UNIT1: INTRODUCTION & CONCEPTS	DEFINITION OF IOT
	CHARACTERISTICS OF IOT
	PHYSICAL DESIGN OF IOT
	LOGICAL DESIGN OF IOT
	IOT PROTOCOLS
	IOT ENABLING TECHNOLOGIES
	IOT LEVELS AND DEPLOYMENT TEMPLATES

Introduction to Internet of Things

The definition of Internet of things is that it is the network in which every object or thing is provided unique identifier and data is transferred through a network without any verbal communication.

Scope of IoT is not just limited to just connecting things to the internet, but it allows these things to communicate and exchange data, process them as well as control them while executing applications.

Definition of IoT (Internet of Things):

The IoT can be defined in two ways based on
- Existing Technology
- Infrastructure

Definition of IoT based on existing technology:
IoT is a new revolution to the internet due to the advancement in sensor networks, mobile devices, wireless communication, networking and cloud technologies.

Definition of IoT based on infrastructure:
IoT is a dynamic global network infrastructure of physical and virtual objects having unique identities, which are embedded with software, sensors, actuators, electronic and

network connectivity to facilitate intelligent applications by collecting and exchanging data.

Goal of IoT:

The main goal of IoT is to configure, control and network the devices or things, to internet, which are traditionally not associated with the internet i.e thermostats, utility meters, a Bluetooth connected headset, irrigation pumps and sensors or control circuits for an electric car's engine that make energy, logistics, industrial control, retail, agriculture and many other domain smarter.

Formal Definition of IoT

A dynamic global network infrastructure with self- configuring capabilities based on standard and interoperable communication protocols, where physical and virtual "things" have identities, physical attributes, and use intelligent interfaces, and are seamlessly integrated into information network that communicate data with users and environments.

Characteristics of IoT

- Dynamic & Self-Adapting
- Self-Configuring
- Interoperable Communication Protocols
- Unique Identity
- Integrated into Information Network

Dynamic and self-adapting:

The IoT devices can dynamically adapt with sensed environment, their operating conditions, and user's context and take actions accordingly. For ex: Surveillance System.

Self-configuring:

I. IoT devices can be able to upgrade the software with minimal intervention of user, Whenever they are connected to the internet.

II. They can also setup the network i.e. a new device can be easily added to the existing Network. For ex: Whenever there will be free Wi-Fi access one device can be connected easily.

Interoperable Communication:

IoT allows different devices (different in architecture) to communicate with each other as
well as with different network. For ex: MI Phone is able to control the smart AC and smart TV of different manufacturer.

Unique identities:

I. The devices which are connected to the internet have unique identities i.e IP address through which they can be identified throughout the network.

II. The IoT devices have intelligent interfaces which allow communicating with users. It

adapts to the environmental contexts.

III. It also allows the user to query the devices, monitor their status, and control them remotely, in association with the control, configuration and management Infrastructure.

Integrated into information network:

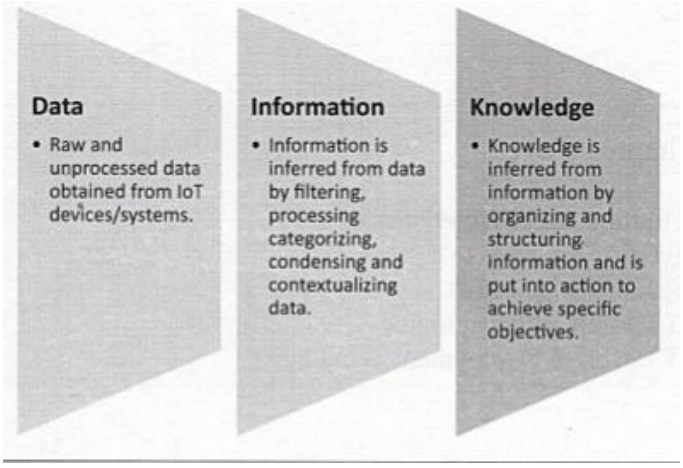
I. The IoT devices are connected to the network to share some information with other connected devices. The devices can be discovered dynamically in the network by other devices.
For ex. If a device has wifi connectivity then that will be shown to other nearby devices having wifi connectivity.

