Internet of Things

Unit I

▼ IoT definition

- The Internet of Things (IoT) describes the network of physical objects—“things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.
- **Why is Internet of Things (IoT) so important?**
    - By means of low-cost computing, the cloud, big data, analytics, and mobile technologies, physical things can share and collect data with minimal human intervention. In this hyperconnected world, digital systems can record, monitor, and adjust each interaction between connected things. The physical world meets the digital world—and they cooperate.
- **What technologies have made IoT possible?**
    - Access to low-cost, low-power sensor technology
    - Connectivity
    - Cloud computing platforms
    - Machine learning and analytics
    - Conversational artificial intelligence (AI)

▼ Characteristics of IoT

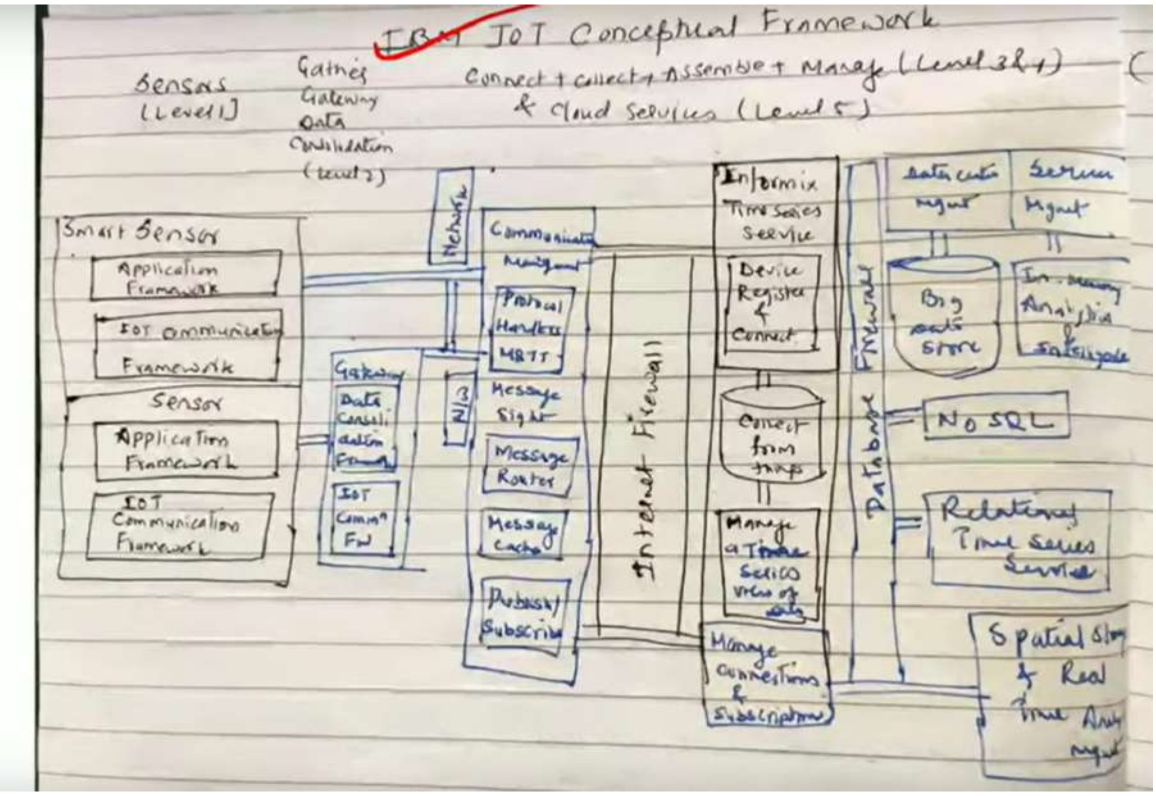
The fundamental characteristics of the IoT are as follows:

1. **Interconnectivity:** With regard to the IoT, anything can be interconnected with the global information and communication infrastructure.
2. **Intelligence:** The intelligence of IoT devices depends on the sensors' intelligence. The sensors send the data to the user for further analysis.
3. **Scalability:** Scalability means the amount of data one can handle efficiently. The IoT has created a setup to handle enormous data and generate useful analysis.
4. **Things-related services:** The IoT is capable of providing thing-related services within the constraints of things, such as privacy protection and semantic consistency between physical things and their associated virtual things.
5. **Heterogeneity:** The devices in the IoT are heterogeneous as based on different hardware platforms and networks. They can interact with other devices or service platforms through different networks.

6. **Dynamic changes:** We need to create IoT devices in a way that they can adapt to the environment. For example, an AC should have a sensor that can send a signal to the cloud and adjust it to the premises of the place.
7. **Self Upgradation:** with its artificial intelligence, IoT upgrades itself without human help. It also allows the set up of a network for the addition of any new IoT devices. Thus, the technology can quickly start working without delay if the setup has already been done.
8. **Enormous scale:** The number of devices that need to be managed and that communicate with each other will be at least an order of magnitude larger than the devices connected to the current Internet.
9. **Safety:** As we gain benefits from the IoT, we must not forget about safety. As both the creators and recipients of the IoT, we must design for safety. This includes the safety of our personal data and the safety of our physical well-being
10. **Connectivity:** Connectivity enables network accessibility and compatibility. Accessibility is getting on a network while compatibility provides the common ability to consume and produce data.

▼ **IoT conceptual framework**

- **Sensors—lvl1**
- **gather—lvl2**
- **connect + collect + assemble + manage + cloud servises—lvl3**


~~IBM~~ IoT Conceptual Framework
Sensors (Level1)
Gather Gateway Data Consolidation (Level 2)
Connect + collect + Assemble + Manage (Level 3 & 4) & Cloud Services (Level 5)
Smart Sensor
Application Framework
IOT Communication Framework
Sensor
Application Framework
IOT Communication Framework
Gateway
Data Consolidation Fw
IOT Comm Fw
Network
N/3
Communication Management
Protocol Handlers
MQTT
Message Sight
Message Router
Message Cache
Publish/Subscribe
Internet Firewall
Informix Time Series Service
Device Register & Connect
Connect from things
Manage a Time Series view of data
Manage Connections & Subscriptions
Database Firewall
Data center mgmt
Service Mgmt
Big data Store
In-memory Analytics & Solutions
No SQL
Relational Time Series Service
Spatial Store & Real Time Analytics mgmt

- The main tasks of this framework are to analyze and determine the smart activities of these intelligent devices through maintaining a dynamic interconnection among those devices.

The proposed framework will help to standardize IoT infrastructure so that it can receive e-services based on context information leaving the current infrastructure unchanged.

- **Connectivity Layer**
  This layer includes all the physical devices involved in the framework and the interconnection among them.
- **Access Layer**
  Access layer comprises topology definition, network initiation, creation of domains etc.
- **Abstraction Layer**
  One of the most important characteristics of Open Flow is to add virtual layers with the preset layers, leaving the established infrastructure unchanged.
- **Service Layer**
  It is not only responsible for storing data but also to provide security along with it.

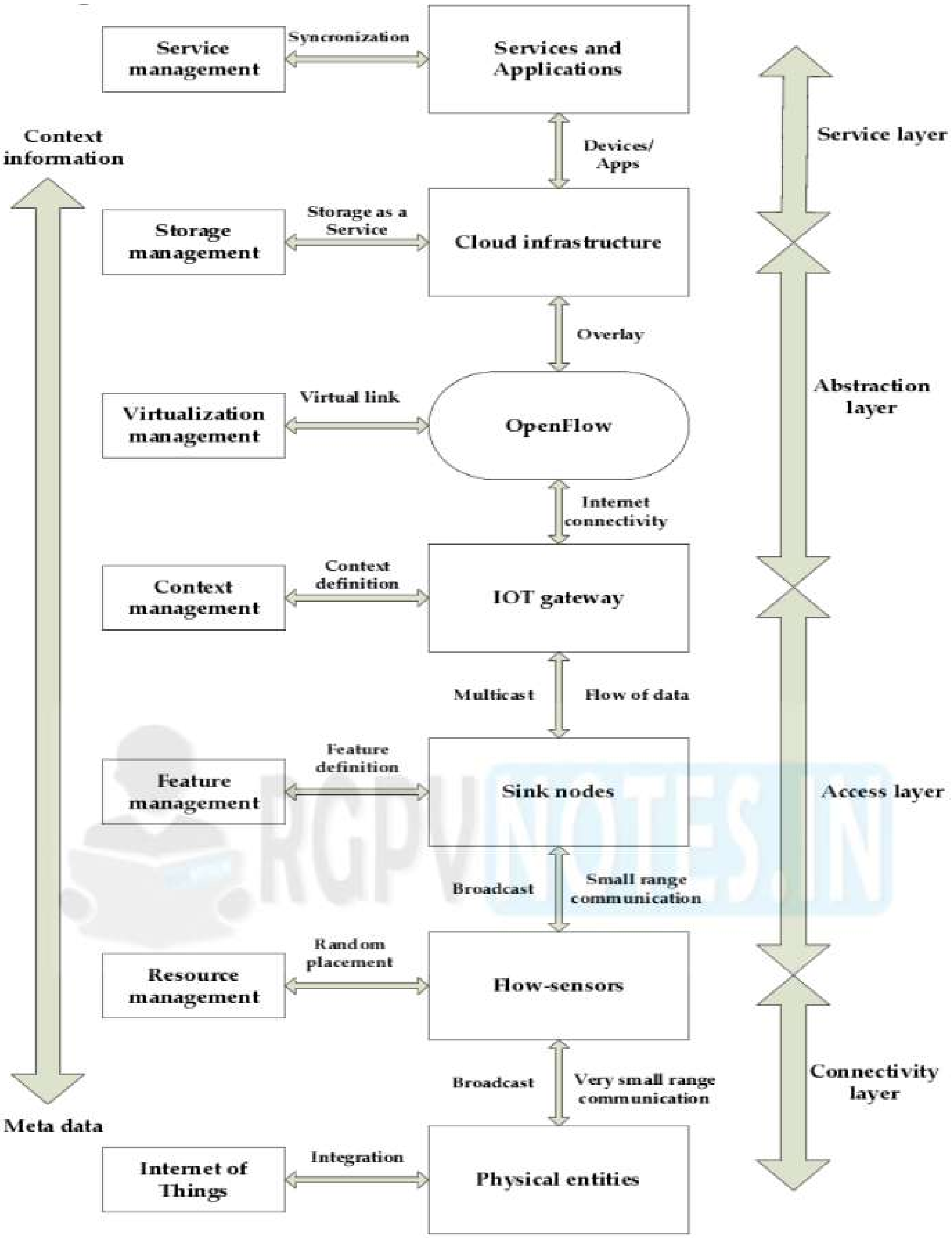
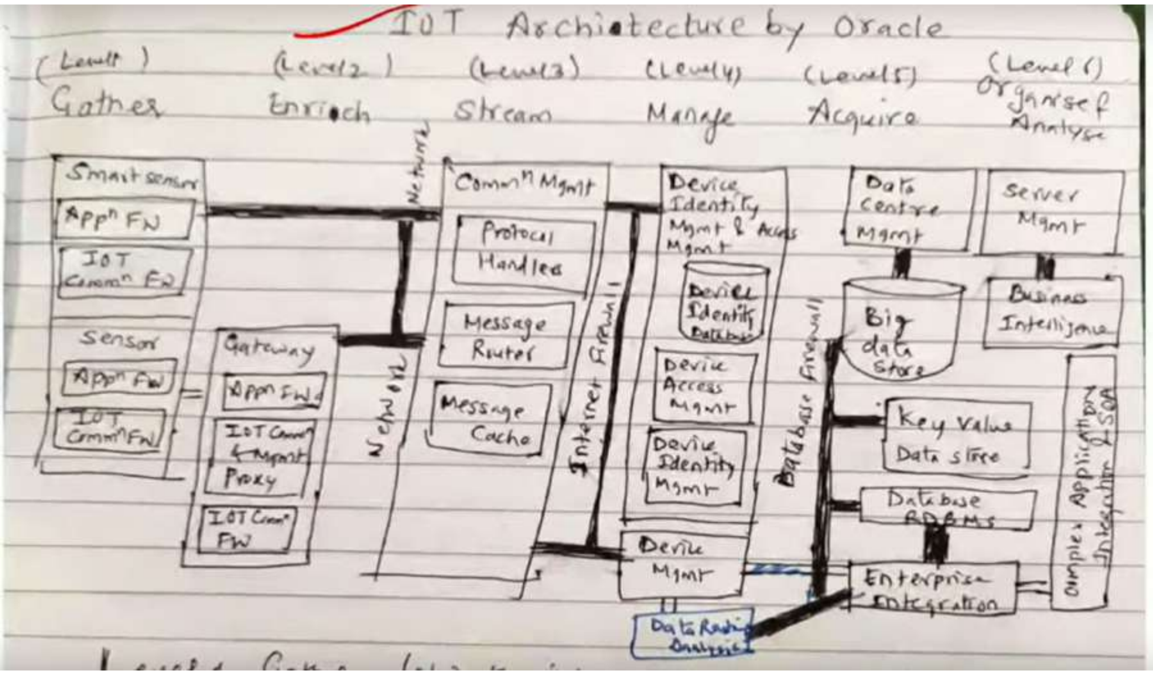


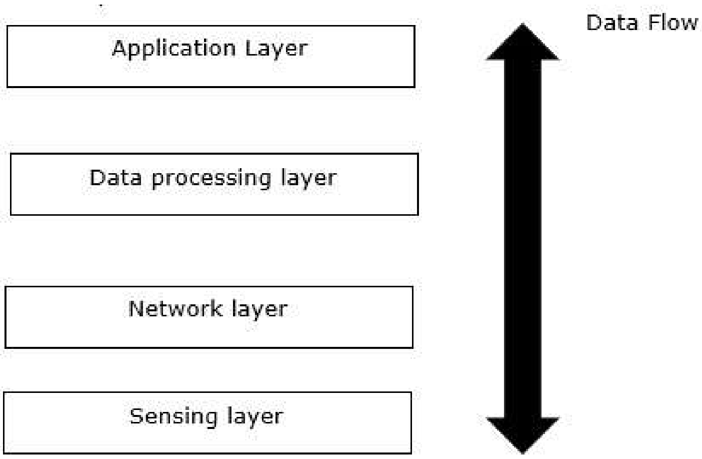
Figure 1.1: Conceptual Framework

▼ IoT architectural framework

- **Gather**—*lvl1* **+ Enrich + Stream + manage + acquire + organize & analyse**—*lvl6*



- IOT architecture consists of different layers of technologies supporting IOT.
- Now let us see the basic fundamental architecture of IoT which consists of four stages as shown in the diagram given below
  - **Sensing Layer −** The first stage of IoT includes **sensors, devices, actuators** etc. which collect data from the physical environment, processes it and then sends it over the network.
  - **Network Layer −** The second stage of the IoT consists of **Network Gateways and Data Acquisition Systems**.
    - **DAS converts the analogue data (collected from Sensors) into Digital Data**. It also performs malware detection and data management.
    - Protocols
    - IOT gateways
  - **Data Processing Layer −** The third stage of IoT is the most important stage. Here, **data is pre-processed** on its variety and separated accordingly. After this, it is sent to Data Centres. Here Edge IT comes into use.
  - **Application Layer −** The fourth stage of IoT consists of Cloud/Data Centres where data is managed and used by applications like agriculture, defence, health care etc.



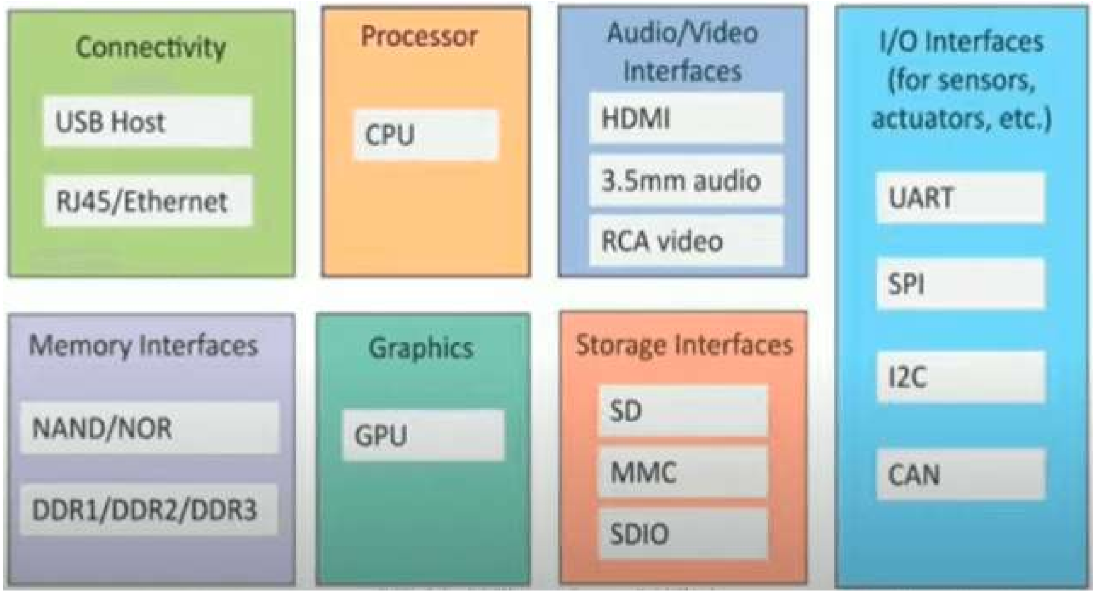
▼ **Components of IoT ecosystems**

- **Sensing and embedding components**
    - first tier of an IoT ecosystem and it forms the backbone of the entire Internet of Things network
    - **Sensors—**work to gather minute data from the surrounding environment. This allows an IoT device to capture relevant data for real-time or post-processing.
    - **Actuators—**Actuators work opposite to that of sensors. While sensors, sense; actuators act. They receive a signal or a command and on its basis they cause an action.
- **Connectivity**
    - **Protocols—**communication channel is necessary between sensors and the cloud. IoT protocols are responsible for transferring data in the online world and this transmission can only be possible if two devices are safely connected.
    - **IoT gateways—**Incoming, raw data from the sensors must pass through gateways to reach the cloud. Gateways translate network protocols ensuring seamless communication of all devices within the network.
- **IoT cloud**
    - This high-performance facility majorly ties the components to the IoT ecosystem together. It handles the data, stores it and makes decisions to make or break a deal. All of this is performed for colossal amounts of data in just under milliseconds
- **IoT analytics and data management**
    - Data may be a small word but it holds immense power that can pose a huge effect on any business. IoT Analytics is used to make sense of the vast amounts of analog data.
- **End-user devices and user interface**
    - The user interface is the visible component that is easily accessible and in control of the IoT user. This is where a user can control the system and set their preferences. The

more user-friendly this component of the IoT ecosystem is, the easier is a user's interaction.

▼ **Physical design of IoT**

- A physical design of an IoT system refers to the individual node devices and their protocols that are utilized to create a functional IoT ecosystem.
- Each node device can perform tasks such as remote sensing, actuating, monitoring, etc., by relying on physically connected devices. It may also be capable of transmitting information through different types of wireless or wired connections.
- The devices generate data, and the data is used to perform analysis and do operations for improving the system. For instance, a moisture sensor is used to obtain the moisture data from a location, and the system analyses it to give an output.



