1. IOT CONCEPTS

1.1. Introduction

- IoT stands for Internet of Things.
- It refers to the interconnectedness of physical devices, such as appliances and vehicles that are embedded with software, sensors, and connectivity which enables these objects to connect and exchange data.
- This technology allows for the collection and sharing of data from a vast network of devices, creating opportunities for more efficient and automated systems.
- In the upcoming years, IoT-based technology will offer advanced levels of services and practically change the way people lead their daily lives.

1.2. Definition

IoT is network of interconnected computing devices which are embedded in everyday objects, enabling them to send and receive data.

1.3. Working of IoT

- Devices have hardware, like sensors, that collect data.
- The data collected by the sensors is processed by the processors.
- The processed data is then shared via the cloud and integrated with software.
- The software then analyzes and transmits the data to users via an app or website.

1.4. Building blocks of IoT

- Five things form basic building blocks of the IoT system –Things or Device, gateways, cloud, analytics and user interface.
- Each of these nodes has to have its own characteristics in order to form a useful IoT system.

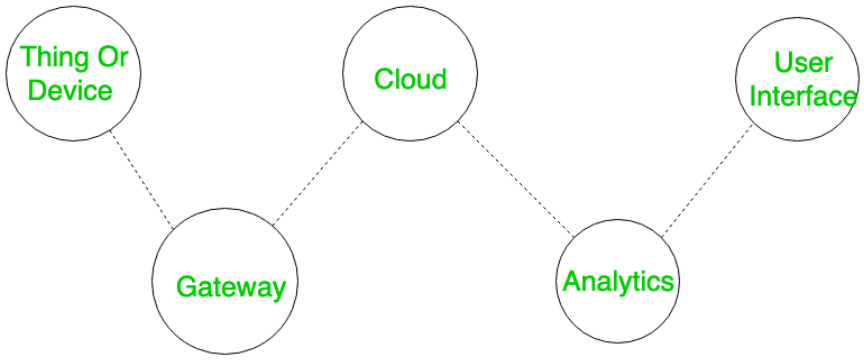

Figure 1: Building Blocks of IoT

Things or Device

- These are fitted with sensors and actuators.
- Sensors collect data from the environment and give to gateway whereas actuators performs the action (as directed after processing of data).

Gateway

- The sensors give data to Gateway and here some kind of pre-processing of data is even done.
- It also acts as a level of security for the network and for the transmitted data.

Cloud

- The data after being collected is uploaded to cloud.
- Cloud in simple terms is basically a set of servers connected to internet 24*7.

Analytics

- The data after being received in the cloud processing is done.
- Various algorithms are applied here for proper analysis of data (techniques like Machine Learning etc. are even applied).

User Interface

- User end application where user can monitor or control the data.

1.5. Advantages of IoT

- It can assist in the smarter control of homes and cities via mobile phones. It enhances security and offers personal protection.
- By automating activities, it saves us a lot of time.
- Information is easily accessible, even if we are far away from our actual location, and it is updated frequently in real time.
- Electric Devices are directly connected and communicate with a controller computer, such as a cell phone, resulting in efficient electricity use. As a result, there will be no unnecessary use of electricity equipment.
- It minimizes human effort because IoT devices connect and communicate with one another and perform a variety of tasks without the need for human intervention.
- Patient care can be performed more effectively in real time without the need for a doctor's visit. It gives them the ability to make choices as well as provide evidence-based care.
- Asset tracking, traffic or transportation tracking, inventory control, delivery, surveillance, individual order tracking, and customer management can all be made more cost-effective with the right tracking system.

1.6. Disadvantages of IoT

- Hackers may gain access to the system and steal personal information. Since we add so many devices to the internet, there is a risk that our information as it can be misused.
- They rely heavily on the internet and are unable to function effectively without it.
- With the complexity of systems, there are many ways for them to fail.
- Unskilled workers are at a high risk of losing their jobs, which could lead to unemployment. Smart surveillance cameras, robots, smart ironing systems, smart washing machines, and other facilities are replacing security guards, maids, ironmen, and dry-cleaning services etc.
- It is very difficult to plan, build, manage, and enable a broad technology to IoT framework.
- Deploying IoT devices is very costly and time-consuming.

2. STANDARDS OF IOT

- Smart objects produce large volumes of data.
- This data needs to be managed, processed, transferred and stored securely.
- Standardization is key to achieving universally accepted specifications and protocols for true interoperability between devices and applications.
- The use of standards:
  - ensures interoperable and cost-effective solutions
  - opens up opportunities in new areas
  - allows the market to reach its full potential
- ISO/IEC and IEEE has developed standards in this area and bring all security requirements to a single universal form.
- Standards provide people and organizations with a basis for a mutual understanding of the IoT.
- This ranges from standards for calibration of gas flow meters, over standards for compliance of wireless communication, to standards for IoT implementations and considerations.
- Following are the few standards by ISO/IEC and IEEE for developing IoT based infrastructure

Standards	**Standard Description**	
Standards pertaining to IoT architecture		
ISO/IEC 30141	Internet of Things Reference Architecture	• Provides a common framework for IoT application designers and developers and facilitating the development of reliable systems • It highlights functional requirement such as Data Management, Device Management, Security, Confidentiality and privacy, • Highlights non-functional requirement such as maintainability, reliability, usability, high availability, and scalability of your system.
ISO/IEC 30149 ED1	Trustworthiness Principles	• Provides methodology for implementing and maintaining trustworthiness of IoT systems and services.
ISO/IEC 30161-1 ED1	Data exchange platform for IoT Services-Part 1: General requirements and architecture	• Specifies requirements for an Internet of Things (IoT) data exchange platform for various services in the technology areas.
ISO/IEC 30165	Internet of Things(Real time IoT Framework)	• Provides a guideline for deploying an RT-IoT system to avoid pitfalls that usually occur during real-time system developments. • It focuses on real-time capability in addition to very general description because failing on timing constraints could cause serious damage to an IoT system or to its environment, including injury or even death of people involved.