II. The devices ssid will be visible though out the network. Due to these things the network is also called as information network.

III. The IoT devices become smarter due to the collective intelligence of the individual devices in collaboration with the information network. For Ex: weather monitoring system. Here the information collected from different monitoring nodes (sensors, arduino devices) can be aggregated and analysed to predict the weather.

DIK principle (Data Information and Knowledge):

IoT is based on DIK principle. DIK refers to
• Data
• Information
• Knowledge



Data: It refers to raw and unprocessed values that are generated by the IoT devices. It does not have meaning until it is contextualised and processed into useful information.

Information: The raw data is processed, contextualised, filtered, contextualised, categorised and condensed in order to refer information. For this different algorithms and applications on IoT networks are used to extract and create information from lower level data. The raw data will be given as input to this program then they will be processed in order to get some kind of information.

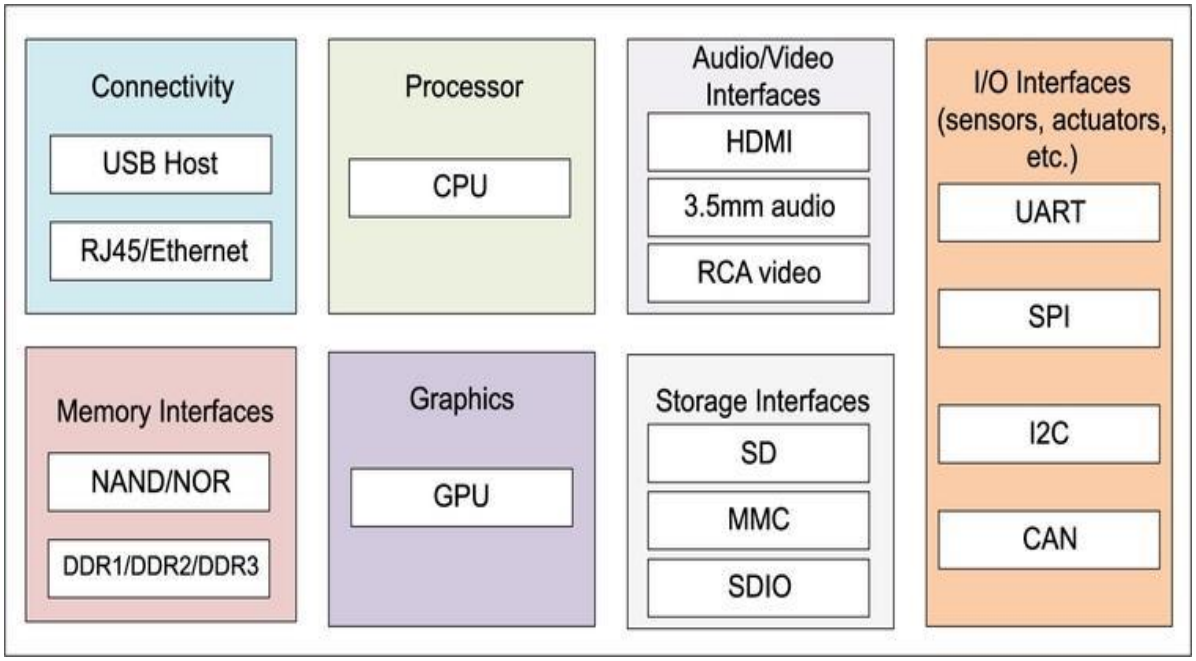
Knowledge: Knowledge can be inferred from information by organizing and structuring information and that can be put into action to achieve specific objectives.