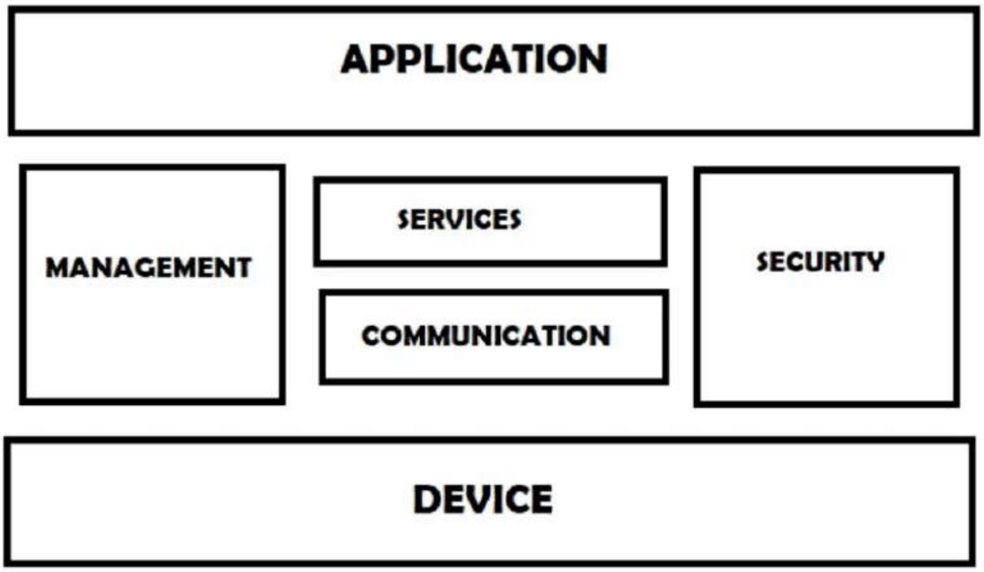
Block Diagram of IoT Devices (Physical Design)

▼ **Logical design of IoT**

- A logical design for an IoT system is the actual design of how its components (computers, sensors, and actuators) should be arranged to complete a particular function.
- **IoT logical design includes:**

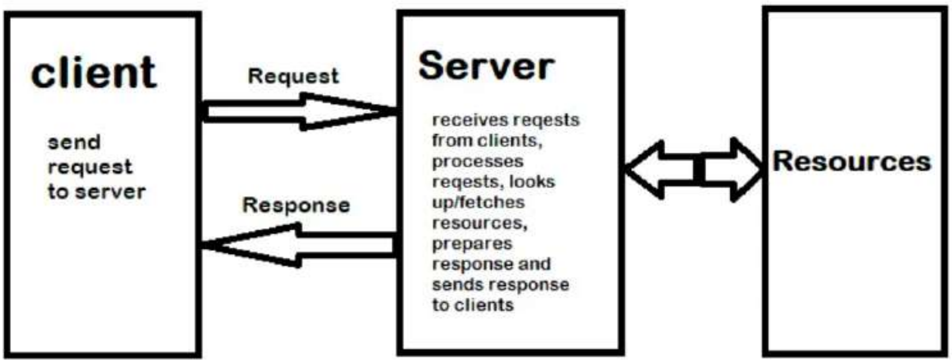
  ▼ **IoT functional blocks**

  1. Functional blocks consist of devices that handle the communication between the server and the host
  2. IoT systems include several functional blocks such as Devices, communication, security, services, and application.



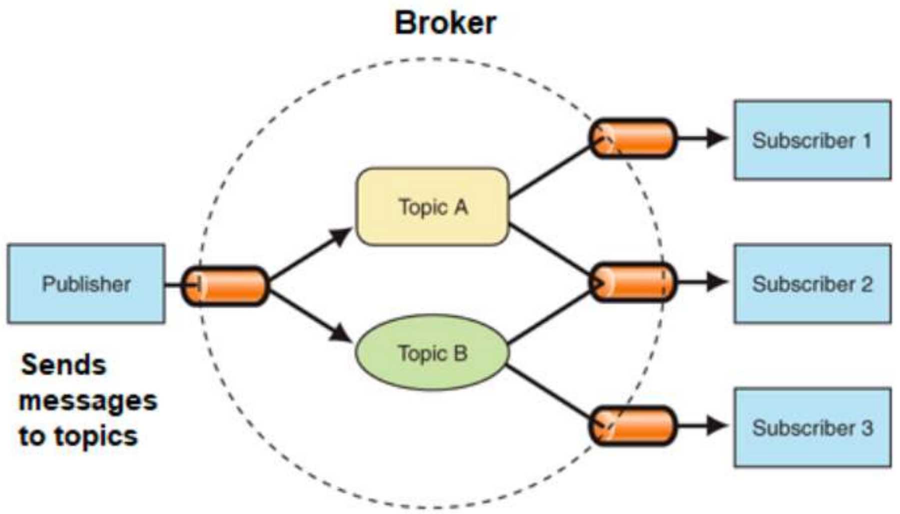
▼ **IoT communications models**

1. **Request-Response Model**
    a. Request-response model is a communication model in which the client sends requests to the server and the server responds to the requests.
    b. HTTP works as a request-response protocol between a client and a server

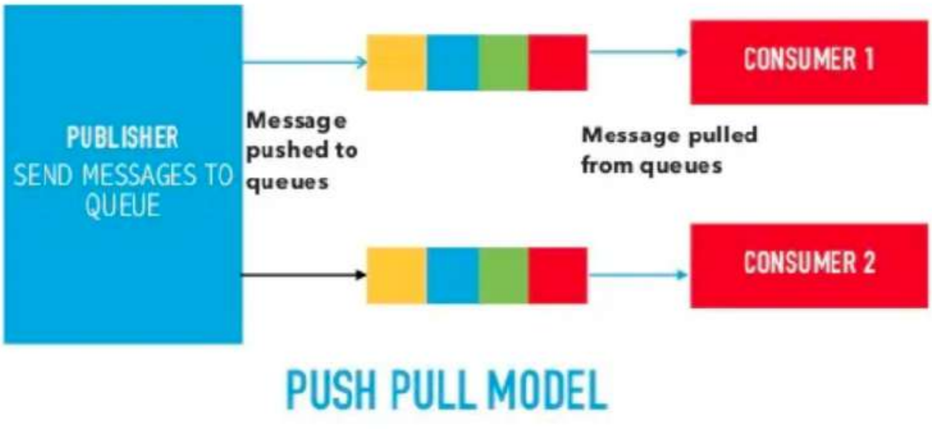


Request-Response Communication Model

2. **Publisher-Subscriber Model**
    a. This model comprises three entities: Publishers, Brokers, and Consumers
    b. **Publishers** are the source of data
    c. **Consumers** subscribe to the topics which are managed by the broker
    d. **Brokers** responsibility is to accept data from publishers and send it to the appropriate consumers

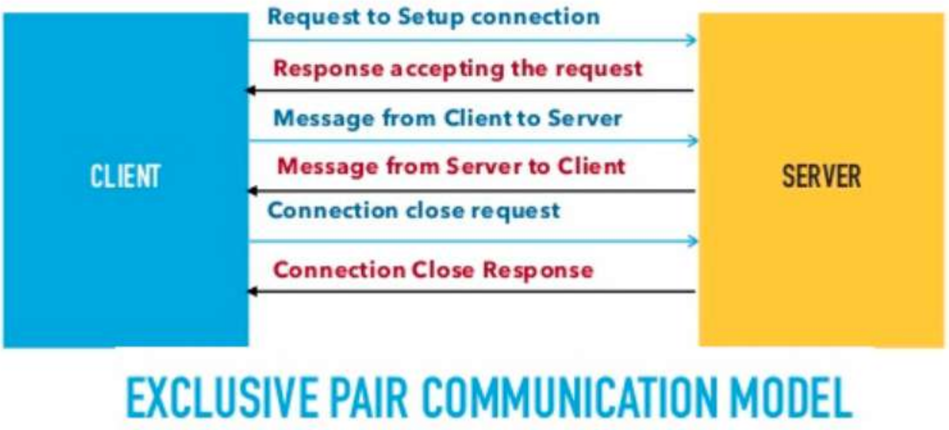


3. **Push-Pull Model**
    a. The push-pull model constitutes data publishers, data consumers, and data queues
    b. **Publishers** and **Consumers** are not aware of each other
    c. Publishers publish the message/data and push it into the queue. The consumers, present on the other side, pull the data out of the queue.
    d. **Queues** help in decoupling the messaging between the producer and consumer. Queues also act as a buffer which helps in situations where there is a mismatch between the rate at which the producers push the data and consumers pull the data.



4. **Exclusive Pair**
    a. **Exclusive Pair** is the bi-directional model, including full-duplex communication between client and server. The connection is constant and remains open till the client sends a request to close the connection.

b. **WebSocket-based communication API is fully based on this model.**


Request to Setup connection
Response accepting the request
Message from Client to Server
Message from Server to Client
Connection close request
Connection Close Response
CLIENT
SERVER
EXCLUSIVE PAIR COMMUNICATION MODEL

▼ **IoT communication APIs**

1. **REST**

    a. REpresentational State Transfer (REST) is a set of architectural principles by which you can design web services and web APIs that focus on a system’s resources and how resource states are addressed and transferred.

    b. REST APIs follow the request-response communication model.

    c. The REST architectural constraints apply to the components, connectors, and data elements, within a distributed hypermedia system.

2. **Web Socket**

    a. Web Socket APIs allow bi-directional, full-duplex communication between clients and servers.

    b. It follows the exclusive pair communication model.

    c. This Communication API does not require a new connection to be set up for each message to be sent between clients and servers.

    d. Once the connection is set up the messages can be sent and received continuously without any interruption.

    e. **WebSocket APIs are suitable for IoT Applications** with low latency or high throughput requirements.

▼ **IoT enablers**

- IoT-enabling technologies primarily focus on converting a standalone device into an IoT device by giving it the additional possibility of connecting to the internet and exchanging information with it.
- **IoT enabling technologies are**

1. Wireless Sensor Network

2. Cloud Computing
3. Big Data Analytics
4. Communications Protocols
5. Embedded System

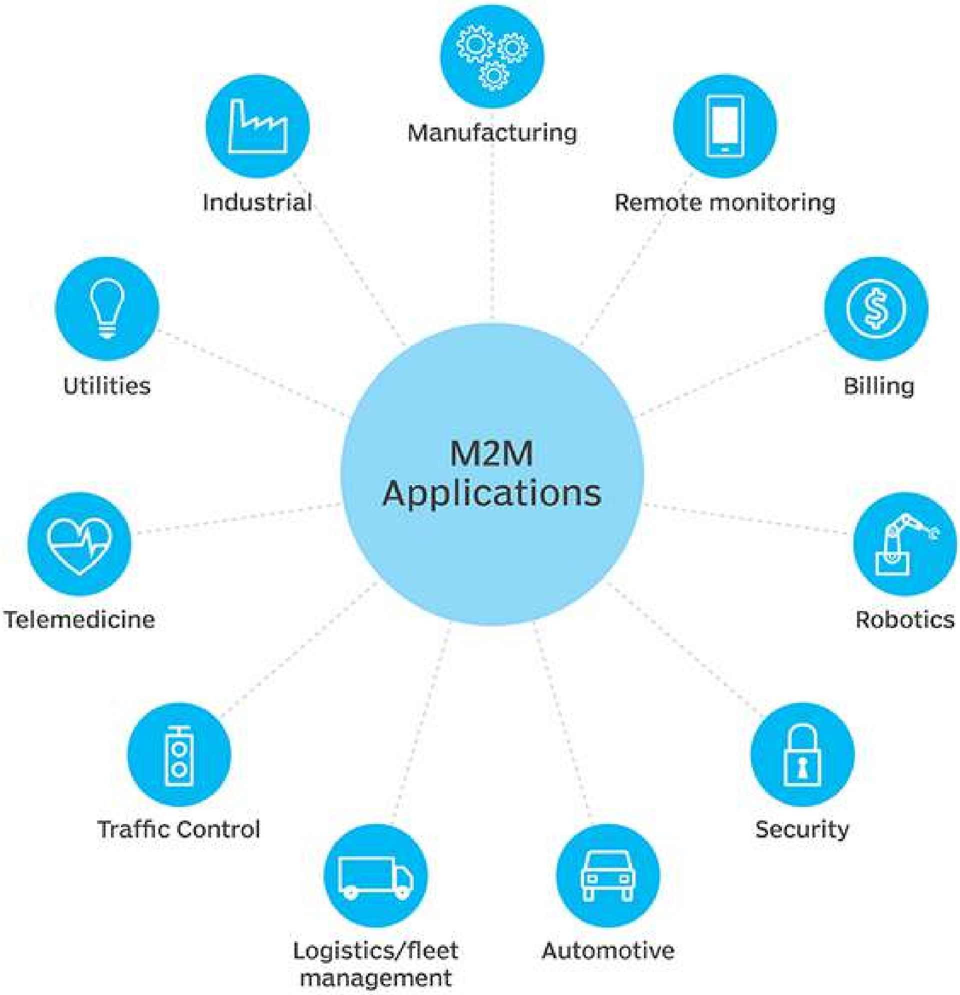
▼ Modern day IoT applications

- There are endless possibilities for having an interconnected web of "things" that can interact with each other over the internet.
- **Smart Agriculture**
- **Smart Home**
- **Smart Pollution Control**
- **Smart Vehicles**
- **Smart Healthcare**
- **Smart Cities**
- **Smart Retail**

▼ Explain M2M communication and bring out differences between IoT and M2M. How data analytics approaches differ in M2M and IoT.

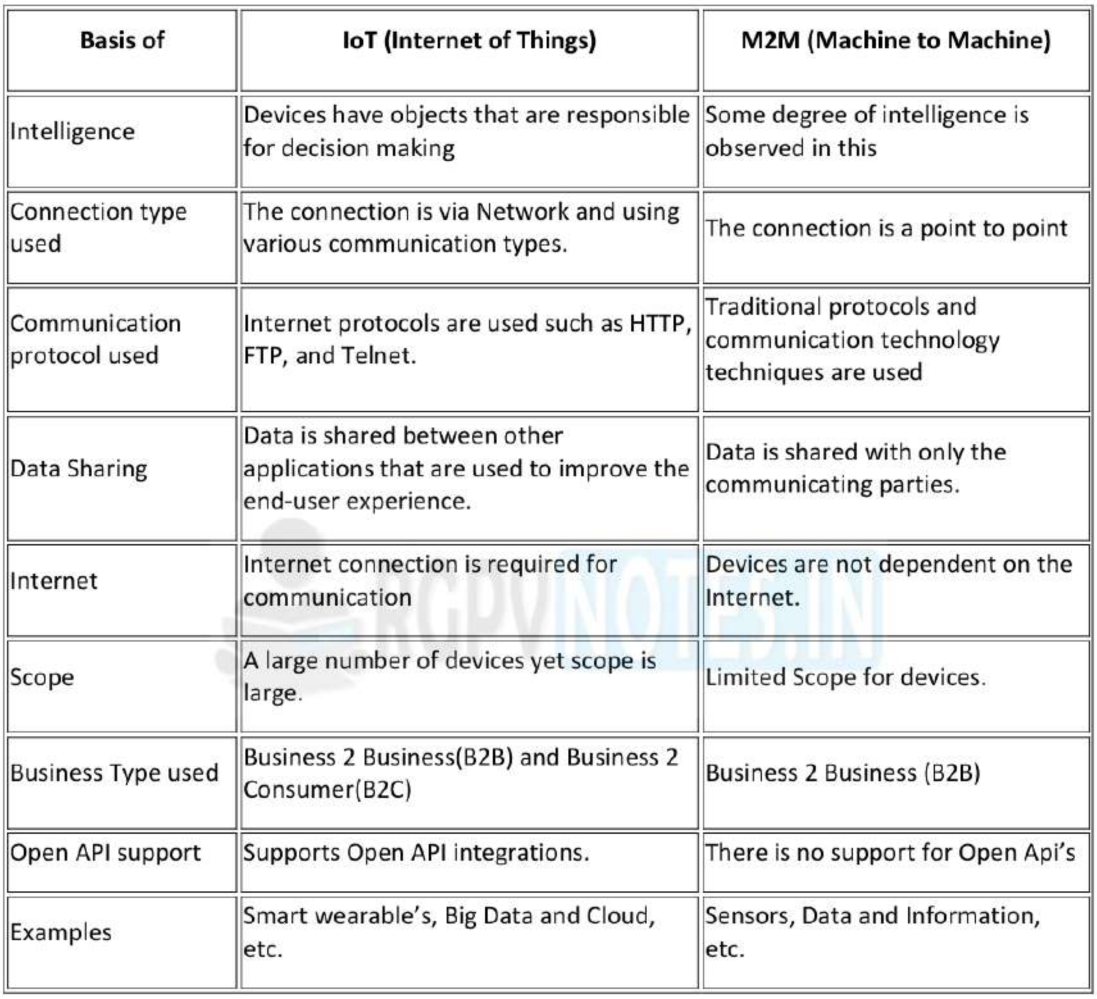
▼ M2M communications

- M2M means two machines "communicating," or exchanging data, without human interfacing or interaction. This includes serial connection, power line connection (PLC), or wireless communications in the industrial Internet of Things (IoT). Switching over to wireless has made M2M communication much easier and enabled more applications to be connected.
- M2M allows virtually any sensor to communicate, which opens the possibility of systems monitoring themselves and automatically responding to changes in the environment, with a much-reduced need for human involvement. M2M can refer to any two machines—wired or wireless—communicating with one another.
- **How M2M works**
  - The main components of an M2M system include sensors, RFID, a Wi-Fi or cellular communications link, and autonomic computing software programmed to help a network device interpret data and make decisions.
  - M2M systems often use public networks and access methods -- for example, cellular or Ethernet -- to make it more cost-effective.



M2M applications

▼ **IoT vs M2M**



Basis of	IoT (Internet of Things)	M2M (Machine to Machine)
Intelligence	Devices have objects that are responsible for decision making	Some degree of intelligence is observed in this
Connection type used	The connection is via Network and using various communication types.	The connection is a point to point
Communication protocol used	Internet protocols are used such as HTTP, FTP, and Telnet.	Traditional protocols and communication technology techniques are used
Data Sharing	Data is shared between other applications that are used to improve the end-user experience.	Data is shared with only the communicating parties.
Internet	Internet connection is required for communication	Devices are not dependent on the Internet.
Scope	A large number of devices yet scope is large.	Limited Scope for devices.
Business Type used	Business 2 Business(B2B) and Business 2 Consumer(B2C)	Business 2 Business (B2B)
Open API support	Supports Open API integrations.	There is no support for Open Api's
Examples	Smart wearable's, Big Data and Cloud, etc.	Sensors, Data and Information, etc.

▼ IoT vs WoT [Web of Things]

- While IoT and Web of Things both essentially serve the same purpose of connecting smart devices over the Internet, there are some key differences to keep in mind
- The main difference between IoT and WoT is the layer at which each establishes interconnectivity between devices. In this case, IoT solves just the network layer between devices.
- In contrast to IoT's network layer solutions, WoT can be thought of as the application layer. It sits on top of IoT conceptually and functionally. WoT is not an alternative or competitor to IoT; instead, it tries to enhance IoT.
- WoT enables devices to connect over the web using mainstream technologies and standards from a technical perspective—for example, HTML 5.0 and Javascript.
- In summary, WoT evolves IoT from a pure concept to a fully-developed architectural approach for smart device interaction.

▼ Describe IoT reference architecture and information model.

IoT reference architecture is a high-level blueprint that defines the interdependent components, systems, and devices involved in an IoT solution. It provides a framework that organizes and specifies the functions, layers, and data flows of an IoT system. The reference architecture typically consists of five layers:

1. Device layer: It consists of edge devices or sensors that collect data and interact with the physical environment.
2. Connectivity layer: Connects edge devices to networks and protocols that transmit data to remote locations.
3. Data management layer: It processes data collected from sensors and devices, stores data securely, and provides analytics for decision-making.
4. Application layer: It resides on top of the data management layer and provides user interfaces, dashboards, and services to interact with the IoT system.
5. Business layer: It encompasses the organization's core business processes and defines the objectives and outcomes of IoT solutions.