IEEE P2413	Standard for an architectural framework for IoT	• Defines an architectural framework for the Internet of Things (IoT), including descriptions of various IoT domains, definitions of IoT domain abstractions, and identification of commonalities between different IoT domains.
IEEE 802.15.4-2015	IEEE Standard for Low Rate Wireless Networks	• Defines protocol and compatible interconnection for data communication devices using low data-rate, low-power, and low-complexity short-range radio frequency (RF) transmissions in a wireless personal area network (WPAN) are defined in this standard.
Standards associated to Interoperability for IoT		
ISO/IEC 21823-1	Internet of Things- Interoperability for IoT Systems-Part 1: Framework	• Is a series which addresses issues that relate to interoperability of the communications between IoT systems entities, both between different IoT systems and within a single IoT system.
ISO/IEC 21823-2	Internet of Things- Interoperability for IoT Systems-Part 2: Transport Interoperability Standard	• Specifies a framework and requirements for transport interoperability, in order to enable the construction of IoT systems with information exchange, peer-to-peer connectivity and seamless communication both between different IoT systems and also among entities within an IoT system.
ISO/IEC 21823-3	Internet of Things- Interoperability for IoT Systems-Part 3: Semantic Interoperability Standard	• Is the facet which enables the exchange of data between IoT systems using understood data information models
Standards associated to security and privacy		
IEEE1451-99	Standard for harmonization and security of IoT	• Utilizes the advanced capabilities of the XMPP protocol, such as providing globally authenticated identities, authorization, presence, life cycle management, interoperable communication, IoT discovery and provisioning.
IEEE P1912	Standard for Privacy and Security Architecture for Consumer Wireless Devices	• Defines by use of a common communication architecture for diverse wireless communication devices such as, but not limited to, devices equipped with near field communication (NFC), home area network (HAN), wireless area network (WAN) and wireless personal area network (WPAN) technologies, or radio frequency identification technology (RFID), and the proximity considerations attendant to these areas.
ISO/IEC 27400	Cybersecurity-IoT security and privacy-Guidelines	• Provides guidance on the principles, information risk and controls for IoT security and privacy. It also provide guidance on security features expected of all IoT devices
ISO/IEC 27402	Cybersecurity-IoT security and privacy-Device baseline requirements	• Provides guidance on the basic, commonplace security features expected of all IoT devices, enabling the IoT security controls
ISO/IEC27402.2	Cybersecurity-IoT security and privacy-Guidelines for IOT domotics	• Provides information security and privacy of IoT for home use is a challenge given the variety of things, home circumstances, security and privacy issues and controls

3. COMPONENTS OF IOT

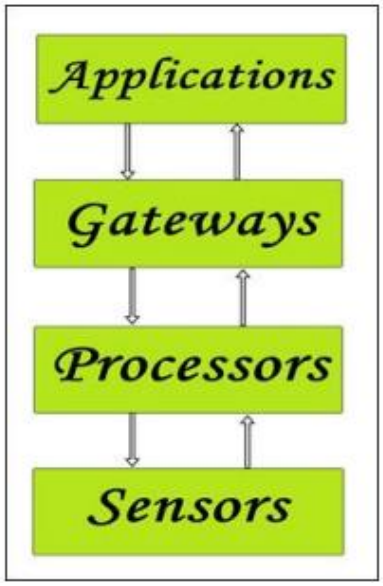

Figure 2: Components of IoT

Sensors

- These form the front end of the IoT devices.
- These are the so-called “Things” of the system.
- Their main purpose is to collect data from its surroundings (sensors) or give out data to its surrounding (actuators).
- These have to be uniquely identifiable devices with a unique IP address so that they can be easily identifiable over a large network.
- These have to be active in nature which means that they should be able to collect real-time data.
- These can either work on their own (autonomous in nature) or can be made to work by the user depending on their needs (user-controlled).
- Examples of sensors are gas sensor, water quality sensor, moisture sensor, etc.

Processors:

- Processors are the brain of the IoT system.
- Their main function is to process the data captured by the sensors and process them so as to extract the valuable data from the enormous amount of raw data collected. In a word, we can say that it gives intelligence to the data.
- Processors mostly work on real-time basis and can be easily controlled by applications.
- These are also responsible for securing the data – that is performing encryption and decryption of data.
- Embedded hardware devices, microcontroller, etc are the ones that process the data because they have processors attached to it.

Gateways:

- Gateways are responsible for routing the processed data and send it to proper locations for its (data) proper utilization.
- In other words, we can say that gateway helps in to and fro communication of the data. It provides network connectivity to the data. Network connectivity is essential for any IoT system to communicate.
- LAN, WAN, PAN, etc are examples of network gateways.