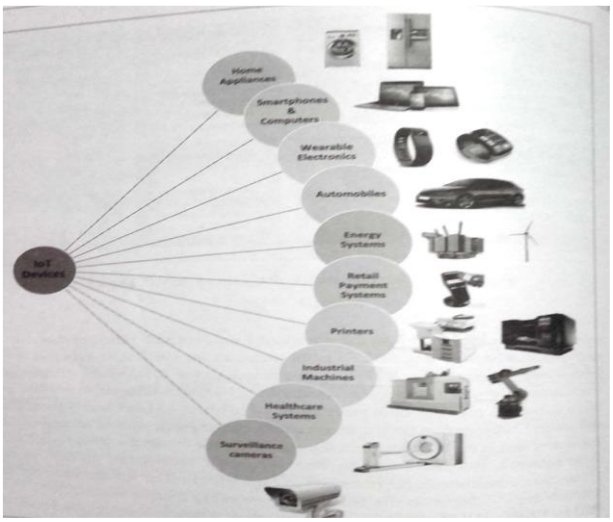
Ex: Consider a series of raw sensor measurement ((72, 45); (84, 56);) generated by a weather monitoring station. This does not have some meaning. To give meaning to the data, a context is added ex: in this example we can add that the data represents the temperature and humidity measured every minute. After adding this we get the information about the data tuple. Further information is obtained by categorising, condensing or processing the data. For ex: the average temperature and humidity reading for last five minutes is obtained by averaging the last five data tuples. The next step is to organise the information and understand the relationships between pieces of information to infer knowledge which can be put into actions. For ex: an alert is raised if the average temperature in last five minutes exceeds 120F.

Physical Design of IoT

1) Things in IoT:

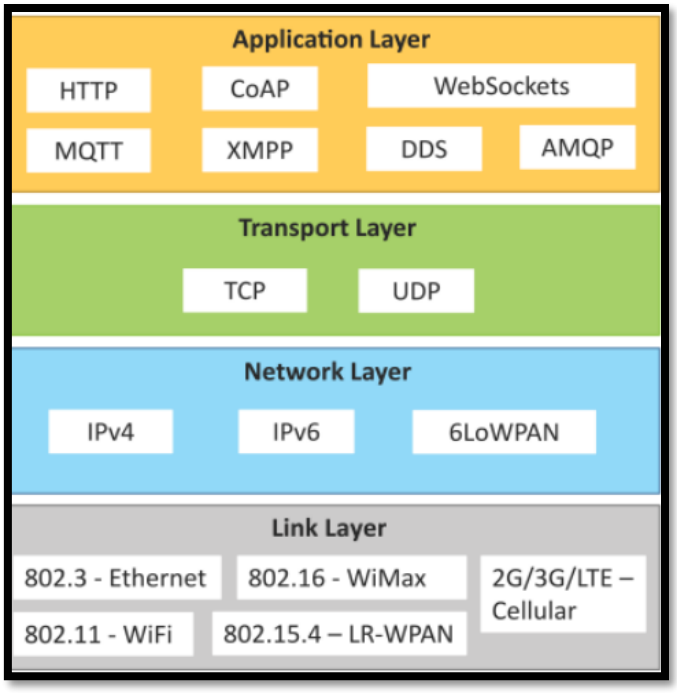
The things in IoT refers to IoT devices which have unique identities and perform remote sensing, actuating and monitoring capabilities. IoT devices can exchange data with other connected devices applications. It collects data from other devices and process data either locally or remotely.
An IoT device may consist of several interfaces for communication to other devices both wired and wireless.
These includes (i) I/O interfaces for sensors, (ii) Interfaces for internet connectivity (iii) memory and storage interfaces and (iv) audio/video interfaces.





2) IoT Protocols:

a) Link Layer : Protocols determine how data is physically sent over the network's physical layer or medium. Local network connect to which host is attached. Hosts on the same link exchange data packets over the link layer using link layer protocols. Link layer determines how packets are coded and signalled by the h/w device over the medium to which the host is attached.