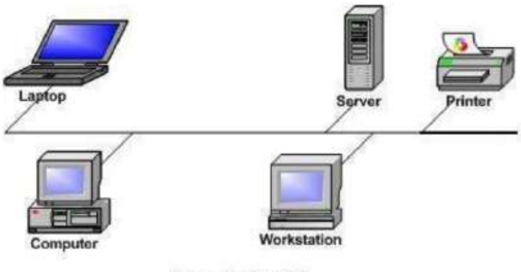
IoT information model, on the other hand, is a specific way of defining and managing data used in IoT systems. It includes a set of rules and specifications for managing the transfer, storage, and processing of data across IoT devices and systems. The information model describes the relationships between devices, sensors, and other components in the IoT ecosystem. The model also considers the data types, protocols, and information exchange mechanisms used to enable interoperability between devices and systems. The information model provides a standard way of organizing data structures and metadata for IoT solutions.

▼ **IoT Network configurations**

- IoT Network configuration is the process of setting a network's controls, flow and operation to support the network communication of an organization and/or network owner. This broad term incorporates multiple configuration and setup processes on network hardware, software and other supporting devices and components. In simple terms, the 4 Stage IoT Network architecture consists of -
    1. Sensors and actuators
    2. Internet getaways and Data Acquisition Systems
    3. Edge IT
    4. Data center and cloud.
- An IoT network refers to a collection of interconnected devices that communicate with other devices without the need for human involvement, such as autonomous cars, smart appliances, and wearable tech.
- To configure wireless IoT device to use a Programmable Wireless SIM requires only a few small configuration settings.
- Need to set the Programmable Wireless Access Point Name (APN) and need to use the Twilio Commands phone number.
- Depending on the device you are using, you may also be required to enter TCP and UDP network timer settings.

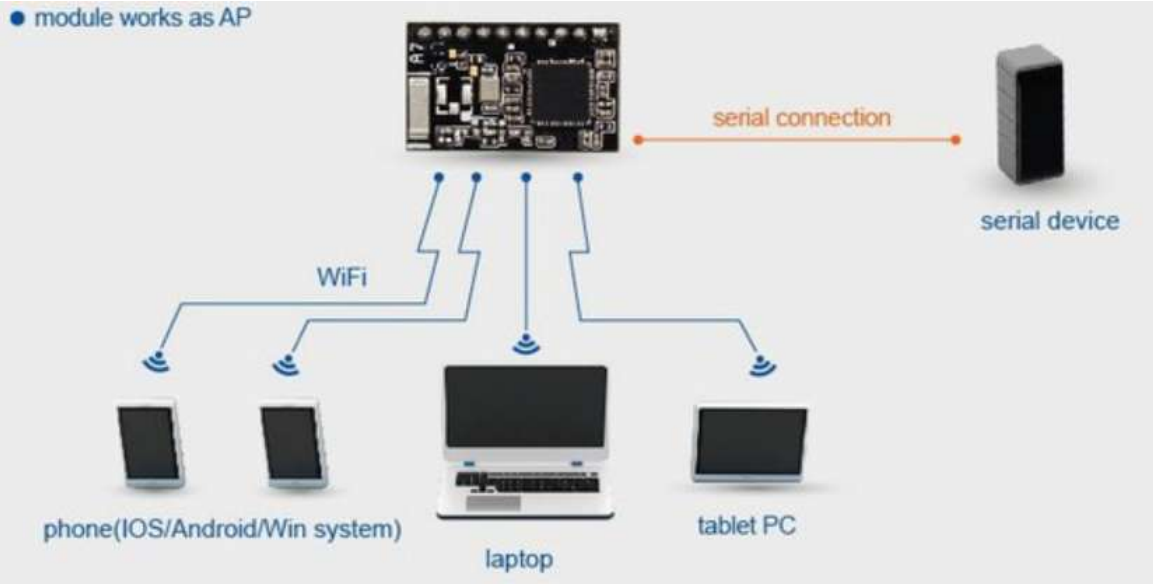
- Configure the Programmable Wireless APN
  - **Broadband IoT**
    The APN for the Programmable Wireless SIM is: **wireless.twilio.com**
    No authentication is required for this APN — leaves any username and password entries blank.
  - **Narrowband IoT**
    The APN for the Narrowband SIM, provided in partnership with T-Mobile USA, is: **iot.nb**
    No authentication is required for the NB-IoT APN — leaves any username and password entries blank.

▼ **IoT LAN**



A Typical LAN

- IoT is the same thing with smaller wireless devices designed for more specific purposes like Sensing or Electrical Control.

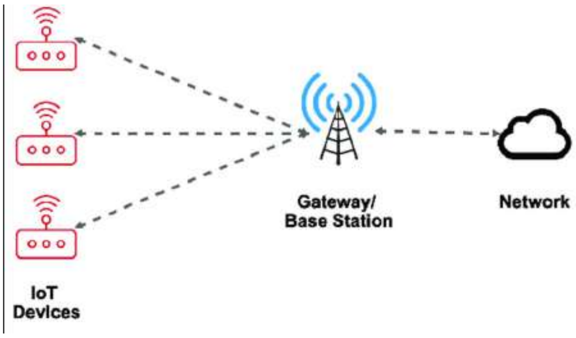


- In LAN networks (Local Network of a Router), the local IP addresses are usually something like 192.168.xxx.yyy . 'xxx' (0–254) can be user-defined, and 'yyy' (0–254) is the unique address given to each device on that network. The addresses are assigned by the Router's DHCP server.

- So if IoT device 1 is connected on 192.168.1.120, and IoT device 2 is on 192.168.1.200, sending a message from device 1 to device 2 is as simple as sending a GET request from device 1 to the device at 192.168.1.200 .
- Given the need for instant communication, most IoT products on LAN don't use HTTP any more, because of it's bulky nature. They go for more low-latency and optimised protocols that work with the TCP/IP stack.

▼ IoT WAN

- There are many low-power wide area (LPWA) radio technologies to choose from when deploying an Internet of Things (IoT) network. The final choice of radio technology, however, is only one of the many considerations when designing the low-power wide area network (LPWAN) where factors such as application type, network topology, total cost of ownership (TCO), reliability, security, and business model must also be considered.
- The LPWA radio technologies choices can be broadly subdivided into those that use the unlicensed ISM (Industrial Scientific and Medical) frequency bands and those that operate in the licensed frequency bands. Each of the LPWA technologies offers a choice or trade-off between transmit range, data rate, frequency, channel bandwidth, and power consumption.



▼ IoT Node

- In IoT, a node is a physical device that can be connected to other devices or networks for data exchange. IoT nodes are in the first category of elements, devices. IoT nodes must be able to fulfill at least one of these functions:
    - Collect data and send it to the internet via a communications interface;
    - Acting on the medium based on the data it receives from the Internet.
- The IoT node as we know it today, in its most minimal use case, can be a sensor embedded in an object that is never serviced again across the life of the device. They can be wireless and operated on a coin cell battery for years. What seemed impossible just a few years ago is now quickly becoming standard. And that's thanks to incredible innovations in low-power operation of wireless modules.
- The most numerous types of device in the IoT can be referred to as the node. These are all the exciting devices that are providing sensor data, or devices that are being controlled

from the cloud. This means things like door locks, security sensors, temperature sensors, and more.

- Laird Connectivity's BL654, for example, is a product that comes from a long line of Bluetooth modules that support Bluetooth Low Energy (BLE). BLE, introduced in the Bluetooth v4.0 specification, enables infrequent status-type messaging between Bluetooth devices with long sleep cycles in between messages.

▼ **IoT Gateway**

- An IoT gateway is a device that bridges the gap of communication that exists between IoT devices, equipment, sensors, and the cloud
- It systematically connects the cloud and the field by offering storage solutions to local processing
- An IoT gateway device is where all data traveling between IoT devices and the cloud is routed through
- It can independently regulate field devices based on the sensor's data input
- The gateway pre-processes the data before passing it along to cloud platforms, where the heavy lifting of transforming data into meaningful intelligence is accomplished[3].
- The IoT gateway is the central hub for sensors that collects their data, and they come in many forms. They interface directly with sensors and provide the path for that data to go to the cloud. Gateways can be designed to operate in so many ways that it can be hard to generalize.
- In some cases, they may listen passively, and the sensor operates without even knowing the gateway is there. In some cases, they may establish bidirectional communication with the sensor, allowing the sensor to be controlled by the cloud through the gateway. Most IoT devices communicate over either Wi-Fi, LTE, Bluetooth, or LoRaWAN.
- **Architectural Overview**
    - The following gateway architecture diagram is the most common architectural design where the gateway itself is not equipped with sensors. The gateway software installed on the device is responsible for collecting data from the sensor, pre-processing that data, and sending the results to the data centre.
    - Keep in mind that it is possible to have variations on this sensor architecture where some of the sensors are located at the gateway device, as illustrated in the following diagram.

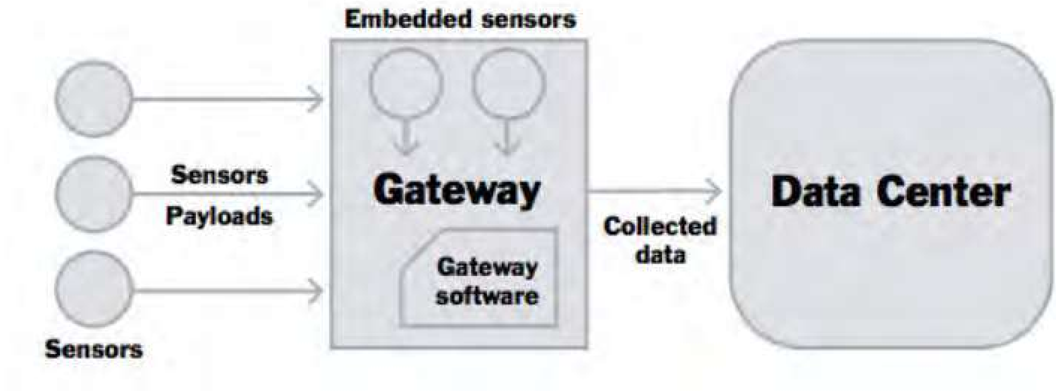


Figure 1.7: Gateway Architecture Diagram

- **IoT Proxy**

  - The IoT proxy is a server as well as a client between the IoT client and Work Space Protocol (WSP). The IoT proxy has the RD functionalities for registering information of resources which expose services in the network and discovering the information by IoT clients.
  - A proxy server, also known as a "proxy" or "application-level gateway", is a computer that acts as a gateway between a local network (for example, all the computers at one company or in one building) and a larger-scale network such as the internet. Proxy servers provide increased performance and security.

- **Review of Basic Microcontrollers and interfacing**

- **Draw and explain IoT level 3 and level 4 system. Give suitable examples for them.**

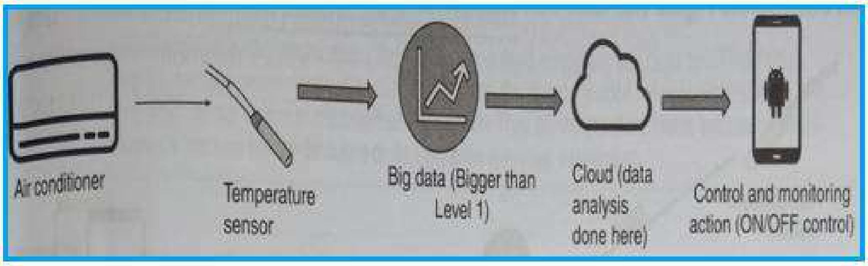
IoT Level 1

- This level consists of air conditioner, temperature sensor, data collection and analysis and control & monitoring app.
- The **data sensed in stored locally.**
- **The data analysis is done locally**.
- Monitoring & Control is done using Mobile app or web app.
- The data generated in this level application is not huge.
- All the control actions are performed through internet.
- **Example:** Room temperature is monitored using temperature sensor and data is stored/analysed locally.

IoT Level 2

- This level consists of air conditioner, temperature sensor, Big data (Bigger than level -1, data analysis done here), cloud and control & monitoring app.

- This level-2 is complex compare to level-1. Moreover rate of sensing is faster compare to level-1.
- This level has voluminous size of data. Hence **cloud storage is used.**
- **Data analysis is carried out locally**. Cloud is used for only storage purpose.
- Based on data analysis, control action is triggered using web app or mobile app.
- **Examples:** Agriculture applications, room freshening solutions based on odour sensors etc.


Air conditioner
Temperature sensor
Big data (Bigger than Level 1)
Cloud (data analysis done here)
Control and monitoring action (ON/OFF control)

IoT Level 3

- As shown in the figure, this level consists of air conditioner, temperature sensor, big data collection (Bigger than level-1), cloud (for data analysis) and control & monitoring app.
- **Data here is voluminous i.e. big data**. Frequency of data sensing is fast and collected sensed data is stored on cloud as it is big.
- **Data analysis is done on the cloud** side and based on analysis control action is triggered using mobile app or web app.
- **Examples:** Agriculture applications, room freshening solutions based on odour sensors etc.

IoT Level 4

- This **level consists of multiple sensors**, data collection and analysis and control & monitoring app.
- At this level-4, multiple sensors are used which are independent of the others.
- The **data collected using these sensors are uploaded to the cloud separately**. The cloud storage is used in this level due to requirement of huge data storage.
- The **data analysis is performed on the cloud** and based on which control action is triggered either using web app or mobile app.

IoT Level 5

- This level consists of multiple sensors, coordinator node, data collection and analysis and control & monitoring app.
- This level is similar to level-4 which also has huge data and hence they are sensed using multiple sensors at much faster rate and simultaneously.
- The data collection and data analysis is performed at the cloud level.

- Based on analysis,control action is performed using mobile app or web app.

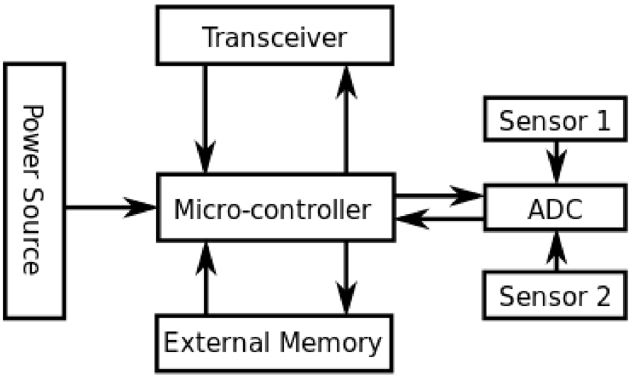
Unit II

▼ Define Sensor

- A sensor is a device that detects and responds to some type of input from the physical environment.
- The input can be light, heat, motion, moisture, pressure or any number of other environmental phenomena.
- The output is generally a signal that is converted to a human-readable display at the sensor location or transmitted electronically over a network for reading or further processing.

▼ Basic components and challenges of a sensor node

- A **sensor node**, consists of an individual node from a sensor network that is capable of performing a desired action such as gathering, processing or communicating information with other connected nodes in a network.



The typical architecture of the sensor node

- The main components of a sensor node usually involve a **microcontroller, transceiver, external memory, power source and one or more sensors.**

Sensors	used by wireless sensor nodes to capture data from their environment. They are hardware devices that produce a measurable response to a change in a physical condition like temperature or pressure
Controller	controller performs tasks, processes data and controls the functionality of other components in the sensor node. While the most common controller is a microcontroller
Transceiver	The functionality of both

	transmitter and receiver are combined into a single device known as a transceiver.
External memory	From an energy perspective, the most relevant kinds of memory are the on-chip memory of a microcontroller and Flash memory—off-chip RAM is rarely, if ever, used
Power source	Power is stored either in batteries or capacitors

- **Challenges of a sensor node**
  - Sensor networks do not fit into any regular topology, because while deploying the sensor nodes they are scattered
  - Very limited resources
  - It comes under fewer infrastructures and also maintenance is very difficult
  - **Challenges in power management:**
Sensor node relies only on battery and it cannot be recharged or replaced. Hardware design for sensor node should also be considered.
  - Node failure, topology changes and adding of nodes and deletion of nodes is another challenging issue.

▼ **Sensor features**

- A sensor is a device that can detect changes in an environment.
- By itself, a sensor is useless, but when we use it in an electronic system, it plays a key role.
- A sensor can measure a physical phenomenon (like temperature, pressure, and so on) and transform it into an electric signal.
- These three features should be at the base of a good sensor:
  - It should be sensitive to the phenomenon that it measures.
  - It should not be sensitive to other physical phenomena.
  - It should not modify the measured phenomenon during the measurement process.
- There is a wide range of sensors we can exploit to measure almost all the physical properties around us.
- A few common sensors that are widely adopted in everyday life include thermometers, pressure sensors, light sensors, accelerometers, gyroscopes, motion sensors, gas sensors, and many more. A sensor can be described using several properties, the most important being:
  - **Range:** The maximum and minimum values of the phenomenon that the sensor can measure.

- **Sensitivity:** The minimum change of the measured parameter that causes a detectable change in output signal.
- **Resolution:** The minimum change in the phenomenon that the sensor can detect.

▼ **Sensor resolution**

- The resolution of a sensor is the smallest change it can detect in the quantity that it is measuring.
- The resolution is related to the precision with which the measurement is made, but they are not the same thing.
- Resolution is an important specification because without sufficient resolution you may not be able to reliably make the needed measurement and an over-performing sensor will burden your budget.
- Resolution is only meaningful within the context of the system bandwidth, the application, and the measurement method and unit of measure used by the sensor manufacturer.
- A simple “resolution spec” in a datasheet rarely provides enough information for a fully informed sensor selection. Understanding this important specification will empower you to confidently make the right displacement sensor choice.
- The resolution is also not accurate. An inaccurate sensor could have high resolution, and a low-resolution sensor may be accurate in some applications.
- It is not the least significant digit in a display or the least significant bit in a conversation between the digital and analog worlds.
- Digital devices have a resolution specification based on the least significant bit, and if insufficient, may further degrade the overall sensor resolution, but the fundamental limit of a sensor’s resolution is determined in the analog world
- **The battle for higher resolutions in sensor design is primarily a fight against electrical noise.**

▼ **Sensor classes: Analog, Digital, Scalar, Vector Sensors**

Classification of Sensors

Category of the sensor in the form of classes are:

1. **Analog**
   a. These types of sensors produce a **continuous o/p signal** which is normally proportional to the quantity which is being measured.
   b. Generally, physical quantities are temperature, pressure, speed, orientation, displacement is the type of analog quantity.
2. **Digital**
   a. These types of sensors give the output in discrete form, which is in the form of voltage.
   b. This sensor produces the output in **True/False or ON/OFF or 0/1** depending on the application.

   c. The advantage of this type of output is, we can store the output.

3. **Scalar**
   a. The sensors produce the signal which is **proportional to the magnitude of the quantity** being measured.
   b. The signal like temperature, color, strain, and pressure are comes under this type of quantity.
   c. E.g., the temperature of a place can be measured using a sensor, which response to temperature changes irrespective of the orientation of the sensor.
4. **Vector**
   a. These sensors produce a signal which is proportional to the magnitude, direction, or orientation.
   b. Physical quantity just like sound, image, orientation, and velocity are all vector quantities. So **here only magnitude is not sufficient to convey the complete information**,
   c. e.g. the acceleration of the car can be measured using an acceleration sensor by calculating x, y, and z axes.