Applications:

- Applications form another end of an IoT system.
- Applications are essential for proper utilization of all the data collected.
- These cloud-based applications which are responsible for rendering the effective meaning to the data collected.
- Applications are controlled by users and are a delivery point of particular services.
- Examples of applications are home automation apps, security systems, industrial control hub, etc.

4. RELEVANCE OF IOT FOR THE FUTURE

IoT is important because it allows the following features

Better Decision Making

- Since devices have multiple sensors, they can acquire considerable data from numerous sources, giving them more information to work with when acting on data received.
- It opens up the possibility of making decisions based on the collected data.

Real-time Tracking and Monitoring

- The potential for web-based tracking and monitoring systems is enormous.
- IoT tracking provides an efficient means to track and monitor anything from vehicle fleets, stolen goods, or shipping containers.
- Particular devices can even detect changes in the environment. There are multiple industries where IoT trackers can immensely improve the efficiency of companies

Automation

- A big reason for the invention of IoT is convenience.
- Smart devices that automate daily tasks allow humans to do other activities. These devices ultimately lighten people's workload.
- An example is a self-driving car, connecting to the Internet to find the quickest route to a destination. This is the ultimate convenience for humans. The room for innovation within IoT is massive.

More Efficient Personal and Business Tasks

- Web-based devices save people money and time.
- This includes planning work schedules, time tracking, effective communication, and setting reminders for daily tasks.

5. IOT APPLICATIONS

- Home
- Cities
- Environment
- Energy
- Retail
- Logistics
- Agriculture
- Industry
- Health & Lifestyle

6. IOT FOR SMART CITIES

- Cities can be made more efficient so that they require fewer resources and are more energy-efficient.
- This can be done with a combination of sensors in different capacities all over the city that can be used for various tasks ranging from
  - managing the traffic,
  - controlling and handling waste management,
  - creating smart buildings,
  - Optimizing streetlights, etc.

Traffic Management

It enables them to;

- Expand the capacity of city streets without having to build new roads.
- Optimize the traffic flow and keep the drivers safe. It would include cameras, sensors, and cellular technologies that automatically adjust traffic lights, expressway lanes, speed limits, and highway exit counters.
- Transmit accurate information about available parking spaces to citizens in real-time
- Collect data on congestion and improve traffic signalling to reduce blockages and optimize commute
- Locate incidents and report them to emergency rooms immediately with road sensors and video surveillance
- Employ real-time data feeds to ensure the streetlights turn dim or brighten up per the changing weather conditions and the onset of day and night

Waste management

- IoT devices turn this model on its head by using smart trash bins to detect location, temperature, and fill level in real time.
- This data is then used to plan optimal collection routes, resulting in an efficient pickup process that saves fuel as well as manpower.
- Additionally, data helps with long-term planning, such as where more bins are needed or where the number can be reduced.
- The data gathered from smart bins also reduces the number of missed pickups or the incidence of overflowing trash bins.
- If a sensor detects that a bin is full, an automatic alert will be sent to waste managers, who can schedule an extra pickup.

Smart Buildings

- IoT in smart buildings simplifies tasks such as:
  - Building temperature control,
  - Smart water usage,
  - Pest control,
  - Fire detection,
  - Security and access control, and
  - Structural health monitoring

- One example of creating smart cities is the Smart Nation Sensor Platform used by Singapore which is believed to be the smartest city in the world. This platform integrates various facets of transportation, streetlights, public safety, urban planning, etc. using sensors in conjugation with IoT.

7. CHALLENGES IN IOT IMPLEMENTATION

- **Scalability**
  - Addressing issues as so many devices are connected
  - Difficulty to manage and maintain these devices and ensure that they are functioning optimally
  - Understanding the data requires big data analytics and cloud storage

- **Technological Standardization and Inter-operability**
  - Technological standards in most areas are still fragmented.
  - These technologies need to be converged. This would help us in establishing a common framework and the standard for the IoT devices.
  - As the standardization process is still lacking, interoperability of IoT with legacy devices is difficult. This lack of interoperability prevents from truly connected everyday interoperable smart objects.

- **Security and Privacy**
  - As billions of connected devices collect and transmit sensitive information, it is essential that these devices and their data are protected from malicious attacks and hacking.
  - Requires strong security measures, such as encryption and secure protocols
  - Requires regular software updates to fix vulnerabilities.

- **Design Based Challenges**
  - Requires the development of accessible and inclusive design principles,
  - Limited computation power, limited energy and limited memory issues

- **Lack of Government Support**
  - Many governments are still slow to provide the necessary support and resources to help the industry grow.
  - Companies need to work with governments to develop regulations that support innovation while still protecting privacy and security.

Open issue	Brief description of the cause
Standards	There are several standardization efforts but they are not integrated in a comprehensive framework
Mobility support	There are several proposals for object addressing but none for mobility support in the IoT scenario, where scalability and adaptability to heterogeneous technologies represent crucial problems
Naming	Object Name Servers (ONS) are needed to map a reference to a description of a specific object and the related identifier, and *vice versa*
Transport protocol	Existing transport protocols fail in the IoT scenarios since their connection setup and congestion control mechanisms may be useless; furthermore, they require excessive buffering to be implemented in objects
Traffic characterization and QoS support	The IoT will generate data traffic with patterns that are expected to be significantly different from those observed in the current Internet. Accordingly, it will also be necessary to define new QoS requirements and support schemes
Authentication	Authentication is difficult in the IoT as it requires appropriate authentication infrastructures that will not be available in IoT scenarios. Furthermore, things have scarce resources when compared to current communication and computing devices. Also man-in-the-middle attack is a serious problem
Data integrity	This is usually ensured by protecting data with passwords. However, the password lengths supported by IoT technologies are in most cases too short to provide strong levels of protection
Privacy	A lot of private information about a person can be collected without the person being aware. Control on the diffusion of all such information is impossible with current techniques
Digital forgetting	All the information collected about a person by the IoT may be retained indefinitely as the cost of storage decreases. Also data mining techniques can be used to easily retrieve any information even after several years