Protocols:

- 802.3-Ethernet: IEEE802.3 is collection of wired Ethernet standards for the link layer. Eg: 802.3 uses co-axial cable; 802.3i uses copper twisted pair connection; 802.3j uses fiber optic connection; 802.3ae uses Ethernet over fiber.
- 802.11-WiFi: IEEE802.11 is a collection of wireless LAN(WLAN) communication standards including extensive description of link layer. Eg: 802.11a operates in 5GHz band, 802.11b and 802.11g operates in 2.4GHz band, 802.11n operates in 2.4/5GHz band, 802.11ac operates in 5GHz band, 802.11ad operates in 60 Ghzband.

- 802.16 - WiMax: IEEE802.16 is a collection of wireless broadband standards including exclusive description of link layer. WiMax provide data rates from 1.5 Mb/s to 1Gb/s.
- 802.15.4-LR-WPAN: IEEE802.15.4 is a collection of standards for low rate wireless personal area network(LR-WPAN). Basis for high level communication protocols such as ZigBee. Provides data rate from 40kb/s to250kb/s.
- 2G/3G/4G-Mobile Communication: Data rates from 9.6kb/s(2G) to up to100Mb/s(4G).

b) Network/Internet Layer: Responsible for sending IP datagrams from source n/w to destination n/w. Performs the host addressing and packet routing. Datagrams contains source and destination address.

Protocols:

- **IPv4:** Internet Protocol version4 is used to identify the devices on a n/w using a hierarchical addressing scheme. 32 bit address. Allows total of 2**32addresses.
- **IPv6:** Internet Protocol version6 uses 128 bit address scheme and allows 2**128 addresses.
- **6LOWPAN:**(IPv6overLowpower Wireless Personal Area Network)operates in 2.4 GHz frequency range and data transfer 250 kb/s.

C) Transport Layer: Provides end-to-end message transfer capability independent of the underlying n/w. Set up on connection with ACK as in TCP and without ACK as in UDP. Provides functions such as error control, segmentation, flow control and congestion control.

Protocols:

- **TCP:** Transmission Control Protocol used by web browsers(along with HTTP and HTTPS), email(along with SMTP, FTP). Connection oriented and stateless protocol. IP Protocol deals with sending packets, TCP ensures reliable transmission of protocols in order. Avoids n/w congestion and congestion collapse.

- **UDP:** User Datagram Protocol is connectionless protocol. Useful in time sensitive applications, very small data units to exchange. Transaction oriented and stateless protocol. Does not provide guaranteed delivery.

Services supported by TCP/UDP	TCP	UDP
1. Abbreviation	Transmission Control Protocol	User Datagram protocol
2. Protocol data unit	segment	datagram
3. Connection	Connection oriented	Connection less
4. Reliability	Reliable and in order as connection oriented	Unreliable and out of order as connection less.
5. Phases	3 phase i.e connection establishment, data transfer, connection release.	One phase i.e data transfer
6. Delay between data unit	Uniform delay as throughout the travel they follow same path.	Non uniform delay as throughout the travel they follow different path.
7. Congestion	Occurs during connection establishment	Occurs during data transfer.
8. Resources	Dedicated	Shared
9. Utilisation of resources	Less utilisation due to connected path and dedicated resources.	Proper utilisation due to shared resources
10. Fault tolerant technique	No fault tolerant technique	Uses fault tolerant technique
11. Message size	Prefer long message	Prefer short message
12. Flow control	Provides flow control with help of ACK (acknowledgement) field and sequence number field.	Does not provide flow control
13. Error control	It provides error control	It does not provide error control
14. Dependency on ICMP (Internet control Message Protocol)	Does not depend upon ICMP	depends upon ICMP at network layer.
15. Multi casting	Does not support	Supports multicasting
16. Broad casting	Does not support	Supports broadcasting
17. Examples of applications using TCP/ UDP	HTTP: 80 HTTPS: 443 SMTP: 25 Telnet: 23	TFTP: 69 NTP: 123 DHCP: 67,68 SNMP: 161

D) **Application Layer:** Defines how the applications interface with lower layer protocols to send data over the n/w. Enables process-to-process communication using ports.