▼ **Sensor**

IoT platforms function and deliver various kinds of intelligence and data using a variety of sensors. They serve to collect data, push it and share it with a whole network of connected devices. All this collected data makes it possible for devices to autonomously function, and the whole ecosystem is becoming “smarter” every day. By combining a set of sensors and a communication network, devices share information and are improving their effectiveness and functionality. Let’s take a look at some of the key sensors, extensively being used in the IoT world.

▼ **Sensor Types**

1. **IR sensor**
   a. IR (infrared) sensor is an electronic device that emits light to sense some object of the surroundings.
   b. An IR sensor can measure the heat of an object as well as to detect motion.
   c. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation.
   d. These types of radiations are invisible to our eyes, but the infrared sensor can detect these radiations.
   e. The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode.
   f. **Types of IR Sensor**
      i. **Active Infrared Sensor**
      Active infrared sensors consist of two elements: an infrared source and an

infrared detector

    ii. **Passive Infrared Sensor**
    Passive infrared sensors are Infrared detectors. Passive infrared sensors do not use an infrared source

2. **Gyroscope**
   sensor or device which is used to measure the angular rate or angular velocity
3. **Temperature sensor**
4. **Proximity sensor**
5. **Pressure sensor**
6. **Water quality sensor**
7. **Motion detection sensor**
8. **Smoke sensor**

▼ **Sensor bias**

- Sensor bias is a phenomenon where sensors or measurement instruments produce a consistent error or deviation in their readings or measurements, resulting in inaccurate or biased data. A sensor may be biased if it is calibrated incorrectly, or if it is affected by external factors such as temperature, humidity, or electromagnetic interference.
- Sensor bias can be corrected through calibration or adjustment of the sensor, or by using statistical methods to estimate and correct the bias in the data.

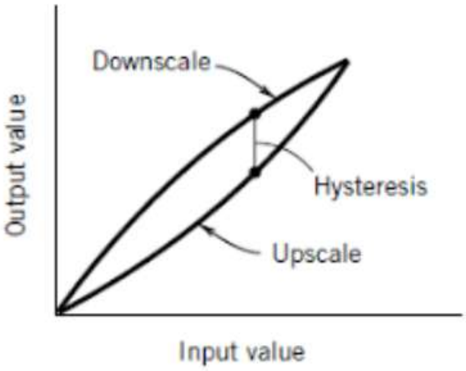
▼ **Sensor drift**

- Sensor drift refers to the gradual deviation of a sensor's accuracy from its original calibration over time.
- This is the low-frequency change in a sensor with time
- In other words, sensors may become less accurate as they age or are subjected to environmental factors.
- This drift can be caused by changes in temperature, humidity, or other environmental factors that affect the sensor's behavior. As a result, sensor readings may become increasingly unreliable, leading to errors in data collection, analysis, and decision-making.
- It is important to periodically recalibrate sensors to ensure they remain accurate and reliable.

▼ **Sensor Hysteresis error**

- Hysteresis error is the difference in the output values of a sensor for the same input value, depending on whether the input value is increasing or decreasing.
- In other words, the sensor produces different responses for an increasing and decreasing input signal, resulting in a nonlinear relationship between the input and output values.

- This error is caused by the physical characteristics of the sensor, such as mechanical friction, magnetic attraction or hysteresis in the material, that affect the response of the sensor to changes in the input signal.
- Hysteresis error can reduce the accuracy and precision of the sensor, and can be corrected by using calibration techniques.



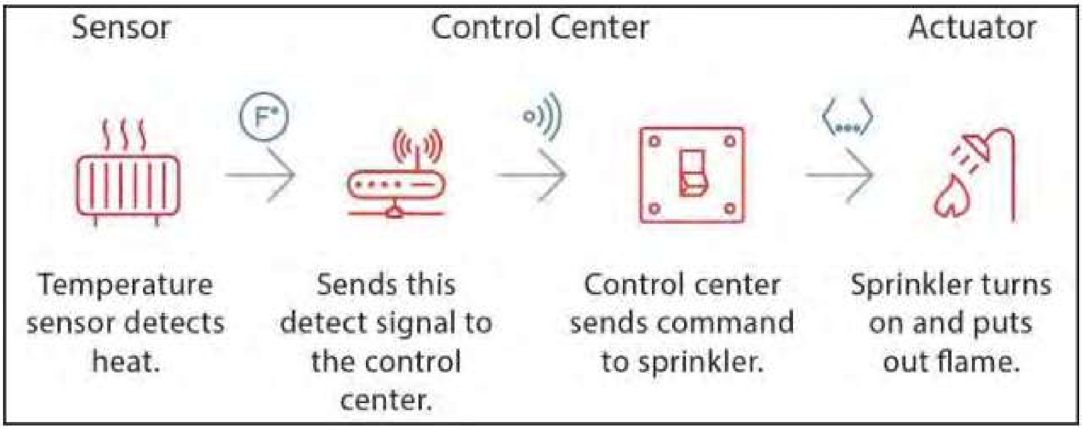
▼ **Sensor quantization error**

- Sensor quantization error refers to the error that occurs when analog signals from sensors are converted into discrete digital signals.
- This type of error occurs because the analog signals are continuous and infinite, while digital signals are discrete and finite.
- When signals are quantized, the information from the analog signal is lost or distorted, which creates an error in the sensor reading.
- The quantization error increases with the number of bits used to represent the analog signal. **Lower numbers of bits mean that there are fewer possible values for the output signal, which results in a larger quantization error.**
- For example, an 8-bit analog-to-digital converter (ADC) can only represent 256 possible values, while a 16-bit ADC can represent 65,536 possible values.
- **The quantization error can be reduced by increasing the number of bits used in the quantization process or by using a higher resolution sensor.**
- However, this may not always be practical or feasible for certain applications.
- **In summary, quantization error is an inherent limitation of digital sensors and must be taken into account when analyzing or interpreting sensor data.**

▼ **Actuator**

An actuator operates in the reverse direction of a sensor. It takes an electrical input and turns it into physical action.

For instance, an electric motor, a hydraulic system, and a pneumatic system are all different types of actuators.


Sensor
Control Center
Actuator
F°
Temperature sensor detects heat.
Sends this detect signal to the control center.
Control center sends command to sprinkler.
Sprinkler turns on and puts out flame.

- ▼ **Actuator types:**
  - ▼ **Hydraulic**
    - This actuator converts mechanical motion into linear, rotary, or oscillatory motion
    - The hydraulic actuator consists of a cylinder or fluid motor which uses hydraulic power to help mechanical operation
    - Example: Hydraulic brake in a vehicle.
  - ▼ **Pneumatic**
    - converts energy formed by vacuum or compressed air at high pressure into linear or rotary motion
    - responsible to convert pressure into force
    - **Advantages**
      - Pneumatic energy responds quickly to start and stop signals
      - Pneumatic actuators produce large forces from relatively small pressure changes
  - ▼ **electrical**
    - Pneumatic actuators produce large forces from relatively small pressure changes
    - **Advantages:** cheap, clean, speedy type of actuator.
    - **Examples:** Solenoid based electric bell-ringing mechanism
  - ▼ **thermal/magnetic**
    - This actuator can be actuated by the application of thermal or magnetic energy
    - This actuator uses shape-memory materials e.g. shape memory alloys.
    - **Advantages:** Compact, light in weight, economical, offers high power density.
    - **Examples:** Thermal actuator is thermostat; magnetic actuator is an electromagnet.
  - ▼ **mechanical actuators**
    - A mechanical actuator is a device that converts energy into mechanical motion.

- It is a component that is typically used to actuate or drive mechanical systems, such as valves, switches, or other types of mechanisms.

▼ **soft actuators**

- Soft actuators refer to devices that are able to produce controlled motion or force in a soft or flexible manner.
- Soft actuators have a variety of applications, including in soft robotics, medical devices, and wearable technology.
- Some common types of soft actuators include pneumatic and hydraulic actuators, shape memory alloys, and electroactive polymers.

▼ **Discuss the building blocks of IoT. What are the most used sensors types in IoT?**

The building blocks of IoT include:

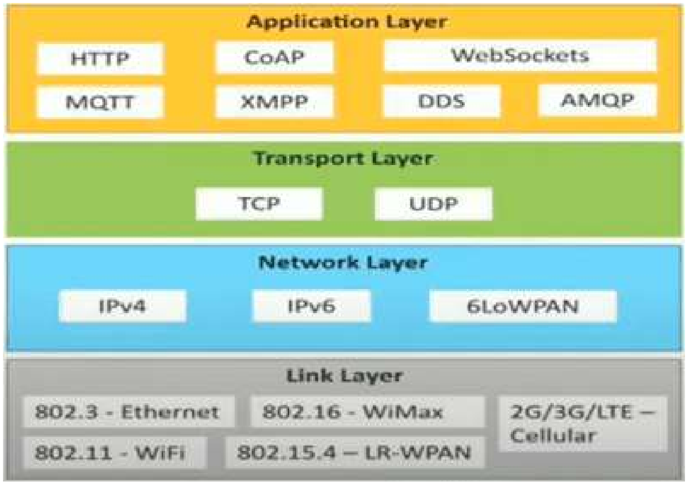
1. **Sensors:** They are the essential building blocks of IoT, which are used to gather data from the physical environment.
2. **Connectivity:** It includes a network that connects sensors and devices to the cloud or a central server, including Bluetooth, Wi-Fi, cellular and Satellite.
3. **Data processing and storage:** Data captured by sensors and devices are processed, analyzed, and stored in a central location like the cloud or data center.
4. **Actuators:** These are devices that receive instructions from the system and perform actions, including LEDs, motors, and relays.
5. **User interface:** This includes the dashboard, mobile applications, and other interfaces, which are used by end-users to access and monitor IoT devices.

Some of the most commonly used sensors in IoT applications include:

1. **Temperature Sensors:** These sensors measure temperature in the environment and are widely used in home automation, environmental monitoring, and industrial applications.
2. **Proximity Sensors:** These sensors detect the presence of objects without any physical contact and are commonly used in security systems, automated doors, and elevators.
3. **Light Sensors:** These sensors detect the light intensity and are used in lighting control systems and smart buildings.
4. **Pressure Sensors:** These sensors detect pressure and are used in various applications, including industrial automation, medical devices, and automotive.
5. **Motion Sensors:** They detect motion and are commonly used in home security systems, wearable devices, and gaming systems.
6. **Humidity Sensors:** These sensors measure the moisture level in the environment and are widely used in HVAC systems, medical devices, and agriculture.

Unit III

▼ **IoT Networking**



LnT A

What is the use of application layer protocols?

- Application layer protocols are used to facilitate communication between different applications on the internet. These protocols define the syntax, semantics, and synchronization of communication between networked devices, and enable the exchange of data between applications.
    - **HTTP**
      Hypertext transfer protocol is a protocol that present in application layer for transmitting media documents. it is used to **communicate between web browsers and servers.** it makes a request to a server and then waits till it receives a response and in between the request server does not keep any data between two requests.
    - **WebSocket**
      This protocol enables **two-way communication** between a client and a host that can be run on an untrusted code in a controlled environment. this protocol is commonly used by web browsers.
    - **MQTT**
      It is a machine-to-machine connectivity protocol that was designed as a **publish/subscribe** messaging transport. and it is used for remote locations where a small code footprint is required.
- **Transport Layer**
  This layer is used to control the flow of data segments and handle the error control. also, these layer protocols provide end-to-end message transfer capability independent of the underlying network.
    - **TCP**
      The transmission control protocol is a protocol that defines how to establish and maintain a network that can exchange data in a proper manner using the internet protocol.
    - **UDP**
      A user datagram protocol is a part of internet protocol called the connectionless

protocol. this protocol not required to establish the connection to transfer data.

- **Network Layer**
  This layer is used to send datagrams from the source network to the destination network. we use IPv4 and IPv6 protocols as a host identification that transfers data in packets.
  - **IPv4**
    This is a protocol address that is a unique and numerical label assigned to each device connected with the network. an IP address performs two main functions host and location addressing. IPv4 is an IP address that is 32 bit long.
  - **IPv6**
    It is a successor of IPv4 that uses 128 bits for an IP address. it is developed by the IETF task force to deal with the long- anticipated problems.
- **Link Layer**
  Link-layer protocols are used to send data over the network's physical layer. it also determines how the packets are coded and signaled by the devices.
  - **Ethernet**
    It is a set of technologies and protocols that are used primarily in LANs. it defines the physical layer and the medium access control for wired ethernet networks.
  - **WiFi**
    It is a set of LAN protocols and specifies the set of media access control and physical layer protocols for implementing wireless local area networks.

▼ **IoT Components**

The fundamental components of IoT system are:

1. **Sensors/Devices**
   a. First, sensors or devices help in collecting very minute data from the surrounding environment. All of this collected data can have various degrees of complexities ranging from a simple temperature monitoring sensor or a complex full video feed. A device can have multiple sensors that can bundle together to do more than just sense things. For example, our phone is a device that has multiple sensors such as GPS, accelerometer, camera but our phone does not simply sense things. The most rudimentary step will always remain to pick and collect data from the surrounding environment be it a standalone sensor or multiple device
2. **Connectivity**
   a. Collected data is sent to a cloud infrastructure but it needs a medium for transport. The sensors can be connected to the cloud through various mediums of communication and transports such as cellular networks, satellite networks, Wi-Fi, Bluetooth, wide-area networks (WAN), low power wide area network and many more. Every option we choose has some specifications and trade-offs between power consumption, range, and bandwidth. So, choosing the best connectivity option in the IOT system is important.
3. **Data Processing**

    a. Once the data is collected and it gets to the cloud, the software performs processing on the acquired data. This can range from something very simple, such as checking that the temperature reading on devices such as AC or heaters is within an acceptable range. Figure 3.9: Components of the Internet of Things

4. **User Interface**

    a. User sometimes might also have an interface through which they can actively check in on their IoT system. For example, a user has a camera installed in his house; he might want to check the video recordings and all the feeds through a web server. However, it’s not always this easy and a one-way street. Depending on the IoT application and complexity of the system, the user may also be able to perform an action that may backfire and affect the system. For example, if a user detects some changes in the refrigerator, the user can remotely adjust the temperature via their phone. There are also cases where some actions perform automatically. By establishing and implementing some predefined rules, the entire IOT system can adjust the settings automatically and no human has to be physically present. Also, in case if any intruders are sensed, the system can generate an alert not only to the owner of the house but to the concerned authorities.

▼ **Functional components of IoT**

There are five key Functional components of IoT, things, gateways, mobile devices, the cloud and the enterprise as described below:

1. **Things**

    a. Physical objects things that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. Things can also be self-sufficient and communicate to the internet for only centralized coordination and analysis.

2. **Gateways**

    a. Gateways provide the application logic, store data and communicate with the internet for the things that are connected to it. Things don’t have to be as smart, because the gateway can provide these resources.

3. **Smartphone’s**

    a. Smart phones (or any mobile device) may house the application logic, store data and communicate with the internet on behalf of things that are connected to it. Things don’t have to be as smart, because the smart phone provides these resources.

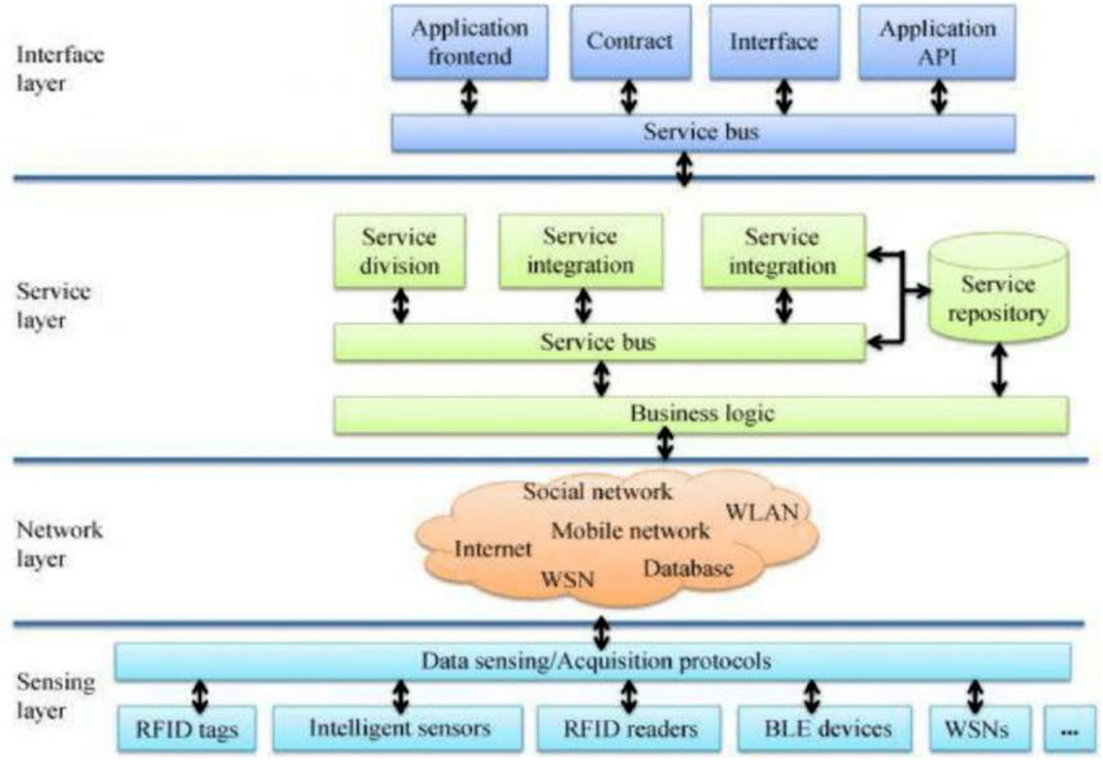
4. **The Cloud**

    a. The cloud can act as the central connection hub, power analytics and provision data storage. Things don’t have to be as smart, because the cloud will provide these resources.

5. **The Enterprise**

    a. This architectural role is focused on keeping connected machines, application logic, and analytics and data storage on- premises that is, behind the enterprise firewall.

- ▼ **IoT service oriented architecture**
  - A service-oriented architecture is an approach used to create an architecture based upon the use of services.
  - Services (such as RESTful Web services) carry out some small function, such as producing data, validating a customer, or providing simple analytical services. The service-oriented architecture is a widely used design pattern. It effectively combines individual units of software to provide higher level of functionality. The communication involves either simple data passing, or it could involve two or more services coordinating some activity. If a service-oriented architecture is to be effective, we need a clear understanding of the term service. A service is a function that is well defined, self-contained, and does not depend on the context or state of other services. SOA provides a strategic capability for integrating business processes, data, and organizational knowledge. SOA is governed by a well-defined set of frameworks. Service composition and service discovery are the two major elements of SOA.



- ▼ **IoT challenges**
  1. **Security**
     a. **Lack of encryption**
     b. **Insufficient testing and updating**
     c. **Brute forcing and the risk of default passwords**
     d. **IoT Malware and ransomware**
     e. **Inadequate device security**

    f. **Vulnerability to network attacks**

    g. The most significant and arguably unsolvable challenge relates to security or, more specifically, cyber security as it pertains to information technology. It's not just the data connection that is vulnerable, but everything connected to the actual hardware. Imagine a smart manufacturing unit equipped with IoT sensors. A property manager or maintenance associate can use a mobile device to check the device status, read incoming data or send commands. But what if a foreign attacker were to seize control of the machine using a known vulnerability or weak security measures to gain access.

2. **Privacy**

    a. Most online connections are secured using a form of encryption. Means the data is locked behind a software key and cannot be decrypted translated without the appropriate authorization. Some of the more basic forms of encryption are easy to break, but they can still take a long time, slowing down any nefarious parties. Ultimately, it means that encrypted data is, and always will be, exponentially more secure than raw, unprotected data.