8. CHARACTERISTICS OF IOT

Characteristics	**Description**
Dynamic & Self Adapting	Devices have the capability to dynamically adapt with the changing contexts and take actions based on their operating conditions, user's context or sensed environment
Self-Configuring	Allows a large number of devices to work together to provide certain functionality
Interoperable Communication Protocols	Supports a number of interoperable communication protocols and can communicate with other devices and also with infrastructure.
Unique Identity	Each IoT device has a unique identity and a unique identifier
Integrated into information network	Allows the devices to communicate and exchange data with other devices and systems.

9. PHYSICAL DESIGN OF IOT

- A physical design of an IoT system refers to the individual node devices and their protocols that are utilised 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 things/devices in the IoT system are used for:
    - Building connections
    - Data processing
    - Providing storage
    - Providing interfaces
    - Providing graphical interfaces
- 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.
- The main two parts of the physical design are:
    - Individual node devices
    - IoT Protocols

Individual node devices

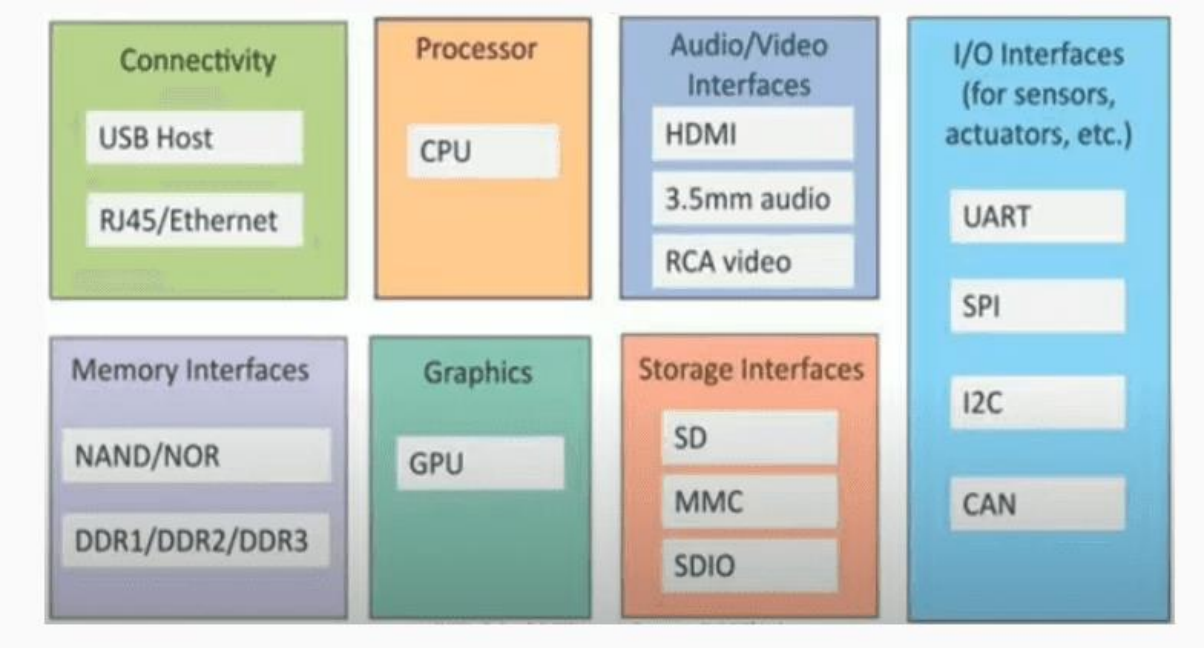

Figure 3: Individual Node Devices- Hardware Interface

- An IoT device may consist of several interfaces connections to other devices, both wired and wireless. These include
    - IoT interfaces for sensors
    - Interfaces for internet connectivity
    - Memory and storage interfaces
    - Audio/video interfaces

IoT protocols

- The set of rules governing all direct or indirect exchange of data between computers on a network.
- These rules are formulated at the application level and are used collectively to define how devices communicate interoperably, irrespective of differences in their internal designs and operations.
- IoT protocols help send commands and data between a network of devices controlled by sensors or other physical attributes like motion, temperature, or vibration.

