Unit 1 : Introduction to IoT

1. IoT Defination:

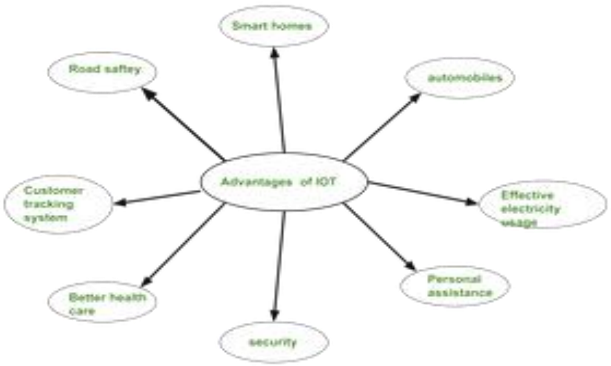
- IoT stands for Internet of Things. It refers to the interconnectedness of physical devices, such as appliances and vehicles,that are embedded with software, sensors, and connectivity which enables these objects to connect and exchange data. This technology allows for the collection and sharing of data from a vast network of devices, creating opportunities for more efficient and automated systems.
- IoT Refers to the interconnection of everyday devices and objects to the internet, allowing them to collect and exchange data. IoT has revolutionized industries and everyday life, offering a wide range of advantage, but it also comes with several disadvantages.
- The **Internet of Things (IoT)** is a network of interconnected, embedded devices that can capture and transmit data without the need for human interaction over a wireless network.

1.1: Advantages and Disadvantages of IoT:

IoT applications in everyday life include smart wearables, smart health monitoring, traffic monitoring, IoT in agriculture with many sensors, smart devices, robots in hospitals, smart grid and water supply, and so on.

Advantages:

- 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.
- Personal assistance can be provided by IoT apps, which can alert you to your regular plans.
- It is useful for safety because it senses any potential danger and warns users. For example, GM OnStar, is a integrated device that system which identifies a car crash or accident on road. It immediately makes a call if an accident or crash is found.
- 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.



Disadvantages:

- 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.
- We lose control of our lives—our lives will be fully controlled and reliant on technology.
- Overuse of the Internet and technology makes people unintelligent because they rely on smart devices instead of doing physical work, causing them to become lazy.
- 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.

- **Advantages & Disadvantages of IoT(Tabular Format)**

Advantages	Disadvantages
Minimizes the human work and effort	Increased privacy concerns
Saves time and effort	Increased unemployment rates
Good for personal safety and security	Highly dependent on the internet
Useful in traffic and other tracking or monitoring systems	Lack of mental and physical activity by humans leading to health issues.
Beneficial for the healthcare industry	Complex system for maintenance
Improved security in homes and offices	Lack of security
Reduced use of many electronic devices as one device does the job of a lot of other devices	Absence of international standards for better communication

---

1.3: Characteristics of IoT

1. **Connectivity:** IoT relies on the internet to connect devices and enable communication and data exchange.
2. **Intelligence:** IoT devices can process data, extract insights, and make decisions, often with the help of embedded sensors and analytics.
3. **Scalability:** IoT systems can accommodate a large number of devices and data sources, making them adaptable to different needs.
4. **Dynamic Adaptability**: IoT devices can adjust to changing environments and operating conditions.
5. **Interoperability**: Different IoT devices and systems should be able to work together seamlessly.
6. **Security:** Protecting data and preventing unauthorized access to devices is crucial in IoT.
7. **Automation:** IoT enables automation of tasks and processes, reducing the need for human intervention.
8. **Data Collection and Analysis:** IoT devices generate vast amounts of data, which can be analyzed to gain valuable insights.
9. **Real-time Capabilities**: Some IoT applications require real-time data processing and responses.
10. **Unique Identity**: Each IoT device typically has a unique identity that allows it to be differentiated and managed.
11. **Ubiquity:** The ability for devices to connect and communicate from anywhere at any time.
12. **Self-Configuring:** The ability for devices to upgrade software and configure themselves with minimal