3. **Resource**

    a. Consumption Electronics require energy to operate and IoT devices are no exception. They must actively transmit data 24/7, which means support from other technologies, including network adapters, gateways and more. Beyond just electricity, data requires physical storage. Even with cloud and edge computing solutions, there is still a remote server connected to the network that is being used to house the digital content. Servers require an excessive amount of energy, as do data centers that require large-scale cooling systems to operate under heavy loads.

4. **Design challenge in IoT**

    a. **Interoperability:** Interoperability refers to the ability of different systems, devices, or components to work together seamlessly and exchange data effectively.

    b. **Scalability**

    c. **Reliability**

    d. **Power consumption**

5. **Deployment challenges in IoT**

    a. **Connectivity**

    b. **Cross platform capability**

    c. **Data collection and processing**

▼ How might Internet Address (IPv6) affect the development and implementation of the Internet of Things?

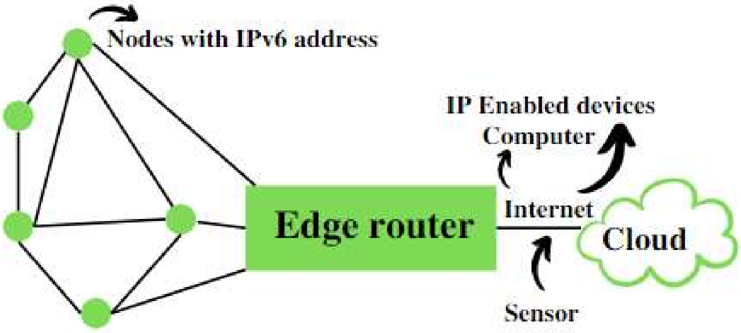
IPv6 (Internet Protocol version 6) is an upgraded version of the current IPv4 that provides more address space and enhanced security features. Its impact on the development and implementation of the Internet of Things (IoT) can be seen in the following ways:

1. **Improved address capacity:** With IPv6, the number of possible IP addresses increases significantly, making it possible to accommodate the billions of devices expected to be connected through the IoT.
   32bits—>128bits
2. **Simplified network architecture:** IPv6 has a simplified network architecture that allows for easier routing of IoT devices, resulting in smoother communication between devices.
3. **Enhanced security:** IPv6 has built-in security features that make it more secure than its predecessor IPv4, making it more suitable for IoT deployments.
4. **Impact on IoT devices and manufacturers:** IoT device manufacturers need to ensure that their devices are IPv6 compatible, and the deployment of IPv6 networks may require changes to current IoT device configurations.
5. **Opportunities for new services and applications:** The increased capacity and improved security of IPv6 provide opportunities for new IoT services and applications that were not possible before.

In summary, IPv6 provides numerous benefits for the development and implementation of the IoT, including improved address capacity, simplified network architectures, enhanced security, and new opportunities for services and applications.

▼ **6LowPAN**

- 6LoWPAN is an IPv6 protocol, and It’s extended from is IPv6 over Low Power Personal Area Network.
- As the name itself explains the meaning of this protocol is that this protocol works on Wireless Personal Area Network i.e., WPAN.
- 6LoWPAN provides the upper layer system for use with low power wireless communications for IoT and M2M, originally intended for 802.15.4, it is now used with many other wireless standards.
- The 6LoWPAN system is used for a variety of applications including wireless sensor networks. This form of wireless sensor network sends data as packets and using IPv6 - providing the basis for the name - IPv6 over Low power Wireless Personal Area Networks.
- It comprises an Edge Router and Sensor Nodes. Even the smallest of the IoT devices can now be part of the network, and the information can be transmitted to the outside world as well. For example, LED Streetlights.



Basic Requirements of 6LoWPAN:

1. The device should be having sleep mode in order to support the battery saving.
2. Minimal memory requirement.
3. Routing overhead should be lowered.

Features of 6LoWPAN:

1. It is used with IEEE 802.15,.4 in the 2.4 GHz band.
2. Outdoor range: ~200 m (maximum)
3. Data rate: 200kbps (maximum)
4. Maximum number of nodes: ~100

Advantages of 6LoWPAN:

1. 6LoWPAN is a mesh network that is robust, scalable, and can heal on its own.
2. It delivers low-cost and secure communication in IoT devices.
3. It uses IPv6 protocol and so it can be directly routed to cloud platforms.
4. It offers one-to-many and many-to-one routing.
5. In the network, leaf nodes can be in sleep mode for a longer duration of time.

Disadvantages of 6LoWPAN:

1. It is comparatively less secure than Zigbee.
2. It has lesser immunity to interference than that Wi-Fi and Bluetooth.
3. Without the mesh topology, it supports a short range.

Applications of 6LoWPAN:

1. It is a wireless sensor network.
2. It is used in home-automation,
3. It is used in smart agricultural techniques, and industrial monitoring.

Security and Interoperability with 6LoWPAN:

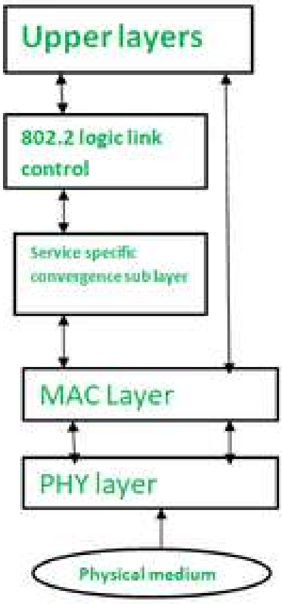
- **Security**: 6LoWPAN security is ensured by the AES algorithm, which is a link layer security, and the transport layer security mechanisms are included as well.
- **Interoperability:** 6LoWPAN is able to operate with other wireless devices as well which makes it interoperable in a network.

▼ **What is IEEE 802.15.4 protocol ? How is it related to IoT?**

- IEEE 802.15.4 is a low-cost, low-data-rate wireless access technology for devices that are operated or work on batteries.
- This describes how low-rate wireless personal area networks (LR-WPANs) function.
- **IEEE 802.15.4e:**
  - 802.15.4e for industrial applications and 802.15.4g for the smart utility networks (SUN)
  - **The 802.15.4e improves the old standard by introducing mechanisms such as time slotted access, multichannel communication and channel hopping.**
- Properties:

  Standardization and alliances: It specifies low-data-rate PHY and MAC layer requirements for wireless personal area networks (WPAN).IEEE 802.15. Protocol Stacks include:

  - **ZigBee:** ZigBee is a Personal Area Network task group with a low rate task group 4. It is a technology of home networking. ZigBee is a technological standard created for controlling and sensing the network. As we know that ZigBee is the Personal Area network of task group 4 so it is based on IEEE 802.15.4 and is created by Zigbee Alliance.
  - **6LoWPAN:** The 6LoWPAN system is used for a variety of applications including wireless sensor networks. This form of wireless sensor network sends data as packets and uses IPv6 – providing the basis for the name – IPv6 over Low power Wireless Personal Area Networks.



The architecture of LR-WPAN Device

Advantages of IEEE 802.15.4:

IEEE 802.15.4 has the following advantages:

- cheap cost
- long battery life,
- Quick installation
- simple
- extensible protocol stack

Disadvantages of IEEE 802.15.4:

IEEE 802.15.4’s drawbacks include:

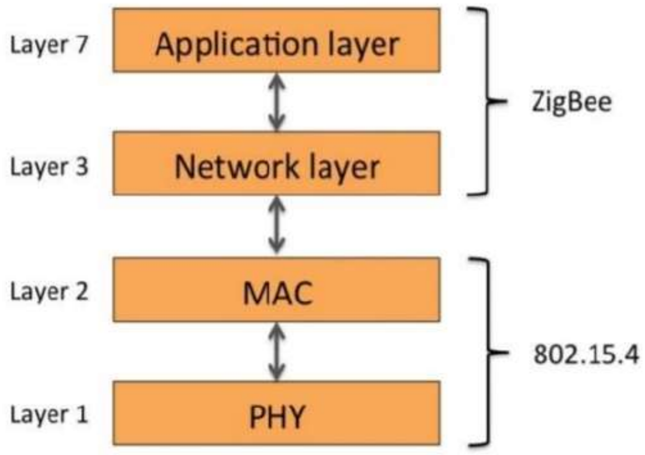
- IEEE 802.15.4 causes interference and multipath fading.
- doesn’t employ a frequency-hopping approach.
- unbounded latency
- interference susceptibility

Applications of IEEE 802.15.4:

IEEE 802.15.4 Applications:

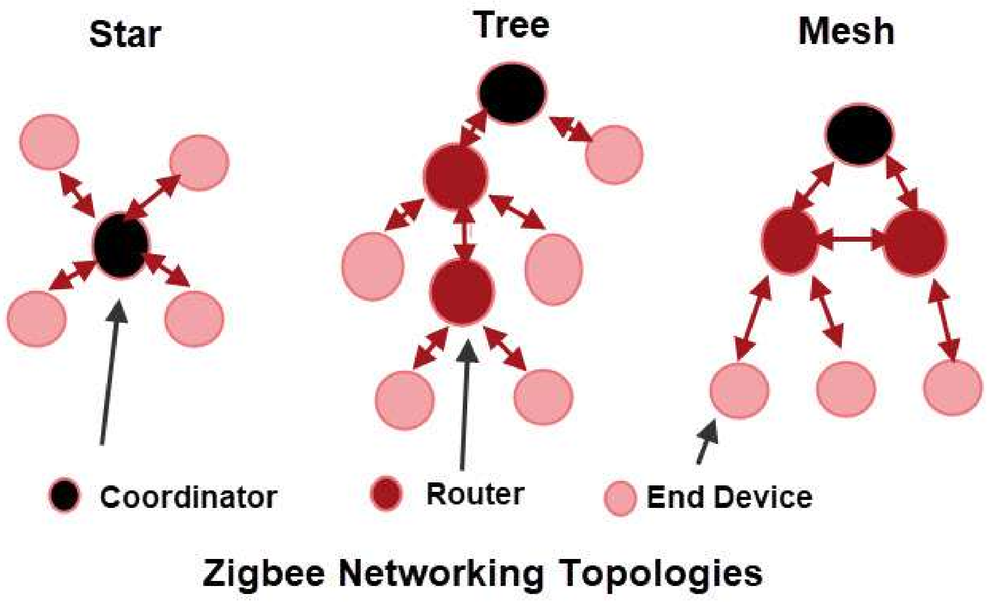
- Wireless sensor networks in the industry
- Building and home automation
- Remote controllers and interacting toys
- Automotive networks

▼ ZigBee and its types



- Zigbee is an IEEE 802.15.4 based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection.
- Hence, Zigbee is a low-power, low data rate, and close proximity (i.e., personal area) wireless ad hoc network. The technology defined by the Zigbee specification is intended to be simpler and less expensive than other wireless personal area networks (WPANs), such as Bluetooth or more general wireless networking such as Wi-Fi.
- Applications include wireless light switches, home energy monitors, traffic management systems, and other consumer and industrial equipment that require short-range low-rate wireless data transfer.
- **Nodes**
  - Zigbee Co-ordinator
  - Zigbee Router
  - Zigbee End devices
- **Types of ZigBee devices**
  - **ZigBee coordinator (ZC):** The most capable device, the coordinator forms the root of the network tree and might bridge to other networks. There is precisely one ZigBee coordinator in each network since it is the device that started the network originally
    - It stores information about the network, including acting as the trust center and repository for security keys.
  - **ZigBee router (ZR):** As well as running an application function, a router can act as an intermediate router, passing data on from other devices.
  - **ZigBee end device (ZED):** Its functionality to talk to the parent node (either the coordinator or a router) it cannot relay data from other devices. This relationship allows the node to be asleep a significant amount of the time thereby giving long battery life.

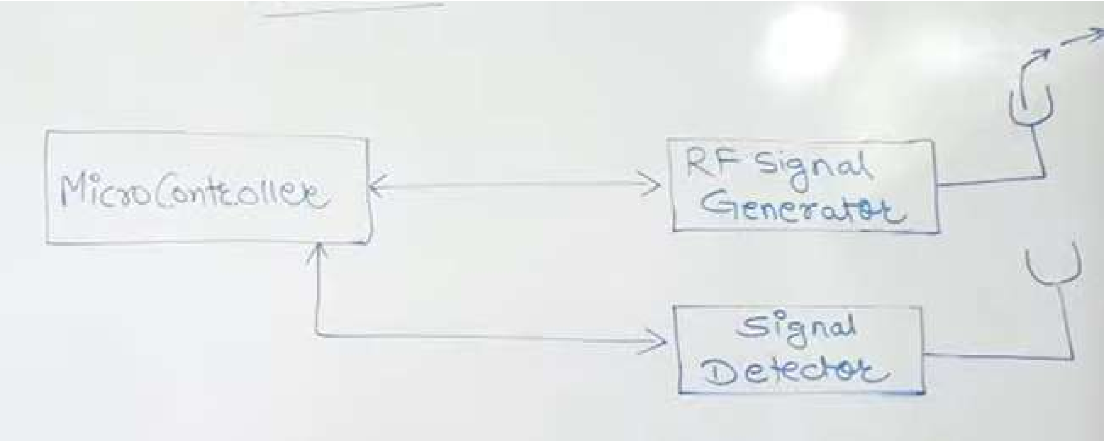
        - A ZED requires the least amount of memory and thus can be less expensive to manufacture than a ZR or ZC.



▼ **RFID**

- RFID (Radio Frequency Identification) technology is a system that uses radio waves to identify and track objects. It works by placing a small electronic tag, which contains a microchip and an antenna, on an object. The tag emits a radio signal that can be detected by an RFID reader, which can be placed at a fixed location or carried by a person.
- RFID technology is used in a variety of industries, including retail, healthcare, manufacturing, and logistics. It can help improve inventory management, supply chain visibility, and asset tracking.
- RFID tags can also be used in access control systems, vehicle tracking, and identification and authentication systems.
- One of the key advantages of RFID technology is that it allows for automated and real-time data collection, which can help improve efficiency and reduce errors. However, there are also concerns about the privacy and security implications of RFID, as the technology can be used to track and monitor individuals and their movements.
- **2 types**
    - Passive RFID
        - NO own power supply
        - dependent on RFID reader for power
    - Active RFID
        - has own power supply

- **AIDC (Automatic identification and data collection)** operates through wired communication but;
  - RFID does this through radio frequencies wirelessly



▼ RFID Features

The main features of RFID are as follows:

1. **Able to Read and Write data without direct contact**
   a. The RF tag can contain up to several kilobytes of rich information. All of the data required for each process (process history, inspection history etc) can be freely stored, without the need for direct contact. This makes it possible to develop paperless sites, where the causes of production stop are reduced.
2. **Highly pliable and reliable system configuration**
   a. With the technology to decentralize information, the load on higher systems is reduced. This means that system development costs can also be reduced, systems can be implemented significantly faster, and the system is much more flexible when making changes. Also, "the unification of items with their information" for each process and site can make it possible to manage production/processes and product quality without errors. And, with the latest information contained in RF tags, work can continue offline in emergencies, significantly shortening the time required to restore processes.
3. **Adoption of space transmission technology and protocols**
   a. As opposed to barcodes which simply look for 1 or 0, advanced space transmission technologies and specialized protocols are employed for transmission through the air. 16 bits CRC is added to the information as it is transmitted. More than 18 bits Burst errors can be detected at a ratio of 00.9985%, providing a very high reliability in the transfer. Also, since there are no mechanical devices involved such as with the raster scan method for barcodes, the likelihood of malfunction and other problems is greatly reduced.
4. **Electric and electromagnetic wave transmission**

    a. Unlike barcodes, since communication occurs by means of electric and electromagnetic waves, erroneous readings due to dirt, moisture, oil etc are cancelled out. Even if there is dust, moisture etc., or anything other than metal between the antenna and the RF tag, it will not affect transmission. And since the communication range is wide, there is no need for extreme positioning which can greatly reduce the time and cost of design.

5. **Simultaneously access information of multiple RF tags**
    a. Some RFID systems are equipped with a function that allows you to simultaneously read the information of multiple RF tags existing within the transmissions area of the Reader/Writer.

▼ RFID working principle

- RFID belongs to a group of technologies referred to as Automatic Identification and Data Capture (AIDC).
- AIDC methods automatically identify objects, collect data about them, and enter those data directly into computer systems with little or no human intervention.
- RFID methods utilize radio waves to accomplish this.
- At a simple level, RFID systems consist of three components: An RFID tag or smart label, an RFID reader, and an antenna.
- RFID tags contain an integrated circuit and an antenna, which is used to transmit data to the RFID reader (also called an interrogator).
- The reader then converts the radio waves to a more usable form of data.
- Information collected from the tags is then transferred through a communications interface to a host computer system, where the data can be stored in a database and analyzed later.

▼ RFID applications

RFID technology is employed in many industries to perform such tasks as:

- Inventory management
- Asset tracking
- Personnel tracking
- Controlling access to restricted areas
- ID Badging
- Supply chain management
- Counterfeit prevention (e.g. in the pharmaceutical industry)

▼ How are RFID and the Internet of Things linked

- RFID (Radio Frequency Identification) technology is closely linked to the Internet of Things (IoT) because it is one of the core components that enable the seamless communication and data exchange between objects and devices.

- RFID tags, which can be attached to various objects, use radio waves to send and receive information to and from RFID readers.
- With the help of RFID technology, objects can be identified, tracked and monitored in real-time, generating valuable data that can be analyzed to optimize business operations and enhance overall efficiency.
- The IoT leverages this technology by connecting a wide range of objects and devices, including those equipped with RFID tags, to the internet, enabling them to communicate and share data with each other.
- By integrating RFID with the IoT, businesses and organizations can gain real-time insights into their operations, allowing for proactive decision-making, improved customer experiences, and increased productivity.

▼ Define Near Field Communication technologies and their applications.

- NFC has its origins in radio frequency identification (RFID) technology, which uses electromagnetic fields to encode and read information. Any NFC-enabled device has a small chip that is activated when it comes in close proximity to another NFC chip (10 centimeters or less). NFC therefore enables simple and safe two-way interactions between electronic devices.
- There are two types of NFC devices: active and passive.
    - Active NFC devices, such as smart phones, are capable of both sending and receiving information.
    - Passive NFC devices can transmit information when read by active devices but cannot read information themselves.
- Application of NFC technology as follows:
    - Performing contactless transactions
    - Connecting electronic devices with a single tap
    - Sharing business cards
    - Accessing information from a smart poster
    - Downloading digital content
    - Providing credentials for security systems
- The benefits of NFC include easy connections, rapid transactions, and simple exchange of data. NFC serves as a complement to other popular wireless technologies such as Bluetooth, which has a wider range than NFC, but which also consumes more power.

▼ Bluetooth

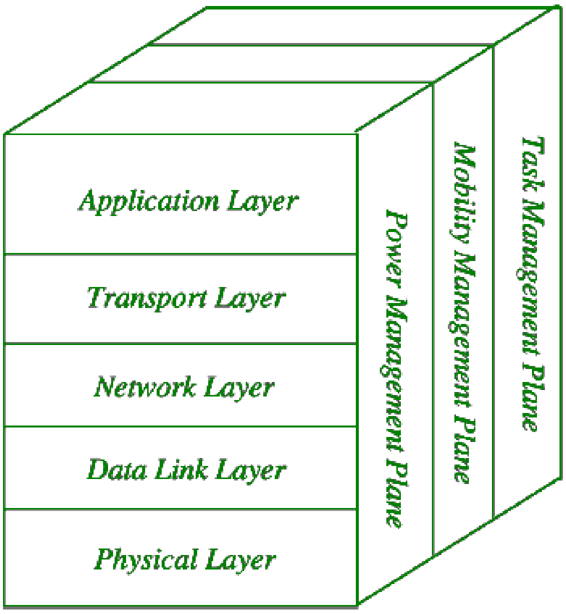
- Bluetooth is also important for the rapidly growing Internet of Things, including smart homes and industrial applications.
- It is a low power, low range, high bandwidth connectivity option.