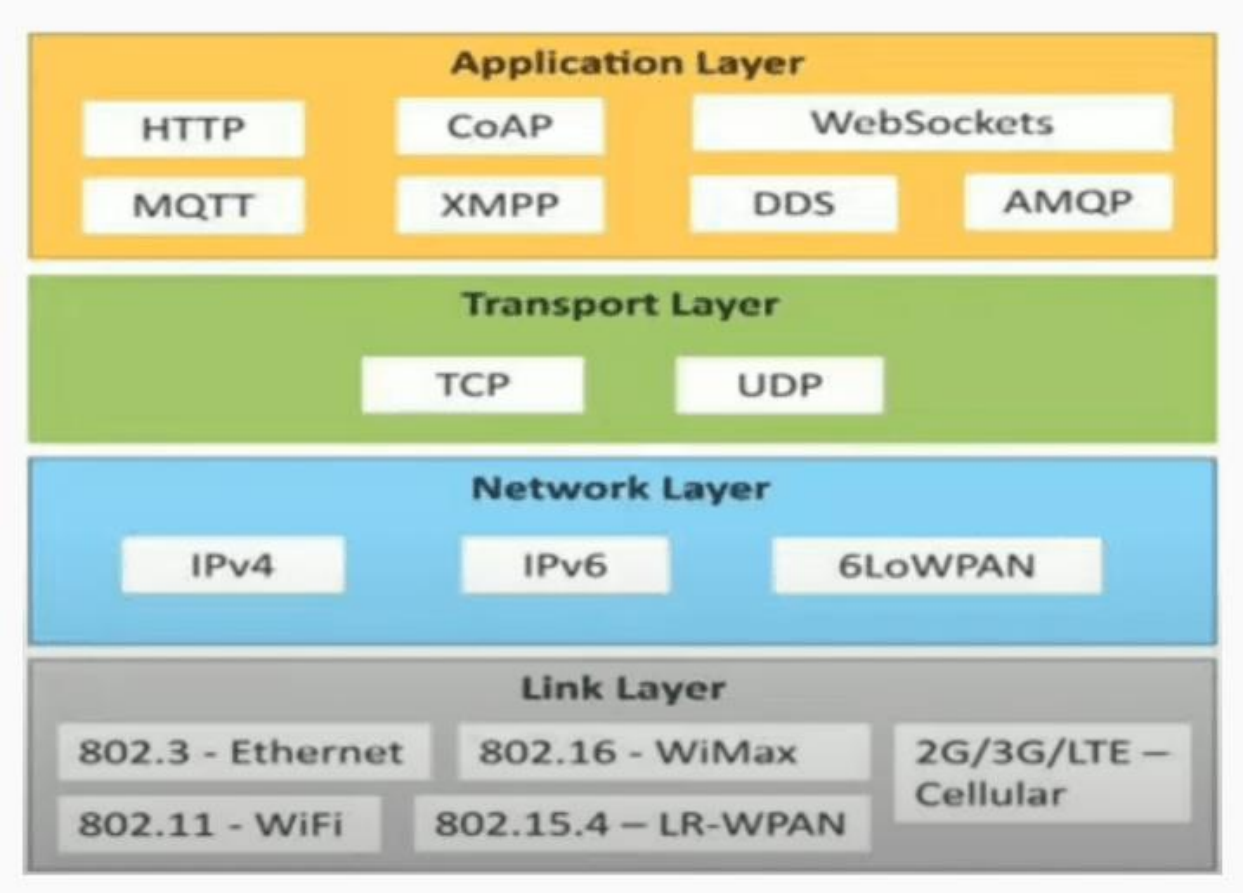

Figure 4: IoT Protocols

Application Layer Protocols

On this layer, protocols use an application interface to define how the data can be sent over the network.
These protocols include HTTP, XMPP, Web Socket, DDS, MQTT, and AMQP.

Protocols	Abbreviations	Description
HTTP	**Hyper Text Transfer Protocol**	• Follows client server architecture • Follows request response model • Uses TCP and can also adapt to UDP • Is a stateless protocol
XMPP	**Extensible Messaging Presence Protocol**	• Based on client-server architecture • Allows sending small chunks of XML Data from one network entity to another in real time. • Designed for sending messages in real time. • Used in Instant messaging apps
DDS	**Data Distribution Service**	• Is a data centric middleware standard for device to device machine to machine • Uses Publish Subscribe Model • Provides Quality of Service (QoS) control and configurable reliability
AMQP	**Advanced Message Queuing Protocol.**	• Open application layer protocol • Used for business messaging • Based on Publish Subscribe Model
MQTT	**Message Queuing Telemetry Transport**	• Uses Transmission Control Protocol • Based on Publish Subscribe Model • Follows Client Server Architecture
CoAP	**Constrained Application Protocol**	• Used in Machine to Machine (M2M) • It uses Request-Response model. • Uses User Datagram protocol (UDP)
Websockets		• Full Duplex Communication • Based on TCP • Allows stream of messages to be sent back and forth to the client and server while keeping TCP Connection open

Transport Layer Protocols

- This layer is responsible for data flow control and error handling, ensuring that there are rules in place to deal with errors.
- This layer also provides end-to-end message transfer capability, independent of the underlying network infrastructure.
- It provides essential connectivity between the two nodes on either end of the point-to-send-point-receive model used by key protocols such as TCP or UDP.
- TCP/IP
    - Connection oriented
    - Stateful Protocol
    - IP protocol deals with sending packets
    - Ensures reliable transmissions of packets in order
    - Provides error detection capability and hence duplicate packets are discarded

- UDP
    - Used where the application has very small units of data to exchange and do not want overhead of connection setup
    - Useful for time sensitive applications
    - Connectionless protocol
    - Stateless Protocol
    - Does not provide guarantee of delivery, ordering of messages and duplicate eliminations

Network Layer

- This layer is used to send data from a source network to a destination network. For this, IPv4 and IPv6 protocols are used for host identification, which transfers data in packets.

IPv4

- It uses 32bit address scheme that allows total of 2^32 addresses

IPv6

- It uses 128bit address scheme that allows total of 2^128 addresses

6LoWPAN

- Defines an approach for routing Internet Protocol version 6 (IPv6) over low-power wireless networks.

