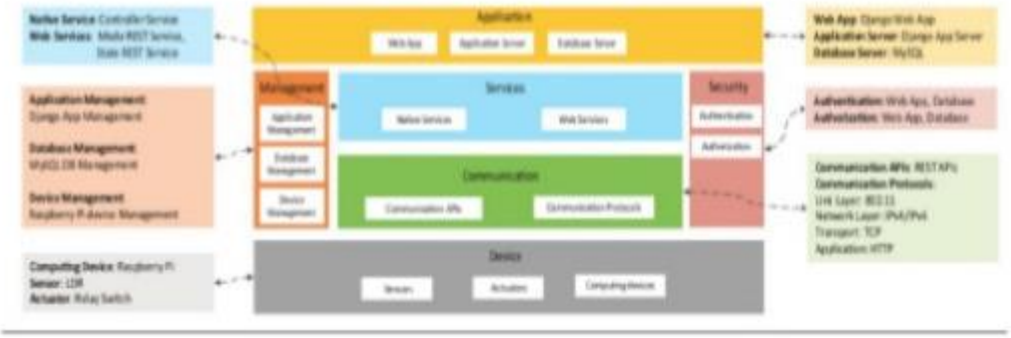
Step 6: IoT Level Specification



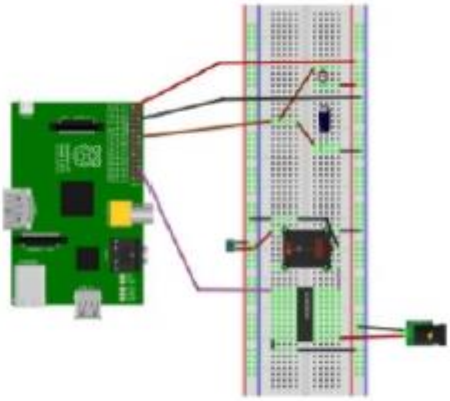
Step 7: Functional View Specification


Local
App
REST Communication
REST Web Services
Database
Controller Service
Resources
Device
Application
Web App
Application Server
Database Server
Management
Application Management
Database Management
Device Management
Services
Native Services
Web Services
Security
Authentication
Authorization
Communication
Communication APIs
Communication Protocols
Device
Sensor
Actuator
Computing device
1
2
3
4
5
6
1. IoT device maps to the Device FG (sensors, actuators devices, computing devices) and the Management FG (device management)
2. Resources map to the Device FG (on-device resource) and Communication FG (communication APIs and protocols)
3. Controller service maps to the Services FG (native service). Web Services map to Services FG (web services)
4. Web Services map to Services FG (web services)
5. Database maps to the Management FG (database management) and Security FG (database security)
6. Application maps to the Application FG (web application, application and database servers), Management FG (app management) and Security FG (app security)

Step 8: Operational View Specification

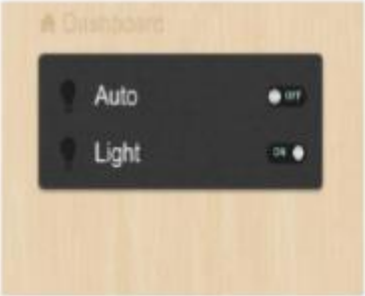


Step 9: Device & Component Integration



Step 10: Application Development

- Auto
  - Controls the light appliance automatically based on the lighting conditions in the room
- Light
  - When Auto mode is off, it is used for manually controlling the light appliance.
  - When Auto mode is on, it reflects the current state of the light appliance.

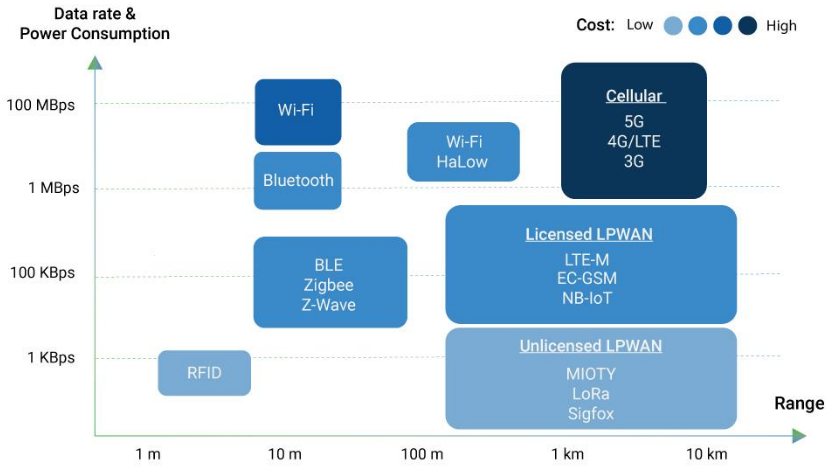


Please Refer Textbook Internet of Things A Hands-on Approach by Arshdeep Bahga, Vijay Madisetti Page No **114-137**