- When Bluetooth devices connect to each, it follows the parent-child model, meaning that one device is the parent and other devices are the children. The parent transmits information to the child and the child listens for information from the parent.
- Invented by Ericsson in 1994, Bluetooth was intended to enable wireless headsets.
- Bluetooth has since expanded into a broad variety of applications including Bluetooth headsets, speakers, printers, video game controllers, and much more.
- A Bluetooth parent can have up to 7 children, which is why your computer can be connected via Bluetooth to multiple devices at the same time.
- **When devices are connected via Bluetooth, it's called a "piconet"**.
- Not only can a device be a parent in one piconet and a child in a different piconet at the same time, but the parent-child relationship can also switch.
- A drawback of Bluetooth is lower bandwidth, but for many industrial applications this higher bandwidth simply isn't needed.
- Bluetooth is also useful in a smart home setting.
- Again, many devices in the smart home don't need high bandwidth connections and it's much easier to set up Bluetooth.

▼ **Wireless Sensor Networks and its Applications**

- A Wireless Sensor Network is a kind of wireless network which includes a large number of circulating, self-directed, minute, low powered devices named sensor nodes.
- These networks certainly cover a huge number of spatially distributed, little, battery-operated, embedded devices that are networked to caringly collect, process, and transfer data to the operators, and it has controlled the capabilities of computing & processing.
- The most common WSN architecture follows the OSI architecture Model.
- **The architecture of the WSN includes five layers and three cross layers**
    - **Application Layer** The application layer is liable for traffic management and offers software for numerous applications that convert the data in a clear form to find positive information. Sensor networks arranged in numerous applications in different fields such as agricultural, military, environment, medical, etc.
    - **Transport Layer** The function of the transport layer is to deliver congestion avoidance and reliability where a lot of protocols intended to offer this function are either practical on the upstream. These protocols use dissimilar mechanisms for loss recognition and loss recovery. The transport layer is exactly needed when a system is planned to contact other networks.
    - **Network Layer** The main function of the network layer is routing, it has a lot of tasks based on the application, but actually, the main tasks are in the power conserving, partial memory, buffers, and sensor don't have a universal ID and have to be self-organized.

- **Data Link Layer** The data link layer is liable for multiplexing data frame detection, data streams, MAC, & error control, confirm the reliability of point to point or multipoint.
- **Physical Layer** The physical layer provides an edge for transferring a stream of bits above physical medium. This layer is responsible for the selection of frequency, generation of a carrier frequency, signal detection, Modulation & data encryption.



- Characteristics of Wireless Sensor Network
  The characteristics of WSN include the following:
  - The consumption of Power limits for nodes with batteries
  - Capacity to handle with node failures
  - Some mobility of nodes and heterogeneity of nodes
  - Scalability to large scale of distribution
  - Capability to ensure strict environmental conditions
  - Simple to use
  - Cross-layer design
- Advantages of Wireless Sensor Networks
  The advantages of WSN include the following:
  - Network arrangements can be carried out without immovable infrastructure.
  - Apt for the non-reachable places like mountains, over the sea, rural areas and deep forests.
  - Flexible if there is a casual situation when an additional workstation is required.
  - Execution pricing is inexpensive.
  - It avoids plenty of wiring.

    - It might provide accommodations for the new devices at any time.
    - It can be opened by using a centralized monitoring.
- Wireless Sensor Network Applications
    - Wireless sensor networks may comprise of numerous different types of sensors like low sampling rate, seismic, magnetic, thermal, visual, infrared, radar, and acoustic, which are clever to monitor a wide range of ambient situations. Sensor nodes are used for constant sensing, event ID, event detection & local control of actuators.
    - The applications of wireless sensor network mainly include:
        - Military Applications
        - Health Applications
        - Environmental Applications
        - Home Applications
        - Commercial Applications
        - Area monitoring
        - Health care monitoring
        - Environmental/Earth sensing
        - Air pollution monitoring
        - Forest fire detection
        - Landslide detection
        - Water quality monitoring
        - Industrial monitoring

▼ **Explain Software Defined Networking. Is SDN a mature technology? Justify it.**

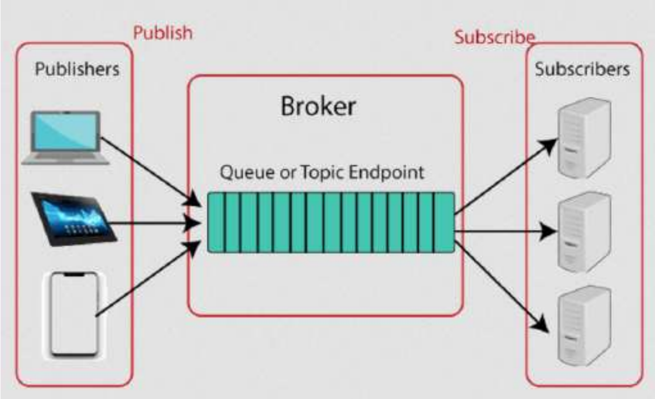
- Software Defined Networking (SDN) is a network architecture that separates the control plane and data plane of a network.
- It enables network administrators to dynamically and centrally manage network resources from a single control point.
- In SDN, the control plane is managed by a software controller that communicates with the data plane through a standardized interface, known as OpenFlow. This separation of control and data planes enables network administrators to update or modify the network's behavior and policies easily.
- SDN allows organizations to simplify network management, reduce operational costs, and improve network efficiency. Also, it enables network administrators to customize their networks by adding new applications and services in real-time. Additionally, SDN enhances network security by enabling administrators to centrally manage firewall and access control policies.

- SDN adoption has increased significantly in recent years. Many large enterprises and service providers have deployed SDN in their networks. Moreover, the adoption of SDN is also growing rapidly in the cloud computing domain.
- Though SDN is a relatively new technology, it is mature enough to be deployed on large-scale networks. It has already proven its worth in many enterprises and service providers, and its adoption is increasing day by day. However, SDN still has room for improvement, such as better security mechanisms and handling of complex network topologies. But overall, SDN is a mature technology that is redefining network management and security.

Unit IV

▼ MQTT

- MQTT stands for Message Queuing Telemetry Transport.
- lightweight messaging protocol introduced by IBM in 1999
- MQTT is a machine to machine internet of things connectivity protocol.
- It is an **extremely lightweight and publish-subscribe messaging transport protocol**.
- This protocol is useful for the connection with the remote location where the bandwidth is a premium.
- The connected devices in the MQTT protocol are known as “clients,” which communicate with a server referred to as the “broker.”
- The broker handles the task of data transmission between clients.
- Whenever a client (known as the “publisher”) wants to distribute information, it will publish to a particular topic, the broker then sends this information to any clients that have subscribed to that topic (known as “subscribers”).
- The publisher does not need any data on the number or the locations of subscribers. In turn, subscribers do not need any data about the publisher.
- Any client can be a publisher, subscriber, or both. The clients are typically not aware of each other, only of the broker that serves as the intermediary.
- This setup is popularly known as the “pub/sub model”.



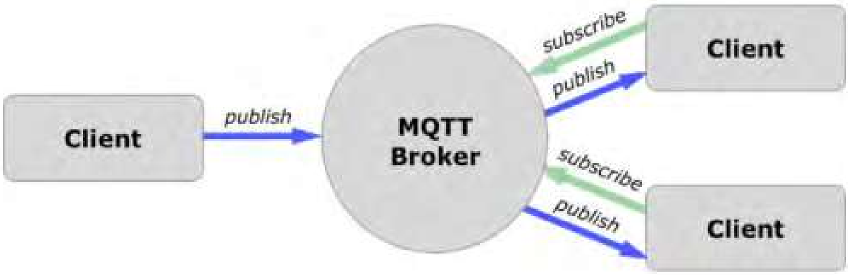
MQTT Architecture

▼ MQTT methods

- MQTT methods are also, referred to as verbs.
- MQTT defines methods to indicate desired actions to be performed on identified resources.
- Resources can be files or the outputs of an executable, found on a server.
- **Methods in MQTT are:**
  - **Connect –** Waits for connection to be established with the server.
  - **Disconnect –** Waits for the MQTT client to finish any work, which needs to be done and for the TCP/IP session to disconnect.
  - **Subscribe –** Requests the server to let the client subscribe to one or more topics.
  - **Unsubscribe –** Requests the server to let the client unsubscribe from one or more topics.
  - **Publish –** Returns immediately to application thread after passing request to the MQTT client.

▼ MQTT Components

- **MQTT main components are as follows:**
  - **A broker -** broker which is the server that handles the data transmission between the clients.
  - **A topic -** which is the place a device wants to put or retrieve a message to/from.
  - **The message** - which is the data that a device receives “when subscribing” from a topic or send “when publishing” to a topic.
  - **Publish -** publish is the process a device does to send its message to the broker.
  - **Subscribe -** where a device does to retrieve a message from the broker.
- Like any other internet protocol, MQTT is based on clients and a server. Likewise, the server is the node who is responsible for handling the client’s requests of receiving or sending data between each other.



MQTT working

- MQTT server is called a broker and the clients are simply the connected devices.

    - When a device (a client) wants to send data to the broker, we call this operation a “publish”.
    - When a device (a client) wants to receive data from the broker, we call this operation a “subscribe”.

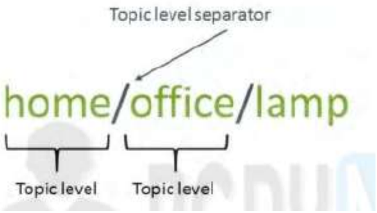
▼ **MQTT communication**

- MQTT (Message Queuing Telemetry Transport) Communication is a process for exchanging messages among devices. It is frequently used in IoT applications.
- MQTT is intended for large networks with low data traffic and designed to minimize data volumes.
- Data transfer using MQTT takes place over TCP.
- It may be encrypted with SSL.
- A “publisher-subscriber” data transfer model is used.
- This means that messages are exchanged using one central hub (a MQTT broker).
- **MQTT publisher**
    - A MQTT publisher sends MQTT messages to the MQTT broker.
    - A MQTT client can publish messages as long as it is connected to a MQTT broker.
    - The MQTT protocol categorizes the messages by the topic.
    - Every message must contain a topic that can be used by the MQTT broker to pass the message on to the subscribed MQTT subscribers.
    - Every message has a payload that is delivered to the subscribers in this way. It can carry any content.
- **MQTT broker**
    - A MQTT broker is a central hub (typically in a cloud in the public internet) that connects MQTT publishers with MQTT subscribers.
    - MQTT publishers send messages and MQTT subscribers subscribe to receive the messages.
    - There can be several MQTT subscribers to the same “topic”.
    - Messages are divided into “topics”; a device may either “publish” a given topic, or “subscribe” to the topic.
    - Within a topic, messages are exchanged as they are received by the MQTT broker and then sent to the subscribed devices.
    - A device (electrical socket) can be simultaneously a publisher for some topics (publishes the measured values) and a subscriber for other topics (reacts to commands for controlling the output).
- **MQTT subscriber**
    - A MQTT subscriber receives MQTT messages from the MQTT broker.

  - Messages are categorized into topics that can be subscribed too.
  - It uses the pub/sub pattern to connect interested parties with each other.
  - It does it by decoupling the sender (publisher) with the receiver (subscriber).
  - The publisher sends a message to a central topic which has multiple subscribers waiting to receive the message.

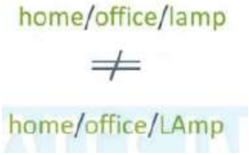
▼ **MQTT Topics and applications**

- Another important concept is the topics. Topics are the way to register interest for incoming messages or how you specify where you want to publish the message.
- Topics are represented with strings separated by a forward slash. Each forward slash indicates a topic level.
- Here's an example on how you would create a topic for a lamp in your home office:

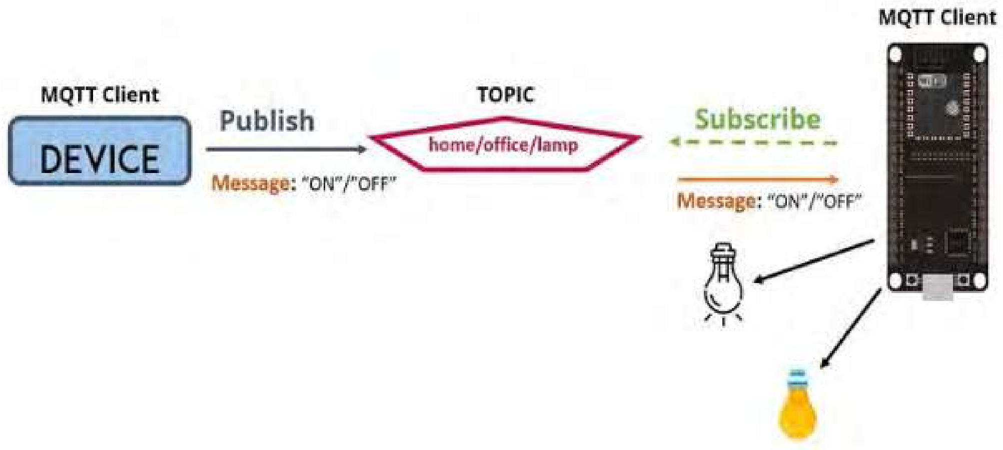


Example of Topic level separator

- Note: topics are case-sensitive, which makes these two topics different:



- **Subscribe to a specific topic**
  - home/hall/temperature
- **Single level wildcard**
  - home/ + / temperature
  - Used to subscribe a single hierarchical level
- **Multi level wildcard**
  - home/ #
  - Used to subscribe multiple hierarchical level
- Below figure shows the scenario to turn on a lamp in the home or office using MQTT



    - Should have a device that publishes “on” and “off” messages on the home/office/lamp topic.
    - Should have a device that controls a lamp (it can be an ESP32, ESP8266, or any other board). The ESP32 that controls your lamp is subscribed to that topic: home/office/lamp.
    - So, when a new message is published on that topic, the ESP32 receives the “on” or “off” message and turns the lamp on or off.
- MQTT was thus created with these features to greatly fit-in for constrained IoT networks.
- There was one more great advantage of MQTT, MQTT was able to Separate data creators & data consumers.
- Data creators are sensor nodes and data consumers could be cloud based applications or could even be other connected devices.
- By incorporating a publish-subscribe model which has a server also called broker in between, MQTT enables massive scalability of devices, which is much essential for IoT.
- **Advantages of MQTT—**
    - Simple to Implement,
    - Quality of Service for Data Delivery
    - Light Weight
    - Bandwidth Efficient
    - Data Agnostic – So that we could send different types of data such as values, images, etc.
    - Continuous Session Awareness- To Know the current status in Real-time

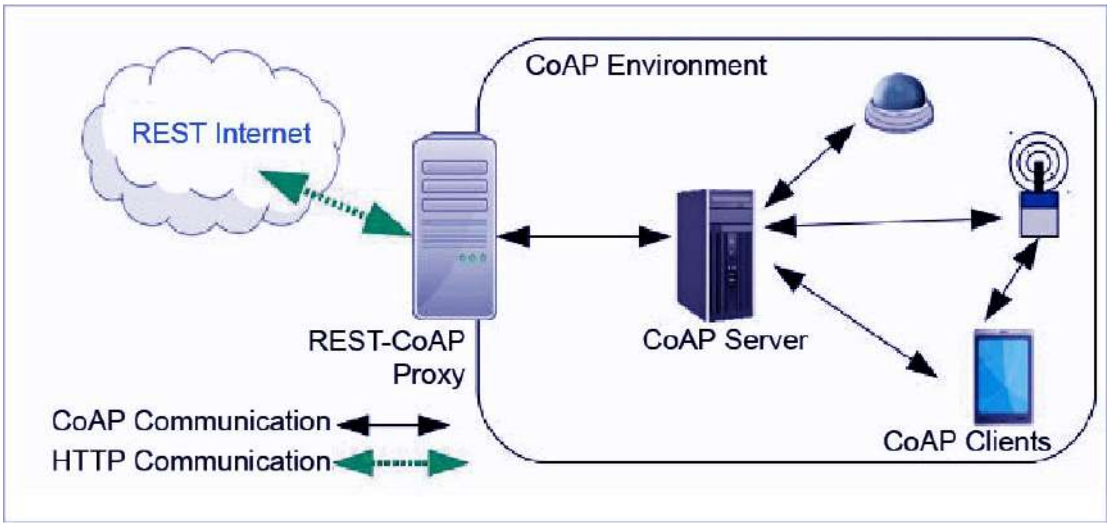
▼ SMQTT

- SMQTT (Secure Message Queue Telemetry Transport) is an extension of MQTT protocol which uses encryption based on lightweight attribute encryption.

- The main advantage of this encryption is that it has a broadcast encryption feature. In this features, one message is encrypted and delivered to multiple other nodes.
- **The process of message transfer and receiving consists of four major stages:**
  1. **Setup:** In this phase, the publishers and subscribers register themselves to the broker and get a secret master key.
  2. **Encryption:** When the data is published to broker, it is encrypted by broker.
  3. **Publish:** The broker publishes the encrypted message to the subscribers.
  4. **Decryption:** Finally, the received message is decrypted by subscribers with the same master key. SMQTT is propose used to enhance MQTT security feature.

▼ **Explain CoAP. How it can be used between devices on the same constrained network? Justify it.**

- CoAP stands for Constrained Application Protocol.
- CoAP is a session layer protocol that provides the RESTful (HTTP) interface between HTTP client and server.
- It is **designed by IETF Constrained RESTful Environment (CoRE) working group**.
- It is designed to use devices on the same constrained network between devices and general nodes on the Internet.
  - Constrained networks are those with limited bandwidth or other constraints that make traditional networking protocols impractical or impossible to use.
- CoAP enables low-power sensors to use RESTful services while meeting their low power constraints.
- This protocol is specially built for IoT systems primarily based on HTTP protocols.
- This network is used within the limited network or in a constrained environment.
- The whole architecture of CoAP consists of CoAP client, CoAP server, REST CoAP proxy, and REST internet.



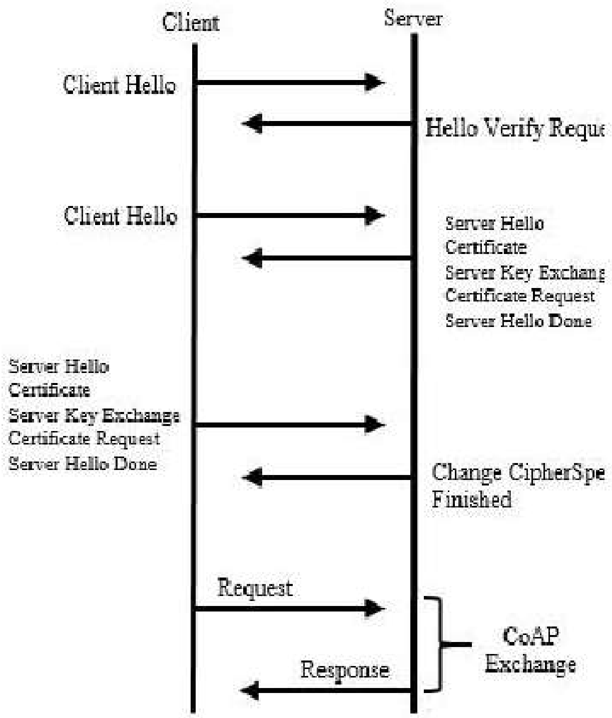
- The data is sent from CoAP clients (such as smart phones, RFID sensors, etc.) to the CoAP server and the same message is routed to REST CoAP proxy. The REST CoAP proxy interacts outside the CoAP environment and uploads the data over REST internet.