Link Layer

- Link-layer protocols are the type of data transmission protocol used to help send data over the physical layer. They also determine how devices signal and code packets on the network.

802.3 Ethernet	• Collection of Wired Ethernet connection
	• Shared medium-coaxial cable, twisted pair wire, Optical Fibre
	• Data rates: 10Mbps -40Gbps
802.1- WiFi	• Collection of Wireless local area network (WLAN)
	• Includes extensive description of the following layers
	• 802.11a : Operates in 5GHz band
	• 802.11b & 802.11g operate in 2.4GHz band
	• 802.11ac operates in 5GHz band
802.16 wiMAX	• Collection of Wireless broadband and Standards
	• Provides data rates from 1.5Mbps to 1Gbps
802.15.4. LR-WPAN	• Collection of standard for low rate wireless personal area network
	• Provides data rates from 40kbps
	• Provides Low Cost and low speed communications for power constrained devices.
2G/3G/4G Mobile Communications	• Different generations of mobile communication standards

10. 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.
- It doesn't go in depth about describing how each component will be built
- It focuses on satisfying Design Factors, Risks, Requirements, Constraints and Assumptions
- Logical Design of IOT includes
    - IoT Functional Blocks
    - IoT Communication Models
    - IoT Communication APIs

IoT functional blocks:

- IoT systems include several functional blocks such as Devices, communication, security, services, and application.
- The functional blocks provide sensing, identification, actuation, management, and communication capability.
- These functional blocks consist of the following:
    - Devices that handle the communication between the server and the host
    - Enable monitoring control functions,
    - Manage the data transfer,
    - Secure the IoT system using authentication and different functions
    - Provide an interface for controlling and monitoring various terms

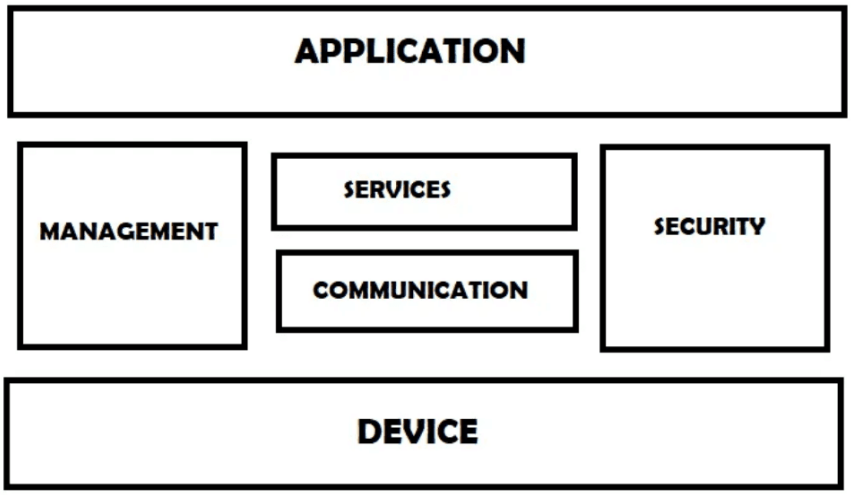

Figure 5: Functional Blocks of IoT

IoT Communication Models

There are multiple kinds of models available in an Internet of Things system that is used for communicating between the system and server, such as:

Request-response model

- Follows client server model.
- Client sends a request to the server.
- When the server receives the request it decides how to respond, fetches the data, retrieves resources, prepares the response and sends it to the client.
- Example: HTTP

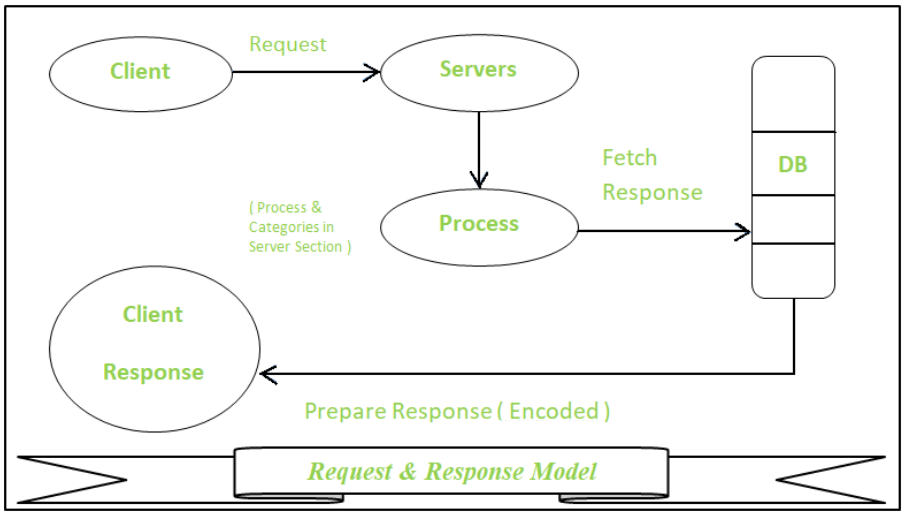

Figure 6: Request Response Model

Push-pull model

The push-pull model constitutes data publishers, data consumers, and data queues.

- **Publishers** and **Consumers** are not aware of each other.
- 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.
- **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.