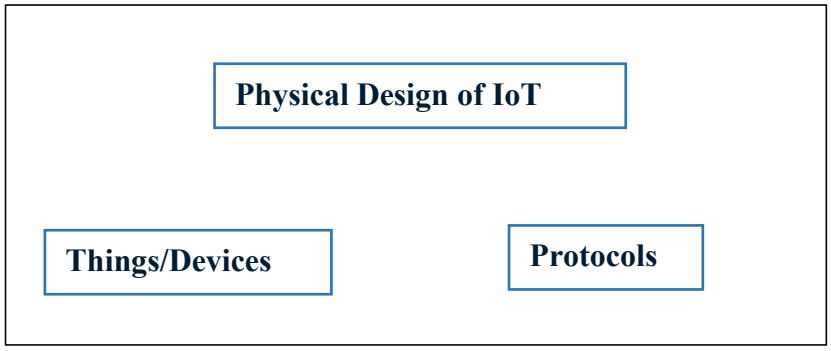
user intervention.
13. **Hybrid Architecture:** IoT systems often involve a combination of hardware, software, and network components.
14. **Energy Management:** Many IoT devices are designed to be energy-efficient to conserve power.

---

2. Physical Design of IoT

The Physical design of an IoT system is referred to as the things/Devices and protocols that are used to build an IoT system. All theses things/devices are called n ode devices and every device has a unique identity that performs remote sensing, actuation and monitoring work and the protocols that are used to establish communication between the node devices and servers over the internet.

Dig. Types of IoT



Physical Design of

1. Things/Devices:

Things/Devices are used to build a connection, process data, provide interfaces, provide storage, and provide graphics interfaces in an IoT system. all these generate data in a form that can be analyzed by an analytical system and program to perform operations and used to improve the system.

for example temperature sensor that is used to analyze the temperature generates the data from a location and is then determined by algorithms.

Connectivity: Devices like USB hosts and ETHERNET are used for connectivity between the devices and the server.

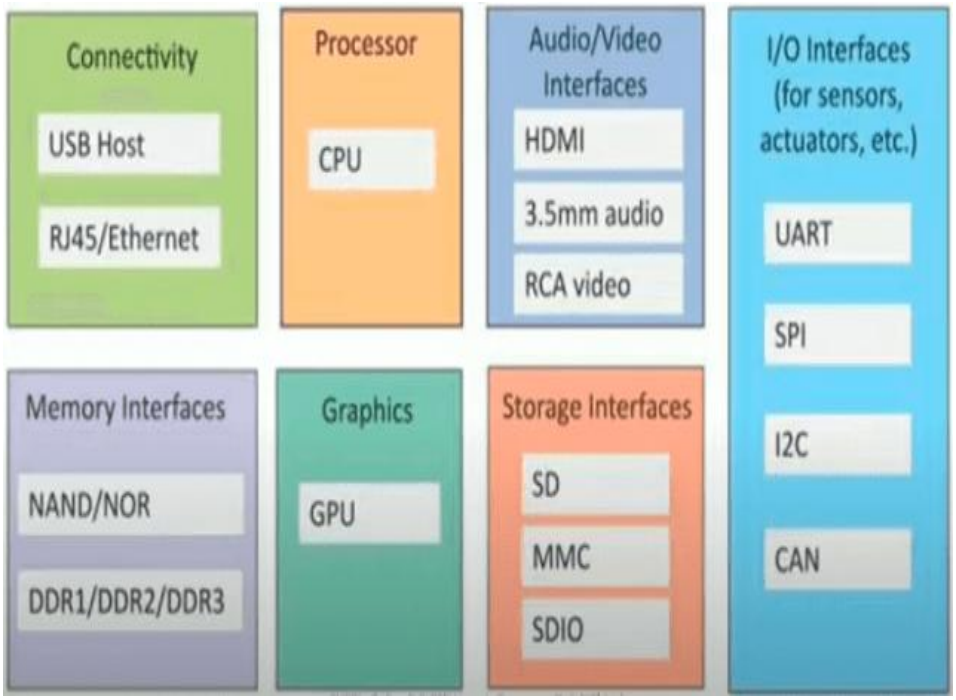
Processor: A processor like a CPU and other units are used to process the data. these data are further used to improve the decision quality of an IoT system.