▼ CoAP message types

- CoAP makes use of **two message types, requests and responses**, using a simple, binary, base header format.
- The base header may be followed by options in an optimized Type-Length-Value format.
- CoAP is by default bound to UDP and optionally to DTLS, providing a high level of communications security.
- Any bytes after the headers in the packet are considered the message body.
- The length of the message body is implied by the datagram length.
- The entire message must fit within a single datagram when bound to UDP.
- When used with 6LoWPAN, as defined in RFC 4944, messages SHOULD also fit into a single IEEE 802.15.4 frame to minimize fragmentation.
- One must take security into account when dealing with IoT protocols. For example, CoAP uses UDP to transport information. CoAP relies on UDP security features to protect information. As HTTP uses TLS over TCP, CoAP uses Datagram TLS over UDP. DTLS supports RSA, AES, and so on.

▼ CoAP Request-Response model

- The CoAP Request/Response is the second layer in the CoAP abstraction layer.
- The request is sent using a Confirmable (CON) or Non-Confirmable (NON) message.
- There are several scenarios depending on if the server can answer immediately to the client request or the answer if not available.
- If the server can answer immediately to the client request, then if the request is carried using a Confirmable message (CON), the server sends back to the client an Acknowledge message containing the response or the error code



CoAP Request/Response with DTLS—4 round trips

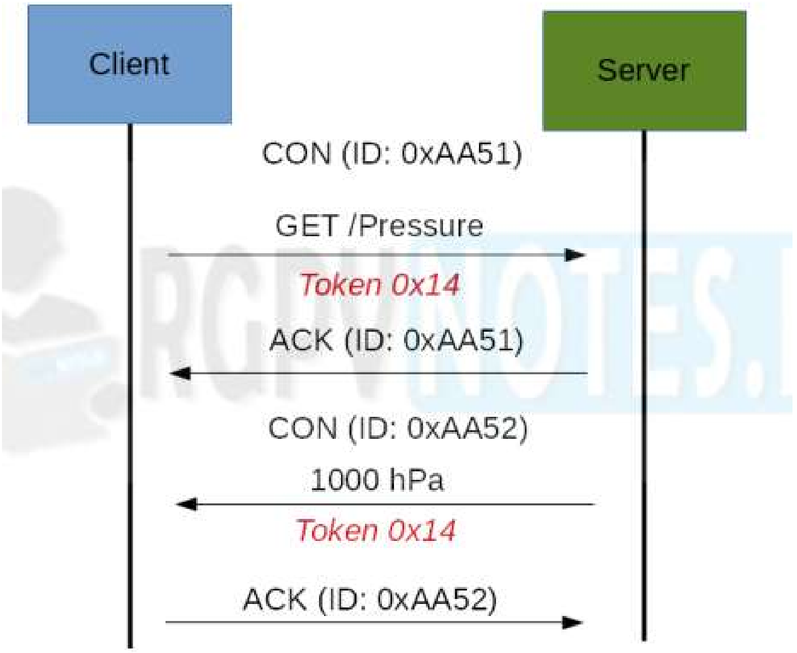


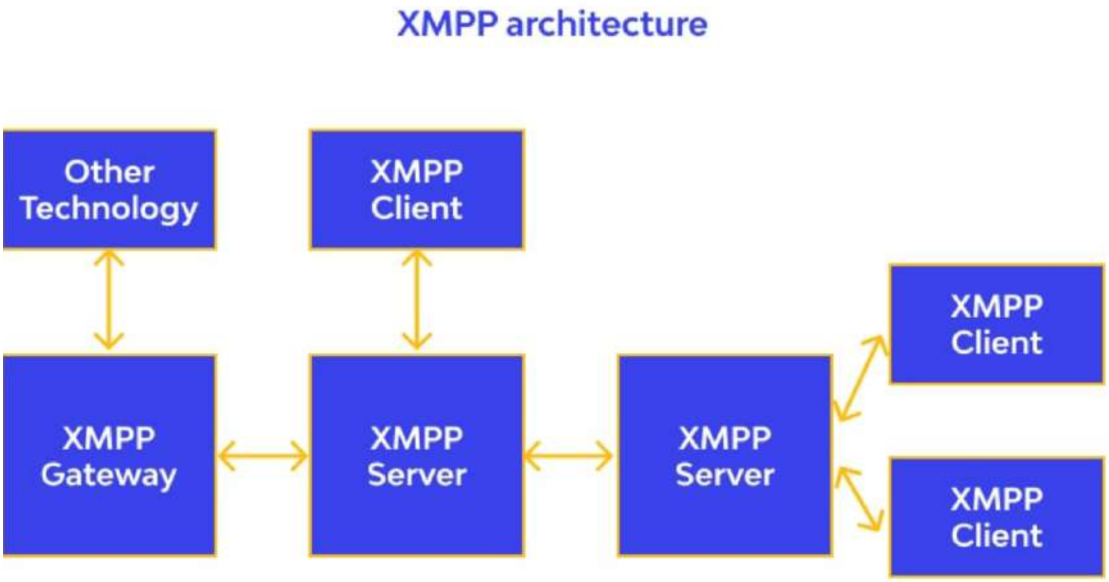
Figure 4.6: CoAP Request/Response Model

- As it can be noticed in the CoAP message, there is a Token.
- The Token is different from the Message-ID and it is used to match the request and the response.

- If the server can't answer to the request coming from the client immediately, then it sends an Acknowledge message with an empty response.
- As soon as the response is available, then the server sends a new confirmable message to the client containing the response.
- At this point, the client sends back an Acknowledge message
- If the request coming from the client is carried using a NON-confirmable message, then the server answer using a NON-confirmable message.

▼ **XMPP**

- XMPP is a short form for **Extensible Messaging Presence Protocol.**
- It is a protocol for streaming XML elements over a network in order to exchange messages and presence information in close to real time.
- This protocol is mostly used by instant messaging applications like WhatsApp.
- **Meaning of each character of word XMPP:**
    - **X:** It means extensible. XMPP is an open-source project which can be changed or extended according to the need.
    - **M:** XMPP is designed for sending messages in real time. It has very efficient push mechanism compared to other protocols.
    - **P:** It determines whether you are online/offline/busy. It indicates the state.
    - **P:** XMPP is a protocol, that is, a set of standards that allow systems to communicate with each other.



XMPP Architecture

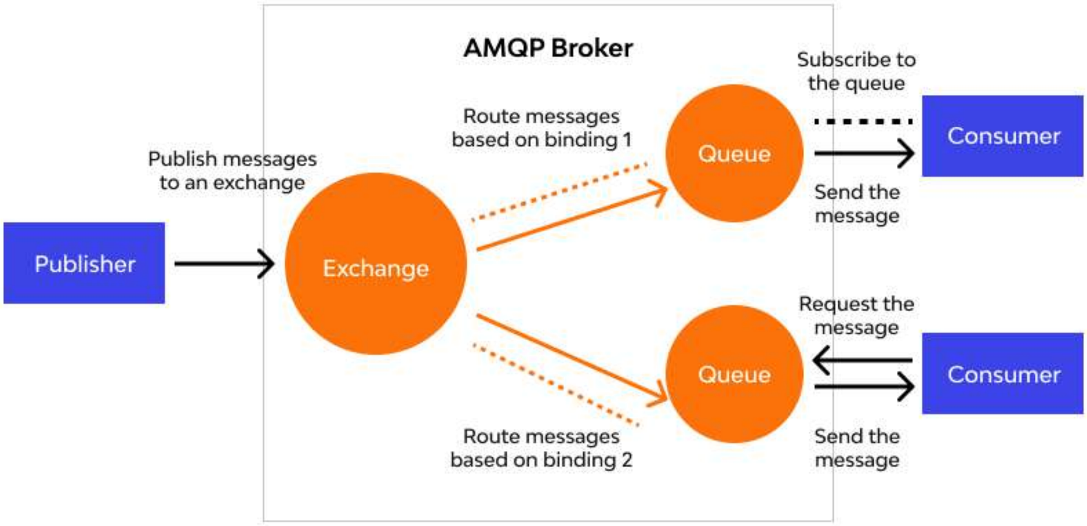
- These are the basic requirements of any Instant Messenger which are fulfilled by XMPP:
    - Send and receive messages with other users.

    - Check and share presence status
    - Manage subscriptions to and from other users.
    - Manage contact list
    - Block communications (receive message, sharing presence status, etc) to specific users.
- **Some other XMPP features:**
    - **Decentralized**
      XMPP is based on client-server architecture, i.e., clients don't communicate directly, and they do it with the help of server as intermediary. It is decentralized means there is no centralized XMPP server just like email, anyone can run their own XMPP server. Resource is used in case the application support mobile as well as desktop or web application, so it can be optional in case a Instant Messenger Application support only single kind of resource.
- **XMPP implementation**
    - The original protocol for XMPP is **Transmission Control Protocol**, using open ended XML streams over long- lived TCP connections.
    - In some cases, there are restricted firewalls, XMPP (port 5222) is blocked, so it can't be used for web applications and users behind restricted firewalls, to overcome this, XMPP community also developed a HTTP transport. And as the client uses HTTP, most firewalls allow clients to fetch and post messages without any problem.
    - **Thus, in scenarios where the TCP port used by XMPP is blocked, a server can listen on the normal HTTP port and the traffic should pass without problems.**

▼ **AMQP features**

- **Advanced Message Queuing Protocol (AMQP)** is an open source published standard for asynchronous messaging by wire.
- AMQP enables encrypted and interoperable messaging between organizations and applications.
- The protocol is used in client/server messaging and in IoT device management.
- AMPQ is efficient, portable, multichannel and secure.
- The binary protocol offers authentication and encryption by way of **SASL *[Simple Authentication and Security Layer]* or TLS *[Transport Layer Security]***, relying on a transport protocol such as TCP.
- The messaging protocol is fast and features guaranteed delivery with acknowledgement of received messages.
- AMPQ works well in multi-client environments and provides a means for delegating tasks and making servers handle immediate requests faster.
- Because AMPQ is a streamed binary messaging system with tightly mandated messaging behavior, the interoperability of clients from different vendors is assured.

- AMQP allows for various guaranteed messaging modes specifying a message be sent:
    - **At-most-once** (sent one time with the possibility of being missed).
    - **At-least-once** (guaranteeing delivery with the possibility of duplicated messages).
    - **Exactly-once** (guaranteeing a one-time only delivery).
- **Publisher—>Exchange—>Queue—>Consumer**



- **Exchange type**

Exchange type	Default pre-declared names
Direct exchange	(Empty string) and amq.direct
Fanout exchange	amq.fanout
Topic exchange	amq.topic
Headers exchange	amq.match (and amq.headers in RabbitMQ)

▼ **AMQP components**

Components of AMQP are as follows :

- **Message queue**
    - A queue acts as a buffer that stores messages that are consumed later. A queue can also be declared with a number of attributes during creation.
    - For instance, it can be marked as durable, auto-delete and exclusive, where exclusive means that it can be used by only one connection and this queue will be deleted when that connection closes.
- **Exchanges and Exchange Types**

    - A channel routes messages to a queue depending on the exchange type and bindings between the exchange and the queue.
    - For a queue to receive messages, it must be bound to at least one exchange.
    - AMQP brokers should provide four exchange types - direct exchange, fanout exchange, topic exchange, and header exchange.
    - An exchange can be declared with a number of attributes during creation.
    - For instance, it can be marked as durable so that it survives a broker restart, or it can be marked as auto-delete meaning that it's automatically deleted when the last queue is unbound.
- **Binding**
    - A binding is a relation between a queue and an exchange consisting of a set of rules that the exchange uses (among other things) to route messages to queues.
- **Message and Content**
    - A message is an entity sent from the publisher to the queue and finally subscribed to by the consumer. Each message contains a set of headers defining properties such as life duration, durability, and priority.
    - AMQP also has a built-in feature called message acknowledgment that is used to confirm message delivery and/or processing.
- **Connection**
    - A connection in AMQP is a network connection between your application and the AMQP broker, e.g. a TCP/IP socket connection.
- **Channel**
    - A channel is a virtual connection inside a connection, between two AMQP peers. Message publishing or consuming to or from a queue is performed over a channel (AMQP). A channel is multiplexed; one single connection can have multiple channels.

▼ **AMQP frame types**

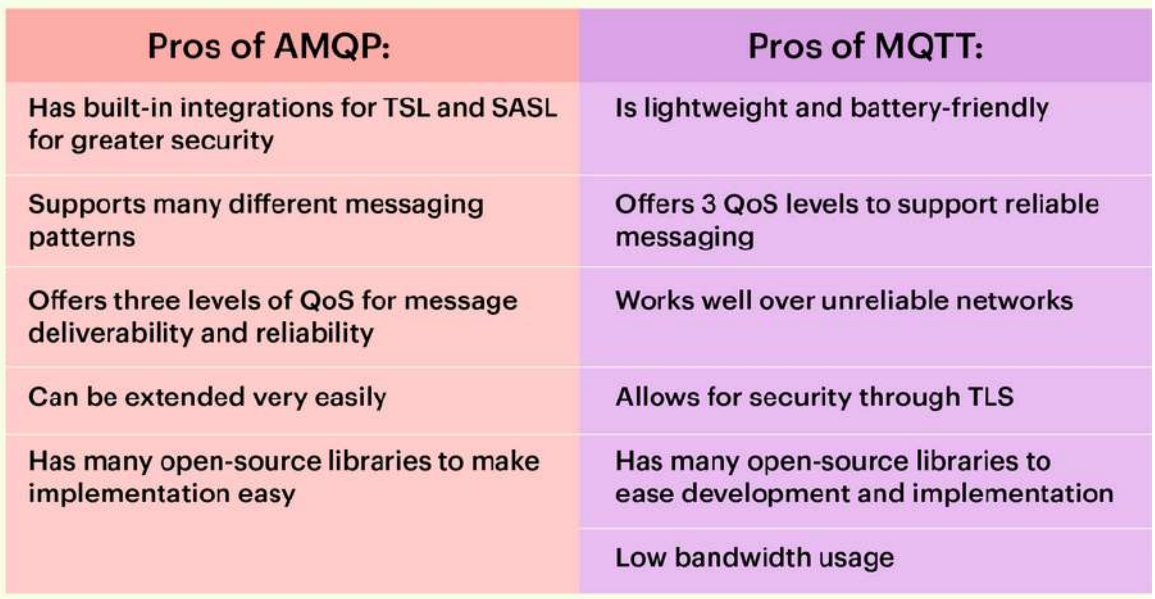
- A frame is the basic unit with AMQP.
- A connection consists of the ordered sequence of frames.
- Order in this case means that the last frame must not arrive at the receiver until all other frames have first reached their destination.
- **Each frame can be divided into three segments (in version 1.0):**
    - **Frame header:** This mandatory header has a size of 8 bytes. Here you will find information that determines the routing of the message.
    - **Extended header:** This area is optional and has no set scope. It serves to expand the header in the future with further information.
    - **Frame body:** The body contains the actual data to be transferred. The size is freely selectable. However, this area can also be left empty, and then the frame only serves to

maintain the connection.

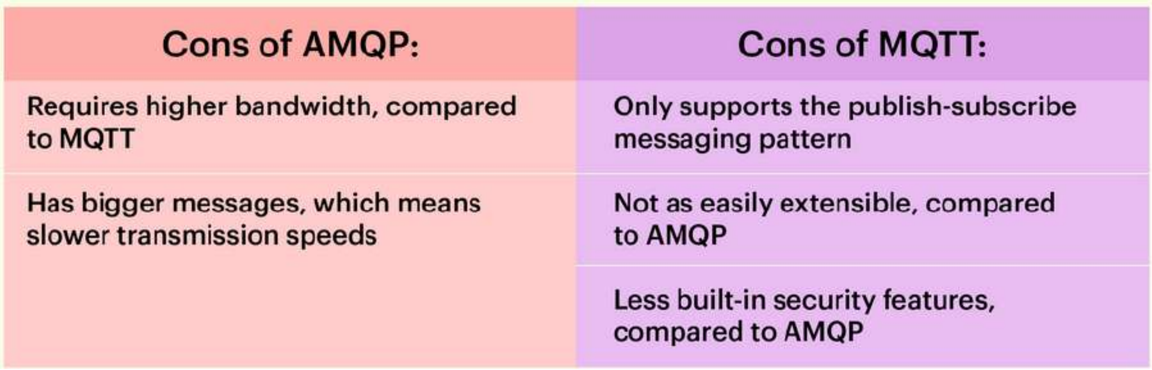
- **The body of a frame, in turn, can take nine different forms:**
  - **open—**Negotiates the connection parameters between broker and client
  - **begin**—Indicates that a connection is starting
  - **attach**—The message is appended with a link that is necessary in order to use the data transfer
  - **flow**—Changes the status of a link
  - **transfer**—The actual message is transmitted with the transfer frame
  - **disposition**—A disposition frame provides information on changes to the information delivery
  - **detach**—Removes the link
  - **end**—Indicates that the connection will be terminated
  - **close—**Terminates the connection and declares that no further frames will be sent
- **Queues & messages**

▼ Differentiate between AMQP and MQTT. Does MQTT use Web Sockets?

- AMQP (Advanced Message Queuing Protocol) and MQTT (Message Queuing Telemetry Transport) are two popular application layer protocols used for messaging and communication between devices. The main difference between AMQP and MQTT is that MQTT is a lightweight protocol that uses very little network bandwidth, while AMQP is a more feature-rich protocol that supports more advanced messaging scenarios.



Pros of AMQP:	Pros of MQTT:
Has built-in integrations for TSL and SASL for greater security	Is lightweight and battery-friendly
Supports many different messaging patterns	Offers 3 QoS levels to support reliable messaging
Offers three levels of QoS for message deliverability and reliability	Works well over unreliable networks
Can be extended very easily	Allows for security through TLS
Has many open-source libraries to make implementation easy	Has many open-source libraries to ease development and implementation
	Low bandwidth usage



Cons of AMQP:	Cons of MQTT:
Requires higher bandwidth, compared to MQTT	Only supports the publish-subscribe messaging pattern
Has bigger messages, which means slower transmission speeds	Not as easily extensible, compared to AMQP
	Less built-in security features, compared to AMQP

- MQTT does support Web Sockets, which allows for bidirectional, low-latency communication between clients and servers over a single, persistent connection. Web Sockets use the same ports as HTTP and HTTPS, and can be used to enable secure communication between web servers and mobile devices or other internet-connected devices.

Unit V

IoT Platforms

IoT platforms are software frameworks that are designed to enable the creation, deployment, monitoring, and management of Internet of Things (IoT) applications and devices. These platforms typically provide a range of tools and services that can help developers to build and deploy IoT applications quickly and easily. Some of the key features of IoT platforms include:

1. Device management: IoT platforms provide tools for managing the lifecycle of IoT devices, including provisioning, configuration, monitoring, and firmware updates.
2. Data management: IoT platforms provide tools for managing the data generated by IoT devices, including data ingestion, storage, processing, and analytics.
3. Application development: IoT platforms provide tools for developing IoT applications, including SDKs, APIs, and development tools.
4. Security: IoT platforms provide tools for ensuring the security of IoT devices and applications, including secure boot, data encryption, and access control.
5. Integration: IoT platforms provide tools for integrating IoT applications with existing enterprise systems and third-party applications, including APIs and connectors.