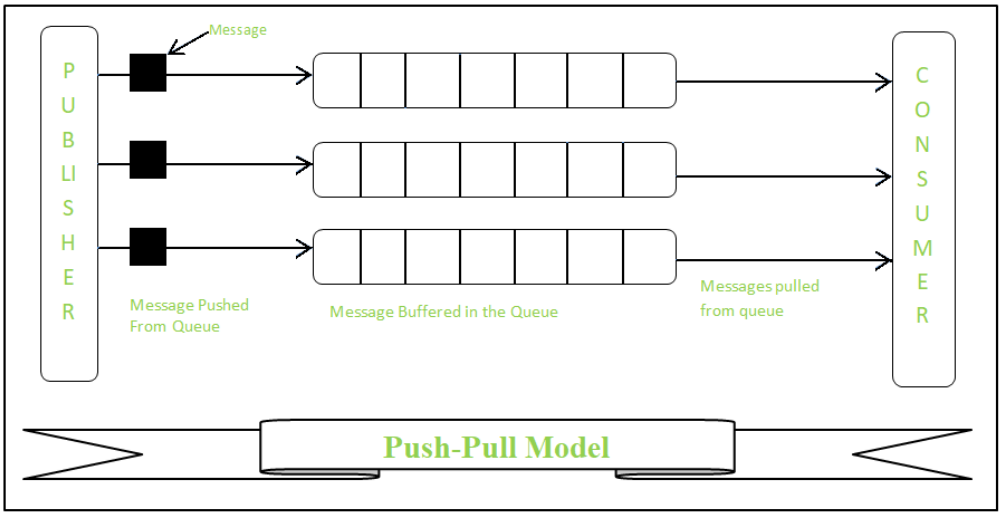

Figure 7: Push Pull Model

Publish-subscribe model

This model comprises three entities: Publishers, Brokers, and Consumers.

- **Publishers** are the source of data. It sends the data to the topic which are managed by the broker. They are not aware of consumers.
- **Consumers** subscribe to the topics which are managed by the broker.
- **Brokers**
  - Accept data from publishers and send it to the appropriate consumers.
  - The broker only has the information regarding the consumer to which a particular topic belongs to, which the publisher is unaware of.

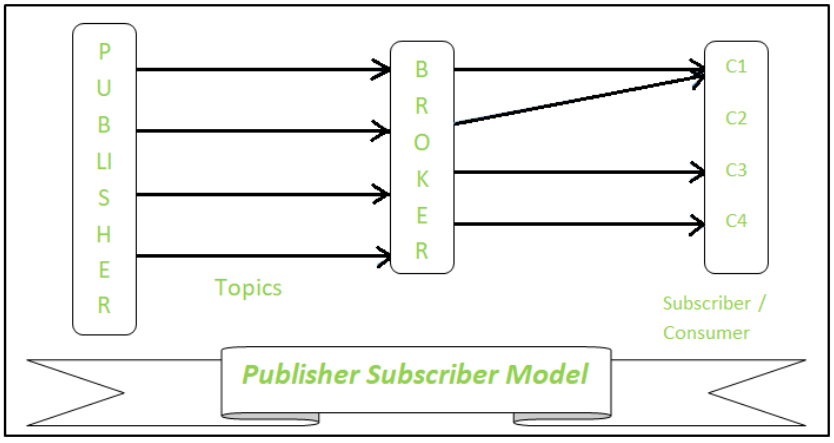

Figure 8: Publish Subscribe Model

Exclusive pair model

- Client server based model
- Exclusive Pair is the bi-directional model, including full-duplex communication among client and server. The connection is constant and remains open till the client sends a request to close the connection.
- The Server has the record of all the connections which has been opened.
- This is a state-full connection model and the server is aware of all open connections.

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

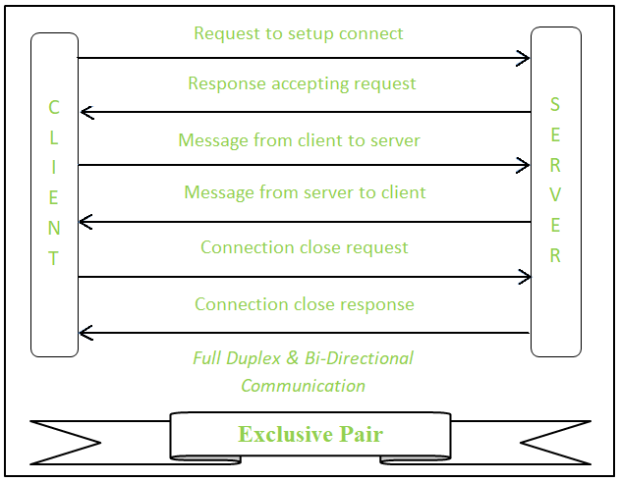

Figure 9: Exclusive Pair Model

IOT APIs

- An API is an interface used by programs to access an application.
- It enables a program to send commands to another program and receive replies from the app.
- IoT APIs are the interface points between an IoT device and the Internet and/or other network components.
- Generally we use 2 APIs for IOT Communication. They are:
  - REST Based APIs
  - Web socket based APIs