Communication Aspects – Wireless Medium Access Issues

Protocols used in Wireless Communication

- Bluetooth
- BLE
- ZigBee
- Z-Wave
- Wi-Fi
- RFID
- Cellular – 2G/3G/4G/LTE/5G
- 6LoWPAN
- NB-IoT
- NFC

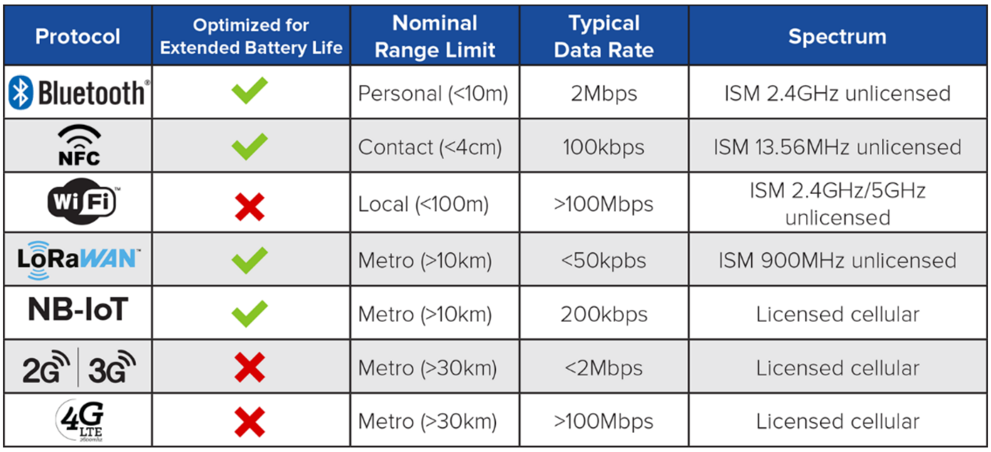


Challenges in Wireless Medium Communication

The Wireless channel is very dynamic and unpredictable. Any loss of communication may result in the form of huge financial losses or it may even result in the loss of life.
Challenges:

- **Scalability** is the main requirement of any IoT system. It poses an important challenge, as number of connected nodes in the system keeps on changing, the amount and type of data generated varies. It causes data variability issues.
- **Heterogeneity** of the devices and technologies creates the issue of veracity of the data. Every technology has its own data rates. This variation in data rates causes the variability in data speed. It also creates the issue of variability in the type of data, as different technologies have different data representation formats.
- **Reliability and Availability:** The Wireless channel is very dynamic and unpredictable. Any loss of communication may result in the form of huge financial losses or it may even result in the loss of life.

- **QoS:** Radio resource control, discontinuous reception/packet loss, end-to-end packet delay, availability of bandwidth are important factors in determining the QoS. Numerous interactions and the session setups in IoT will create the problem of Signal load in the network. Interference, end-to-end packet delay, power saving and signal saving are a few other challenges under providing QoS
- **Energy Efficiency** is usually defined as the number of bits that can be sent over a unit of power consumption which is usually quantified by bits per Joule. Any method that helps in reduction of power consumption must be given a thought if it can be incorporated in creation of an efficient system for the future.



Protocol	Optimized for Extended Battery Life	Nominal Range Limit	Typical Data Rate	Spectrum
Bluetooth	✓	Personal (<10m)	2Mbps	ISM 2.4GHz unlicensed
NFC	✓	Contact (<4cm)	100kbps	ISM 13.56MHz unlicensed
Wi Fi	✗	Local (<100m)	>100Mbps	ISM 2.4GHz/5GHz unlicensed
LoRaWAN	✓	Metro (>10km)	<50kpbs	ISM 900MHz unlicensed
NB-IoT	✓	Metro (>10km)	200kbps	Licensed cellular
2G | 3G	✗	Metro (>30km)	<2Mbps	Licensed cellular
4G LTE	✗	Metro (>30km)	>100Mbps	Licensed cellular

Benefits and Issues in Wireless Communication Technologies

- Bluetooth:
  - Benefits:
    - Low Power Consumption
    - Easily upgradable
    - Low Cost od equipment
  - Issues:
    - Low bandwidth
    - Short Range
    - Security Risk as it leaves data vulnerable to interception

- Wi-Fi:
  - Benefits:
    - Low cost
    - Commonly available
    - Simplicity – No need for extra hardware
  - Issues:
    - Wi-Fi is not designed to create mesh networks
    - High Power Usage
    - Limited number of devices can be connected

- ZigBee:
  - Benefits:

- It is less complex than Bluetooth
- Zigbee has a mesh network topology with low cost, multi hope data transmission and is power effective
- Long battery life
- Supports many nodes

☛ Issues:

- Short range
- Low complexity, and low data speed
- High maintenance cost
- Replacement with Zigbee compliant appliances can be costly
- Zigbee is not secure like WiFi based secured system