Protocols:

- **HTTP:** Hyper Text Transfer Protocol that forms foundation of WWW. Follow request-response model Stateless protocol.
- **CoAP:** Constrained Application Protocol for machine-to-machine (M2M) applications with constrained devices, constrained environment and constrained n/w. Uses client-server architecture.
- **WebSocket:** allows full duplex communication over a single socket connection.
- **MQTT:** Message Queue Telemetry Transport is light weight messaging protocol based on publish-subscribe model. Uses client server architecture. Well suited for constrained environment.
- **XMPP:** Extensible Message and Presence Protocol for real time communication and streaming XML data between network entities. Support client-server and server-server communication.

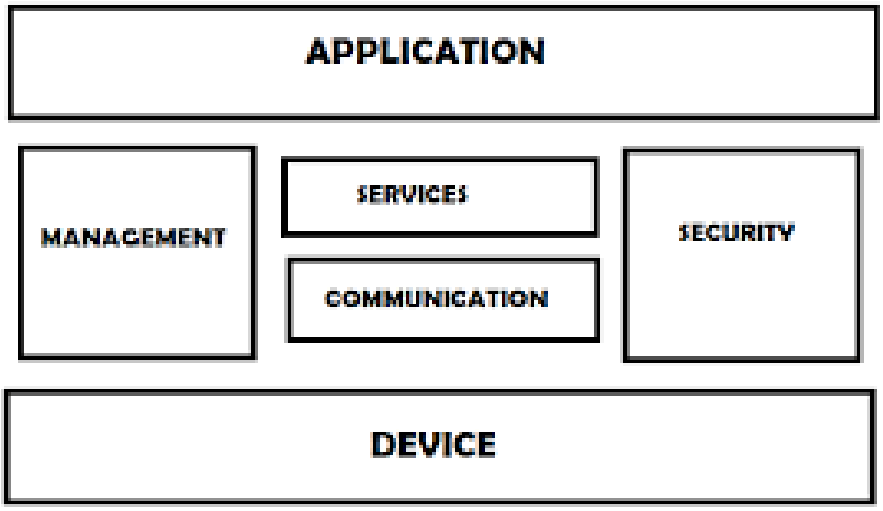
- **DDS:** Data Distribution Service is data centric middleware standards for device-to-device or machine-to-machine communication. Uses publish-subscribe model.
- **AMQP:** Advanced Message Queuing Protocol is open application layer protocol for business messaging. Supports both point-to-point and publish-subscribe model.

LOGICAL DESIGN of IoT

Refers to an abstract represent of entities and processes without going into the low level specifies of implementation.

1) IoT Functional Blocks 2) IoT Communication Models 3) IoT Comm. APIs

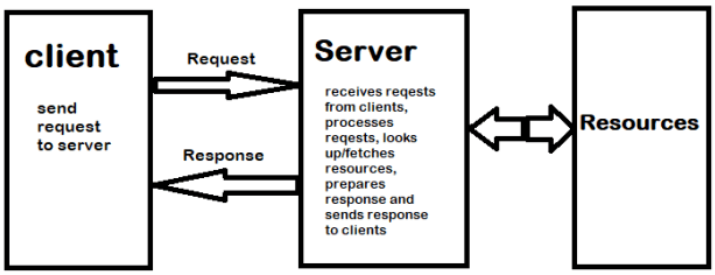
1) IoT Functional Blocks: Provide the system the capabilities for identification, sensing, actuation, communication and management.



- **Device:** An IoT system comprises of devices that provide sensing, actuation, monitoring and controlfunctions.
- **Communication:** handles the communicationfor IoTsystem.
- **Services:** for device monitoring, device control services, data publishing services and services for devicediscovery.
- **Management:** Provides various functions to govern the IoTsystem.
- **Security:** Secures IoT system and priority functions such as authentication,authorization, message and context integrity and datasecurity.
- **Application:** IoT application provide an interface that the users can use to control and monitor various aspects of IoT system.