Audio/Video Interfaces: An interface like HDMI and RCA devices is used to record audio and videos in a system.

Input/Output interface: To give input and output signals to sensors, and actuators we use things like UART, SPI, CAN, etc.

Storage Interfaces: Things like SD, MMC, and SDIO are used to store the data generated from an IoT device.

Other things like DDR and GPU are used to control the activity of an IoT system.



Dig. Generic Block diagram of IoT Devices

2. IoT Protocols

These protocols are used to establish communication between a node device and a server over the internet. it helps to send commands to an IoT device and receive data from an IoT device over the internet. we use different types of protocols that are present on both the server and client side and these protocols are managed by network layers like application, transport, network, and link layer.

- **Application Layer protocol :** In this layer, protocols define how the data can be sent over the network with the lower layer protocols using the application interface. these protocols include HTTP, WebSocket, XMPP, MQTT, DDS, and AMQP protocols.

1. **HTTP :** Hypertext transfer protocol is a protocol that presents in an 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 the two requests.

2. **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.

3. **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 error control. also, these layer protocols provide end-to-end message transfer capability independent of the underlying network.

1. **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.

2. **UDP :** a user datagram protocol is a part of an internet protocol called the connectionless protocol. this protocol is 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 host identification that transfers data in packets.

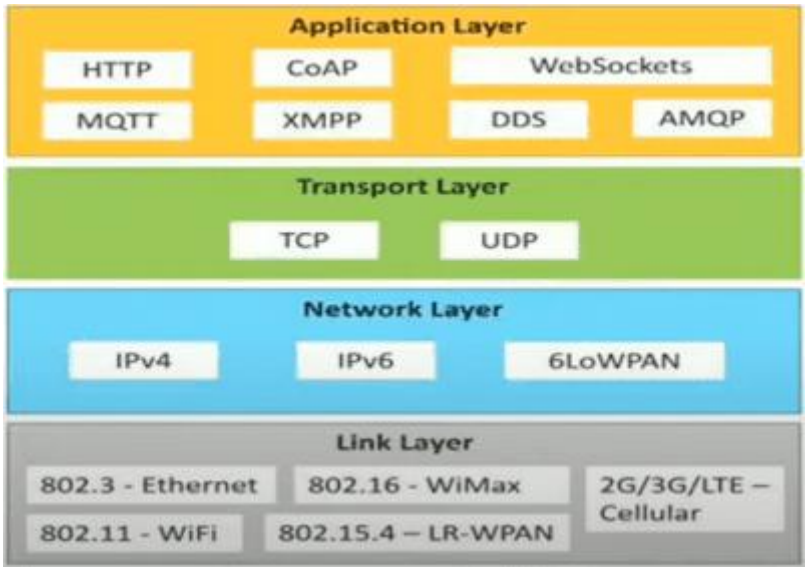
1. **IPv4 :** This is a protocol address that is a unique and numerical label assigned to each device connected to the network. an IP address performs two main functions host and location addressing. IPv4 is an IP address that is 32-bit long.

2. **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 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.

1. **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.

2. **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.



Dig. Generic Block diagram of IoT Protocols

3. 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 into the depth of describing how each component will be built with low-level programming specifics.

IoT logical design includes:

1. **IoT functional blocks**
2. **IoT communications models**
3. **IoT communication APIs**

1. 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 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, and provide an interface for controlling and monitoring various terms.