REST-based APIs

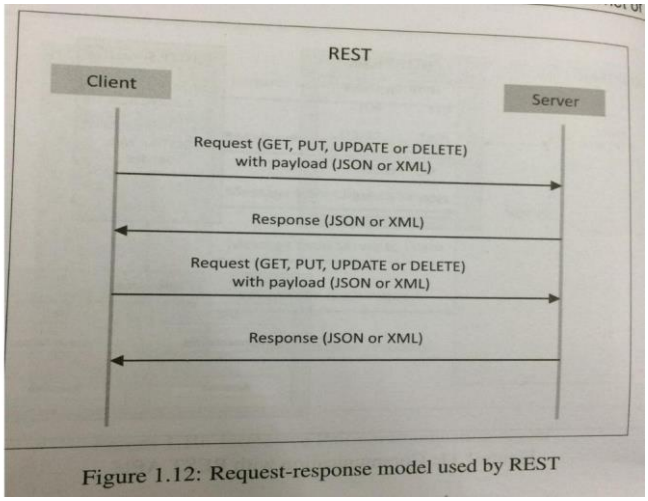

Figure 10: REST API

- **Representational state transfer (REST)** is a set of architectural principles by which you can design Web services and Web APIs that focus on the system's resources and how resource states are addressed and transferred.
- The REST APIs follow the request-response model.

- Client tries to access these resources via URIs using commands like GET, PUT, POST, DELETE and so on that are defined by HTTP.
- In response, the server responds with a JSON object or XML file.
- The rest architectural constraints are as follows:
    - **Client-server Seperation**: The client should not be concerned with the storage of data which is a concern of the server, similarly, the server should not be concerned about the user interface, which is the concern of the client. Separation makes it possible for the client and server to be developed and updated independently.
    - **Stateless**: The status of the session depends entirely on the client. It will not store any data. The request from client to server must contain all the information to understand the request.
    - **Cache-able:** This property defines whether the response to any request can be cached or not. If a response can be cached, then a client cache is granted the right to reuse that response data for subsequent matching requests.
    - **Layered system:** A layered system defines the boundaries of the components within each specific layer. For example, a client is unable to tell whether it is connected to the end server or an intermediate node.
    - **Uniform interface:** This specifies that the technique of communication between a client and a server must be uniform throughout the communication period.
    - **Code on Demand (Optional Constraint):** Servers may provide executable code or scripts for execution by clients in their context.

Web socket based APIs

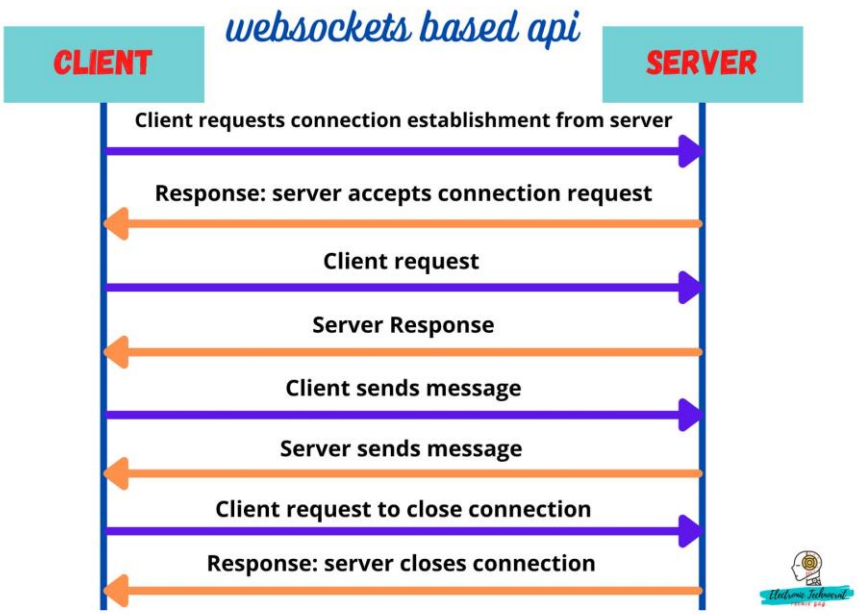

Figure 11:Websocket based API

- Web socket APIs enable bi-directional and duplex communication between customers and servers.

- Unlike REST, There is no need to set up a connection every now and then to send messages between a client and a server.
- It works on the principle of the exclusive pair model
- It is a Stateful type i.e. it stores data.
- Due to one-time dedicated connection setup, there is less overhead, lower traffic and less latency and high throughput.
- So Web socket is the most suitable IoT Communication APIs for IoT System.

Difference between Rest API and Web Socket API:

S.NO.	REST API	WEB SOCKET API
1.	It is Stateless protocol. It will not store the data.	It is Stateful protocol. It will store the data.
2.	It is Uni-directional. Only either server or client will communicate.	It is Bi-directional. Messages can be received or sent by both server and client.
3.	It is Request-response model.	It is Full duplex model.
4.	HTTP request contains headers like head section, title section.	It is suitable for real-time applications. It does not have any overhead.
5.	New TCP connection will be set up for each HTTP request.	Only Single TCP connection.
7.	It depends upon the HTTP methods to retrieve the data.	It depends upon the IP address and port number to retrieve the data
8.	It is slower than web socket regarding the transmission of messages.	Web socket transmits messages very quickly than REST API.
9.	It does not need memory or buffers to store the data.	It requires memory and buffers to store the data.

Parameters	Physical Design	Logical Design
Definition	A physical design of an IoT system refers to the individual node devices and their protocols that are utilised to create a functional IoT ecosystem.	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.
Details	Provides Detailing	Does not provide detailing
Focus	Focuses on explaining how things are configured or assembled	Focuses on satisfying Design Factors, Risks, Requirements, Constraints and Assumptions