Some of the top IoT platforms available in the market include:

1. AWS IoT
2. IBM Watson IoT Platform
3. Microsoft Azure IoT
4. Google Cloud IoT Core
5. Cisco IoT Cloud Connect
6. Oracle IoT

7. Bosch IoT Suite.

▼ Arduino *[working?]*

Arduino is an open-source prototyping platform that uses a microcontroller board and software to control electronic devices. It is a physical computing device that can interact with the physical world using inputs like sensors and switches and outputs like motors, LEDs, and displays. The working of Arduino involves the following steps:

1. **Microcontroller Board:** Arduino uses a microcontroller board, like the ATmega328P, which is the primary controller of the system. The board is connected to a computer via USB cable and powered through a DC power source.
2. **Programming:** Once the board is connected to a computer, it is programmed using the Arduino Integrated Development Environment (IDE), which is free software used to write and upload code to the board.
3. **Input:** The microcontroller reads input from various sensors, buttons, and switches connected to pins on the board. These inputs can be analog, digital, or both.
4. **Processing:** Once the input is received, the microcontroller processes this data according to the code written in the IDE.
5. **Output:** The processed data is sent to output devices like LEDs, motors, and LCDs, which are wired to pins on the board. These outputs can be analog, digital, or both.
6. **Control:** The Arduino board acts as the control center that coordinates the inputs, processing, and outputs. It is programmed to perform specific functions based on the inputs it receives.

In summary, Arduino is a flexible platform for designing and building electronic systems. It is an open-source project, which means that the source code and design files are available to anyone who wants to modify or improve upon it. Arduino makes it possible for anyone to build interactive projects and to learn and experiment with electronics and programming.

▼ Raspberry Pi Board *[working?]*

- Raspberry Pi is a single-board computer that is designed to be a low-cost, compact, and portable computing device. The board is equipped with a microprocessor, memory, input/output pins, and connectivity options such as Wi-Fi and Ethernet.
- When powered on, Raspberry Pi boots the operating system from a microSD card inserted into the card slot. The user can choose to install one of several operating systems, including Raspbian, Ubuntu, and Windows 10 IoT Core.
- The board is powered by a micro USB cable that can be connected to a wall adapter or a power bank, making it portable and easy to carry. The Raspberry Pi supports a wide range of programming languages, including Python, C++, and Scratch, making it a popular choice for programming enthusiasts, makers, and educators.

▼ How Raspberry pi (3) is different from desktop computer. Write use of SPI, I2C interfaces and GPIO pins of Raspberry pi (3).

There are several key differences between a Raspberry Pi 3 and a desktop computer:

1. **Processing power**
   The Raspberry Pi 3 has significantly less processing power than a typical desktop computer. It uses a quad-core ARM processor with a clock speed of 1.2GHz, while a modern desktop computer can have a processor with multiple cores and speeds of 3GHz or higher.
2. **RAM**
   The Raspberry Pi 3 typically has 1GB of RAM, while a desktop computer can have anywhere from 4GB to 32GB of RAM.
3. **Graphics**
   The Raspberry Pi 3 has an integrated graphics processor, which is sufficient for basic tasks but not ideal for high-end gaming or other graphics-intensive applications.
4. **Storage**
   The Raspberry Pi 3 uses microSD cards for storage, which are typically smaller than the hard drives used in desktop computers.
5. **Connectivity**
   The Raspberry Pi 3 has built-in Wi-Fi and Bluetooth, while desktop computers typically require the use of separate peripherals.
6. **Power consumption**
   The Raspberry Pi 3 has a much lower power consumption than a desktop computer, making it ideal for applications where power efficiency is important.
7. **Size**
   The Raspberry Pi 3 is tiny compared to a desktop computer, making it ideal for use in projects where space is limited.

The Raspberry Pi has several interfaces and GPIO pins that can be used for a variety of purposes. Here are some use cases for each of the interfaces:

1. **SPI interface**
   SPI (Serial Peripheral Interface) is a synchronous serial communication interface that allows communication between devices using a master/slave architecture. The Raspberry Pi has two SPI interfaces that can be used for connecting to various sensors and other devices. Some common use cases of the SPI interface include:
   - Connecting to a SPI-based touch screen display for user input and output.
   - Controlling an LED strip using a SPI-based LED driver.
   - Reading data from a SPI-based accelerometer or gyroscope for orientation sensing.
2. **I2C interface**
   I2C (Inter-Integrated Circuit) is a serial communication bus that allows devices to communicate with each other over a shared set of wires. The Raspberry Pi has two I2C interfaces that can be used for connecting to various sensors and other devices. Some common use cases of the I2C interface include:
   - Reading temperature and humidity data from a I2C-based sensor such as the HTU21D.

- Reading barometric pressure and altitude data from a I2C-based sensor such as the BMP280.
- Controlling a servo motor using a I2C-based driver.

3. **GPIO pins**
   GPIO (General Purpose Input/Output) pins are pins on a microcontroller or computer that can be programmed to act as either an input or an output.

   The Raspberry Pi has 26 GPIO pins that can be used for connecting to various sensors and other devices. Some common use cases of the GPIO pins include:

   - Controlling an LED or other output device using a GPIO pin as an output.
   - Reading input from a switch or other sensor using a GPIO pin as an input.
   - Interfacing with a serial device using the UART pins as GPIOs.

▼ Google cloud IoT platform

Google launched its platform for Internet of Things development based on its end-to-end Google Cloud Platform.

Currently, it’s one of the world’s top Internet of Things platforms.

Google cloud IoT is the integration of various services that add value to connected solutions.

- Cloud IoT Core allows you to capture and handle device data. A device manager component is used to register devices with the service and monitor and configure them. MQTT and HTTP protocol bridges are used for device connection and communication with the Google cloud platform.
- Cloud pub/Sub performs data ingestion and message routing for further data processing.
- Google BigQuery enables secure real-time data analytics.
- AI platform applies machine learning features.
- Google data studio visualizes data by making reports and dashboards.
- Google maps platform helps visualize the location of connected assets.

The platform automatically integrates with internet of Things hardware producers such as Intel and microchip.

It supports various operating systems, including debian Linux OS.

Core features of Google cloud IoT:

- AI and machine learning capabilities
- Real-time data analysis
- Strong data visualization
- Location tracking

▼ Data Analytics for IoT

Data analytics for IoT involves the collection, processing, and analysis of large amounts of data generated by connected devices and sensors. This data is used to extract insights, identify patterns, and drive decision-making in order to optimize performance, improve efficiency, and enhance user experiences across various industries.

Data Analytics has a significant role to play in the growth and success of IoT applications and investments.

Analytics tools will allow the business units to make effective use of their datasets as explained in the points listed below:

- **Volume:** There are huge clusters of data sets that IoT applications make use of. The business organizations need to manage these large volumes of data and need to analyze the same for extracting relevant patterns. These datasets along with real-time data can be analyzed easily and efficiently with data analytics software.
- **Structure:** IoT applications involve data sets that may have a varied structure as unstructured, semi- structured and structured data sets. There may also be a significant difference in the data formats and types. Data analytics will allow the business executive to analyze all of these varying sets of data using automated tools and software.
- **Driving Revenue:** The use of data analytics in IoT investments will allow the business units to gain an insight into customer preferences and choices. This would lead to the development of services and offers as per the customer demands and expectations. This, in turn, will improve the revenues and profits earned by the organizations.
- **Competitive Edge:** IoT is a buzzword in the current era of technology and there are numerous IoT application developers and providers present in the market. The use of data analytics in IoT investments will provide a business unit to offer better services and will, therefore, provide the ability to gain a competitive edge in the market.

▼ Cloud for IoT

- IoT cloud refers to any number of cloud services that power the IoT.
- These include the underlying infrastructure needed for processing and storing IoT data, whether in real time or not.
- IoT cloud also includes the services and standards necessary for connecting, managing, and securing different IoT devices and applications.
- As with other types of cloud services, such as software-as-a-service, organizations consume IoT cloud services as they need them, rather than building a datacenter or other on-premises infrastructure to deliver those services locally.
- IoT cloud offers a more efficient, flexible, and scalable model for delivering the infrastructure and services needed to power IoT devices and applications.
- The IoT is virtually limitless in scale, unlike most organizations' resources. The cloud computing model effectively offers that kind of on-demand hyper scale, and it can do so in a cost-effective manner.
- IoT cloud enables organizations to leverage the significant potential of IoT without having to build the underlying infrastructure and services from scratch.

- IoT cloud also help promote and ensure standardization in key areas, including how devices communicate with each other, device management, and security.

Cloud storage models

Cloud storage models are models of cloud computing that stores data on the internet via cloud computing providers. These providers manage and operate data storage as a service.

Three major classes of physical storage models are in use today: direct attached storage (DAS), the storage area network (SAN), and network attached storage (NAS).

DAS (Direct attached storage)

- Direct attached storage is the simplest storage model.
- We are all familiar with DAS; this is the model used by most laptops, phones, and desktop computers.
- The fundamental unit in DAS is the computer itself; the storage for a server is not separable from the server itself.
- In the case of a phone it is physically impossible to remove the storage from the compute, but even in the case of servers, where it is theoretically possible to pull disk drives, once a drive is separated from the server, **it is generally wiped before reuse.**
- **SCSI and SATA are examples of DAS protocols.**

SAN (Storage Area Network)

- Eventually the storage industry recognized the utility of separating storage from the compute.
- Rather than attaching disks to each individual computer, we placed all the disks on a single cluster of servers and accessed the disk over the network.
- This simplifies storage management tasks such as backup and failure repair.
- This division of storage and compute is often called shared storage, since multiple computers will use a single pool of storage.
- **Fibre Channel and iSCSI are examples of SAN protocols**
- In a SAN an administrator will group a set of disks (or a portion of a set of disks) into a LUN (logical unit), which then behaves like a single disk drive to outside computers. The LUN is the fundamental unit used to manage SAN storage.

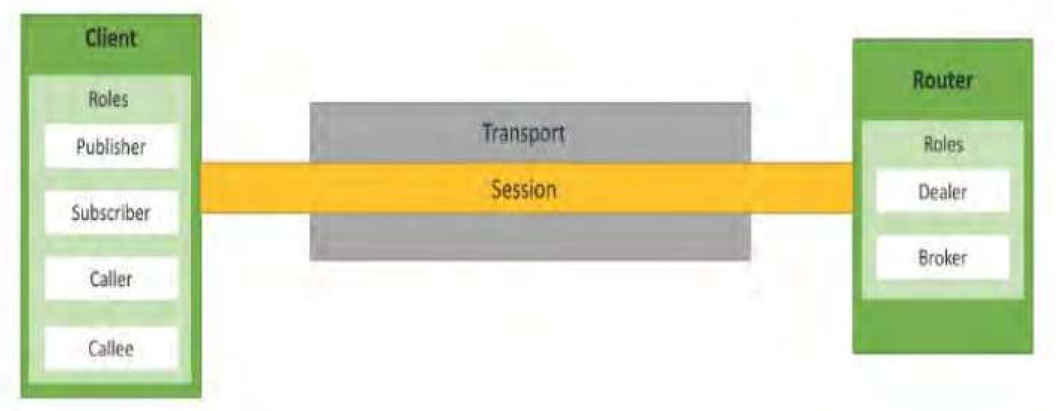
NAS (Network Attached Storage)

- While SANs allow us to move LUNs between one computer and another, the block protocols they use were not designed to concurrently share data in the same LUN between computers.
- To allow this kind of sharing we need a new kind of storage built for concurrent access.
- In this new kind of storage, we communicate with the storage using file system protocols, which closely resemble the file systems run on local computers. This kind of storage is known as network attached storage.

    - **NFS and SMB are examples of NAS protocols.**
    - The file system abstraction allows multiple servers to access the same data at the same time. Multiple servers can read the same file at the same time, and multiple servers can place new files into the file system at the same time.
    - Thus, NAS is a very convenient model for shared user or application data.
    - Popular Models for Cloud storage are:
        - Amazon Web Service (AWS)
        - XivelyCloud (PAAS)

▼ **Cloud communication APIs**

- Cloud Models are relied on Communication API.
- Communication API facilitates data transfer, control information transfer from application to cloud, one service to another.
- It also exists in the form of Communication Protocols. It supports RPC, PUBSUB and WAMP.
- E.g. Popular API is RESTful API (communication in cloud model).
- Django web framework is used to implement Communication API.
- **WAMP for IoT**
    - Web Application Messaging Protocol (WAMP) is a sub-protocol of Web Socket which provides publish– subscribe and remote procedure call (RPC) messaging patterns.
    - **WAMP Concepts**
        - **Transport:** Transport is a channel that connects two peers.
        - **Session:** Session is a conversation between two peers that runs over a transport.
        - **Client:** Clients are peers that can have one or more roles.
        - In the publish–subscribe model, the Client can have the following roles:–
            - Publisher: Publisher publishes events (including payload) to the topic maintained by the Broker.
            - Subscriber: Subscriber subscribes to the topics and receives the events including the payload.
        - In the publish–subscribe model, the Router has the role of a Broker. It acts as a Router and routes messages published to a topic to all the subscribers subscribed to the topic.



▼ **Twilio communication API**

- Twilio is a cloud communications API. This is sometimes referred to as communications platform as a service, or CPaaS.
- The term CPaaS emerged when companies like Twilio started offering application program interfaces (APIs)—a more developer-friendly and lower-cost option to integrate communications capabilities like voice, messaging, email, and video directly into software applications.
- Rather than building their own communications infrastructure from scratch, businesses that use cloud- based APIs from CPaaS vendors add real-time communications into their applications with a few lines of code.
- Unlike traditional communications infrastructure, communications built on a cloud communications platform are available without the burdens of capacity planning, carrier contracts, telecom hardware integration, and fragmented security.
- Cloud communications platforms like Twilio bring the world of communications to every web and mobile developer in the programming languages they already use.
- If a company is building a large, complex contact center, replacing a legacy corporate phone system, or building SMS notifications into a supply chain management app, the Twilio platform makes it simple.
- Without separate equipment, protocols, traditional infrastructure, telecom contracts, and software to deal with, the focus can be on building and iterating on the right solution for the task.

▼ **Smart farming**

- IoT smart agriculture products are designed to help monitor crop fields using sensors and by automating irrigation systems. As a result, farmers and associated brands can easily monitor the field conditions from anywhere without any hassle.
- **ROBOTICS IN AGRICULTURE**
    - **Weeding Robots**
    - **Machine Navigation**
    - **Harvesting Robotics**

  - **Material Handling**
- **DRONES IN AGRICULTURE**
- **REMOTE SENSING IN AGRICULTURE**
  - **Crop Monitoring**
  - **Weather conditions**
  - **Soil quality**
- **COMPUTER IMAGING IN AGRICULTURE**
  - **Quality control**
  - **Sorting and grading**
  - **Irrigation Monitoring**

▼ **Explain various security concerns related to IoT? Discuss in detail.**

The Internet of Things (IoT) is the interconnection of various physical devices, vehicles, home appliances, and other devices that have the ability to collect and exchange data. While IoT devices bring numerous benefits and convenience to individuals and organizations, these devices pose significant security challenges. Here are some of the security concerns related to IoT.

1. Weak authentication: Many IoT devices lack robust authentication mechanisms, making them easy targets for hackers. They can easily exploit such devices to launch denial-of-service attacks, disseminate malware or gain unauthorized access to networks.
2. Privacy issues: IoT devices collect and share large volumes of data, including personal information, without the user’s consent or knowledge. Hackers can exploit this information to conduct identity theft or blackmail users.
3. Inadequate software protections: Most IoT devices are built on custom software or firmware, which can be easier for hackers to target and exploit. IoT devices also lack robust security updates, making them susceptible to attacks, even if vulnerabilities are detected.
4. Cybersecurity regulation: Many IoT devices are not subject to cybersecurity regulations, making it easier for manufacturers to produce insecure devices. As a consequence, many consumers unwittingly deploy insecure IoT devices into their networks, creating an environment that can be exploited by hackers.
5. Interconnected systems: IoT devices are interconnected, meaning that a vulnerability in one device can compromise others within the network. This can lead to a domino effect that can result in data breaches or unauthorized access without the user’s knowledge.
6. Lack of vendor transparency: Many IoT devices come from vendors that do not disclose their source code, making it difficult for security analysts to evaluate the security of the devices. This lack of transparency makes it hard for consumers and businesses to assess the security of the IoT devices they purchase.
7. Physical security: IoT devices are vulnerable to physical attacks, which can damage or alter the device’s hardware or firmware. Attackers can exploit this weakness to gain access to

the device or the network and perpetrate damaging attacks.

In conclusion, IoT security concerns need to be addressed proactively to safeguard both the privacy of users and the security of their devices and networks. Manufacturers are encouraged to work collaboratively with security experts and leverage best practices to ensure they build secure IoT devices.

▼ Attacks in IoT system

▼ Vulnerability analysis in IoT

▼ Which kinds of vulnerability have been observed in IoT. Which attacks can exploit the vulnerabilities in Application/Service Layer?

1. Weak or default passwords: Many IoT devices have weak or default passwords, which can be easily cracked by attackers.
2. Insecure communication: IoT devices often communicate over unencrypted or weakly encrypted channels, leaving them vulnerable to eavesdropping and interception.
3. Lack of firmware updates: Many IoT devices do not receive regular firmware updates or security patches, making them vulnerable to known exploits and vulnerabilities.
4. Lack of end-to-end encryption: End-to-end encryption is not always implemented in IoT devices or services, which can leave information exposed during transit.
5. Insecure web applications: Many IoT devices come with web applications that can be easily hacked, giving attackers access to sensitive data.
6. Inadequate access control: Many IoT devices lack adequate access control mechanisms, such as multi-factor authentication or role-based access control, making them vulnerable to unauthorized access.
7. DDoS attacks: IoT devices are often used in botnets to launch DDoS attacks, exploiting their weak security to take down websites and online services.
8. Malware attacks: IoT devices can become infected with malware, which can be used to steal data or launch further attacks.
9. DNS attacks: IoT devices can be used in DNS attacks, redirecting users to malicious websites or stealing sensitive information.
10. Cryptojacking: IoT devices with weak security can be exploited by attackers to mine cryptocurrency, using the device's computing power without the owner's knowledge or consent.

▼ Explain the applications of IoT in home automation systems.

Internet of Things (IoT) technology enables the connectivity of devices and appliances within homes to create a smart home experience. The applications of IoT in home automation systems are many and varied, and some of the most common ones include:

1. Security: IoT-enabled security systems allow homeowners to remotely manage their security systems (like surveillance cameras, door locks) and monitor their home even when they’re away from home.

2. Energy efficiency: IoT-enabled home appliances and systems (like smart thermostats, lighting systems, and HVAC systems) can be automated and controlled remotely, which helps reduce energy consumption, and, in turn, reduces energy bills.
3. Convenience: IoT-powered home automation systems, like voice-activated virtual assistants (such as Amazon Alexa or Google Assistant) can help homeowners manage their home settings, such as turning off lights, changing the temperature, or playing music, all by using simple voice commands.
4. Health: IoT devices can help monitor various health metrics (like blood pressure, heart rate, and sleep patterns). These devices can aid in managing long-term illnesses, tracking physical activity, and making healthier choices in daily life.
5. Home Entertainment: Smart home systems can integrate with various entertainment devices and enable systems (like TVs, gaming consoles) and provide seamless entertainment experiences that can be controlled by a remote or voice-activated virtual assistant.

Overall, IoT-enabled home automation systems offer homeowners greater convenience, security, cost savings, energy efficiency, and improved quality of life.