The **Functional blocks** are:

Device: An IoT system comprises of devices that provide sensing, actuation, monitoring, and control functions.

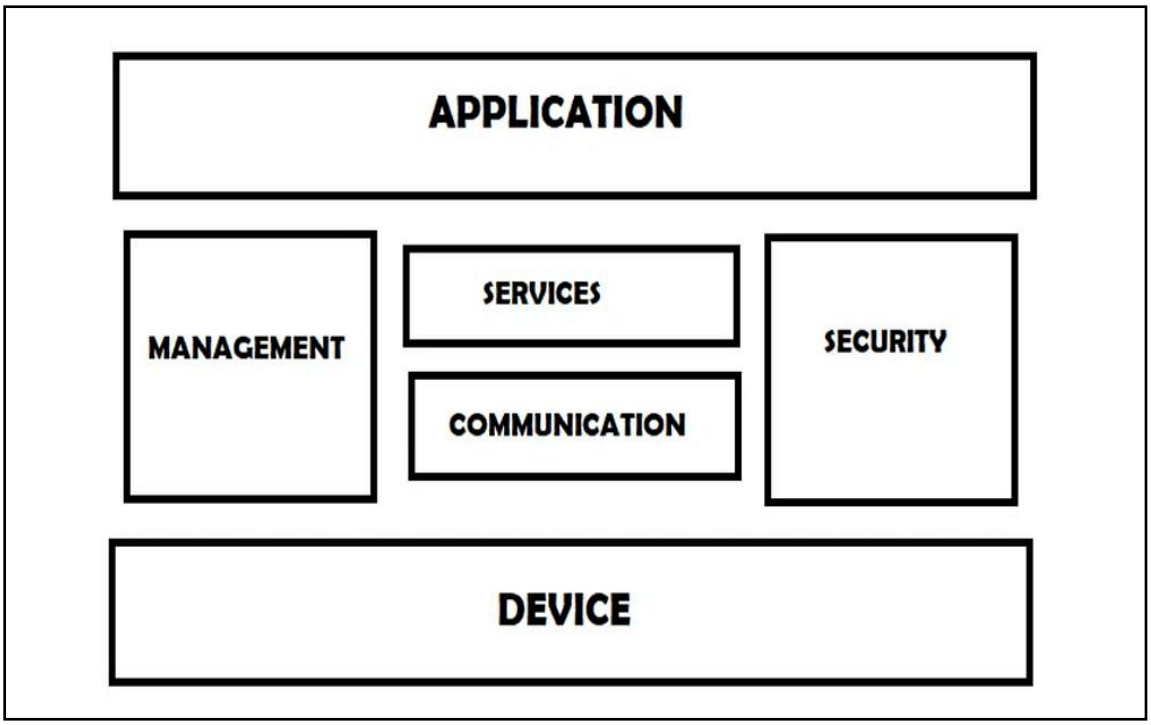
Communication: Handles the communication for the IoT system.

Services: services for device monitoring, device control service, data publishing services, and services for device discovery.

Management: this block provides various functions to govern the IoT system.

Security: This block secures the IoT system and by providing functions such as authentication, authorization, message and content integrity, and data security.

Application: This is an interface that the users can use to control and monitor various aspects of the IoT system. The application also allows users to view the system status and view or analyze the processed data.



Dig. IoT functional Blocks

2. IoT Communication Models

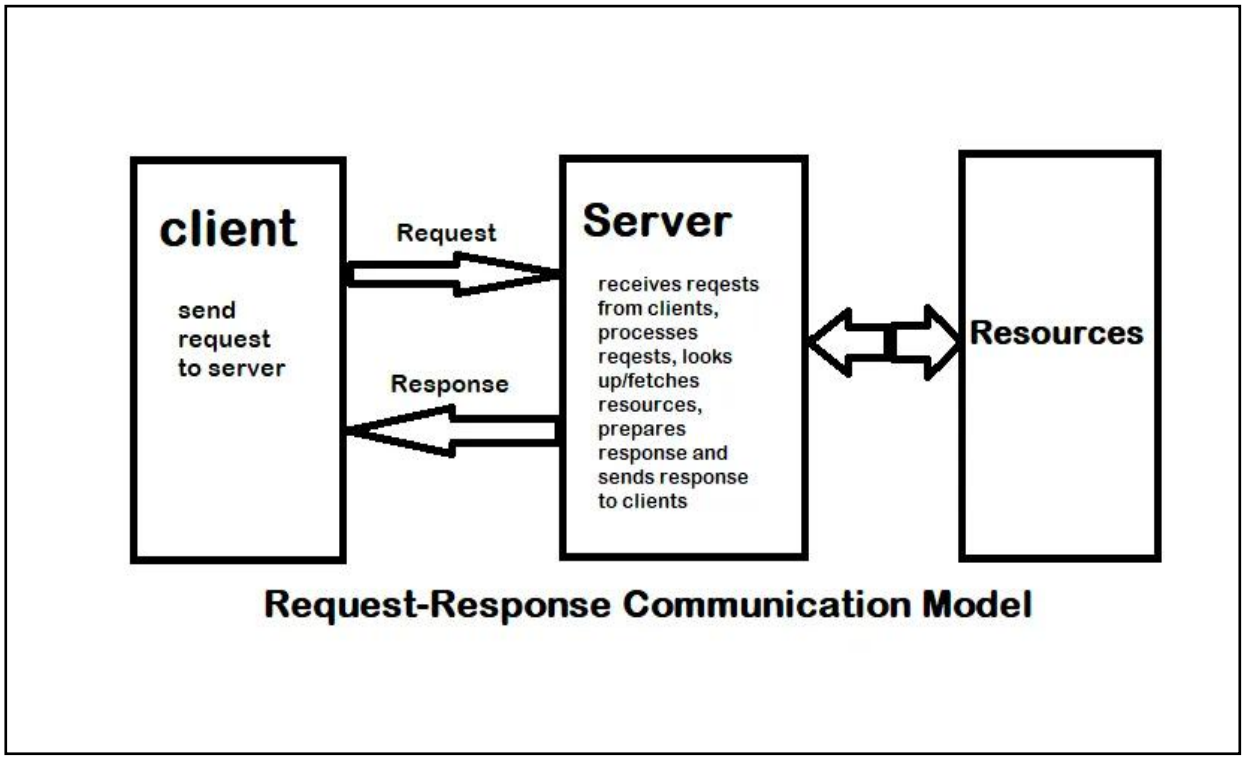
There are multiple kinds of models available in an Internet of Things system that is used for communicating between the system and server, such as:

- **Request-Response Model**

Request-response model is a communication model in which the client sends requests to the server and the server responds to the requests. When the server receives a request, it decides how to respond, fetches the data, retrieves resource representation, prepares the response, and then sends the response to the client. Request-response is a stateless communication model and each request-response pair is independent of the others.

HTTP works as a request-response protocol between a client and a server. A web browser may be the client, and an application on a computer that hosts a website may be the server.

Example: A client (browser) submits an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content.



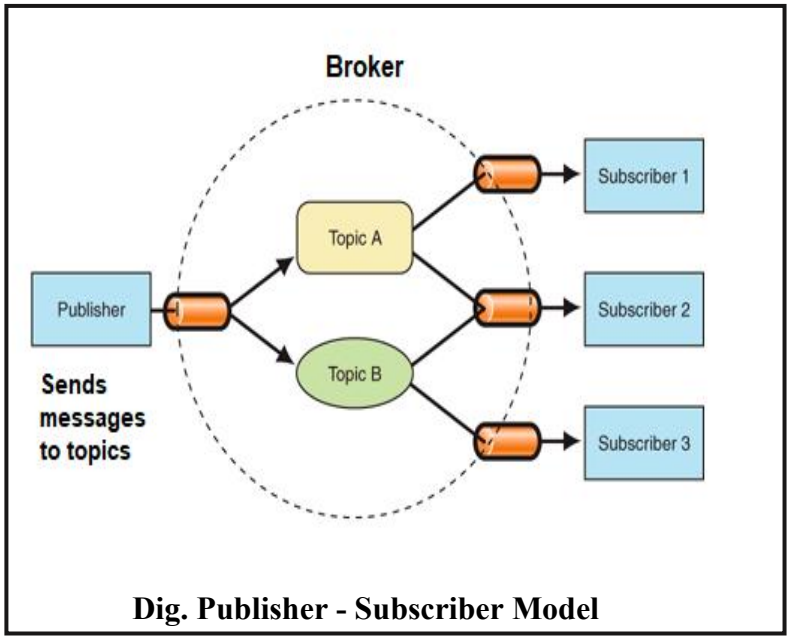
Request-Response Communication Model

- **Publisher-Subscriber Model** — This model comprises three entities: Publishers, Brokers, and Consumers.

Publishers are the source of data. It sends the data to the topic which is managed by the broker. They are not aware of consumers.

Consumers subscribe to the topics which are managed by the broker.

Brokers' responsibility is to 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 which the publisher is unaware.



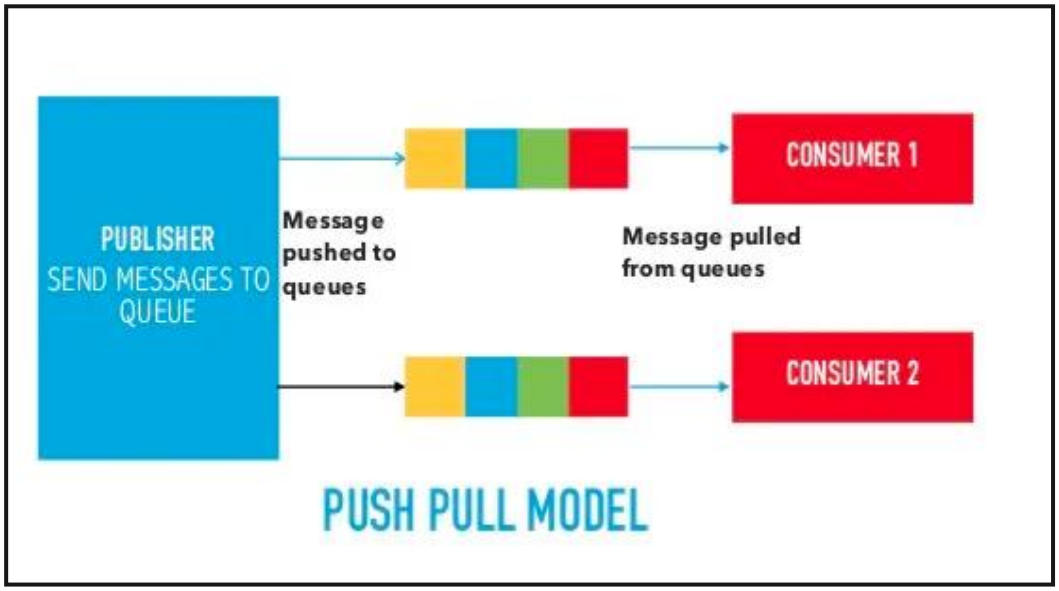
Dig. Publisher - Subscriber Model

- **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. Thus, the queue acts as the buffer for the message when the difference occurs in the rate of push or pull of data on the side of a publisher and consumer.

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.



- **Exclusive Pair –**

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.

- 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.

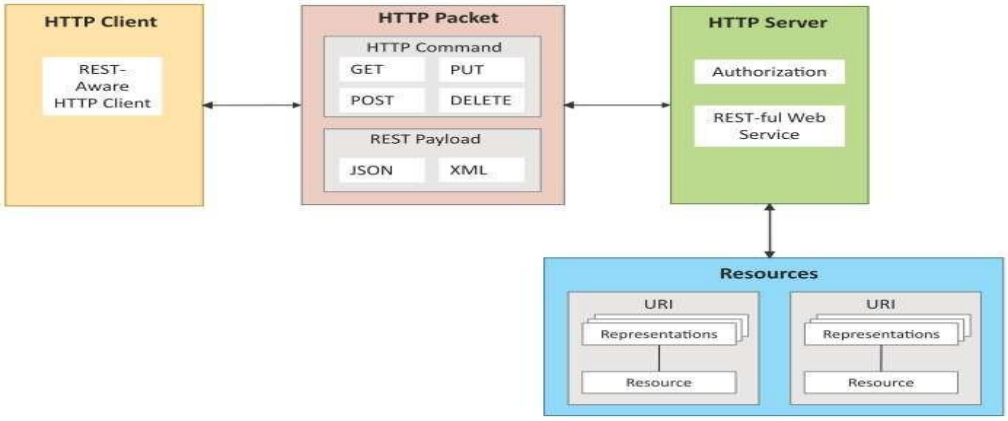
3. IoT communication API

In **IoT**, there are 2 communication APIs –

- **REST** — based Communication APIs
- **Web Socket** — based Communication APIs

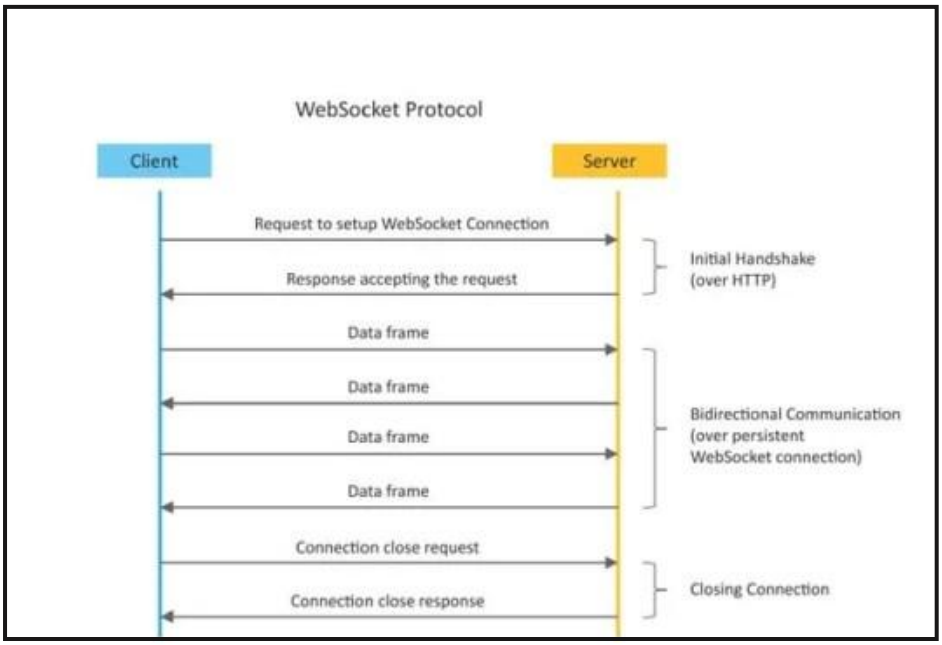
Web service can either be implemented using REST principles or using Web Socket Protocol –

1. **REST-Based Communication API:** 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. REST APIs follow the request-response communication model. The REST architectural constraints apply to the components, connectors, and data elements, within a distributed hypermedia system.



Dig. Rest-Based Communication API

2. **Web Socket-Based Communication APIs:** Web Socket APIs allow bi-directional, full-duplex communication between clients and servers. It follows the exclusive pair communication model. This Communication API does not require a new connection to be set up for each message to be sent between clients and servers. Once the connection is set up the messages can be sent and received continuously without any interruption. WebSocket APIs are suitable for IoT Applications with low latency or high throughput requirements.



Dig. WebSocket-Based Communication API

IoT Tools:

IoT (Internet of Things) development utilizes a variety of tools for hardware, software, and cloud-based solutions. These tools encompass development boards like Arduino and Raspberry Pi, software platforms like Node-RED and various cloud services (AWS IoT, Azure IoT, Google Cloud IoT), and programming

languages like Python and JavaScript. Security tools and data analytics platforms are also crucial components of the IoT ecosystem.

Hardware Tools:

- Development Boards:

Arduino and Raspberry Pi are popular choices for prototyping and building IoT devices.
Sensors and Actuators:
These are essential for gathering data and interacting with the physical world.

- Gateways:

These devices facilitate communication between sensors and the cloud.

- SoMs and SoCs:

System-on-Modules and System-on-Chips offer compact solutions for integrating processing and connectivity.

Software Tools:

- Programming Languages: Python, JavaScript, C++, and Java are commonly used for developing IoT applications.
- IoT Platforms: These platforms provide tools for device management, data collection, and application development. Examples include AWS IoT, Azure IoT, Google Cloud IoT, ThingsBoard, and Blynk.
- Node-RED: A visual tool for building IoT applications and workflows.
- Embedded Operating Systems: Operating systems like Embedded Linux are used for resource-constrained IoT devices.
- Development Environments: IDE's (Integrated Development Environments) like Arduino IDE and PlatformIO simplify the development process.
- Data Visualization Tools: Tools like PowerBI and Grafana are used to visualize and analyze IoT data.

Cloud Platforms:

- AWS IoT: A comprehensive suite of services for connecting, managing, and securing IoT devices.
- Azure IoT: Microsoft's cloud platform for building and deploying IoT solutions.
- Google Cloud IoT: Google's platform for connecting, managing, and analyzing IoT data.
- IBM Watson IoT: A platform that leverages AI and machine learning for intelligent IoT applications.

Security Tools:

- AWS IoT Device Defender: Monitors devices for security threats and provides real-time alerts.
- Azure Sphere: Provides built-in security for IoT devices, including secure boot and device authentication.
- TLS/DTLS: Protocols for secure communication over networks and datagrams.

Other Important Tools:

- Network Analyzers: Tools like Wireshark are used to monitor and analyze network traffic.
- Power Supplies and Battery Testers: Essential for testing and optimizing the power consumption of IoT devices.
- Environmental Chambers: Used to simulate real-world conditions for testing hardware.
- Security Testing Tools: Vulnerability scanners and penetration testing tools are used to identify and fix security vulnerabilities.

Difference between Rest API and Web Socket API :

REST API	WEB SOCKET API
It is a Stateless protocol. It will not store the data.	It is Stateful protocol. It will store the data.
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.
It is a Request-response model.	It is a Full duplex model.
HTTP request contains headers like head section, and title section.	It is suitable for real-time applications. It does not have any overhead.
A new TCP connection will be set up for each HTTP request.	Only Single TCP connection.
Both horizontal and vertical scaling (we can add many resources and a number of users both horizontally and vertically).	Only vertical scaling (we can add resources only vertically).
It depends upon the HTTP methods to retrieve the data	It depends upon the IP address and port number to retrieve the data
It is slower than the web socket regarding the transmission of messages.	web socket transmits messages very fast than REST API.
It does not need memory or buffers to store the data.	It requires memory and buffers to store the data.

Difference Between Physical and Logical Design of IoT

Physical Design	Logical Design
Physical design is highly detailed.	Logical design is a high-level design and doesn't provide any detail.
Physical design is more graphical than textual; however, it can comprise both.	Logical design can be textual, graphic, or both.
A physical design focuses on specific solutions explaining how they are assembled or configured.	A logical design focuses on satisfying the design factors, including risks, requirements, constraints, and assumptions.