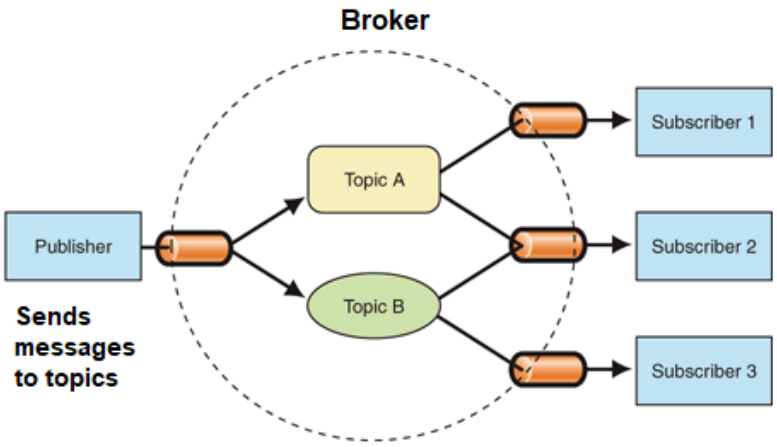
IoT Communication Models:
1) Request-Response 2) Publish-Subscibe 3)Push-Pull 4) Exclusive Pair

Request-Response Model: In which the client sends request to the server and the server replies to requests. Is a stateless communication model and each request-response pair is independent of others.



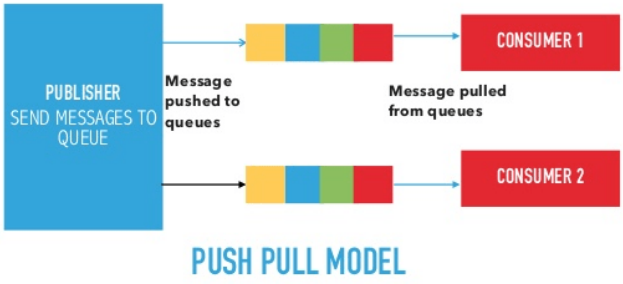
Request-Response Communication Model

2) Publish-Subscibe Model:
Involves publishers, brokers and consumers. Publishers are source of data. Publishers send data to the topics which are managed by the broker.



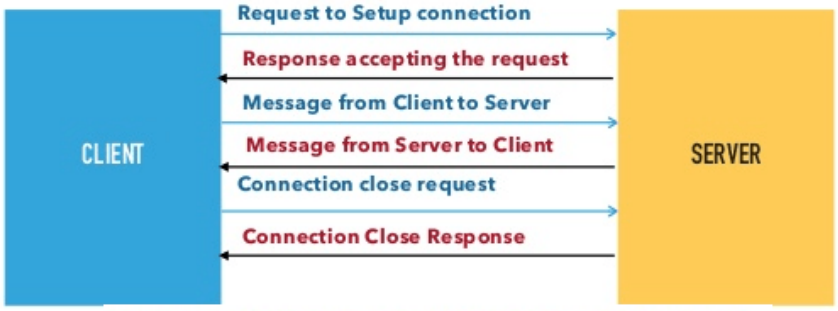
Publishers are not aware of the consumers. Consumers subscribe to the topics which are managed by the broker. When the broker receives data for a topic from the publisher, it sends the data to all the subscribed Consumers.

3) Push-Pull Model: in which data producers push data to queues and consumers pull data from the queues. Producers do not need to aware of the consumers. Queues help in decoupling the message between the producers and consumers.



PUSH PULL MODEL

4) **Exclusive Pair:** is bi-directional, fully duplex communication model that uses a persistent connection between the client and server.



EXCLUSIVE PAIR COMMUNICATION MODEL

Once connection is set up it remains open until the client send a request to close the connection. Is a stateful communication model and server is aware of all the open connections.

IoT Communication APIs

There are 2 APIs used for IoT communication

1. REST-based communication APIs
2. Websocket based communication APIs

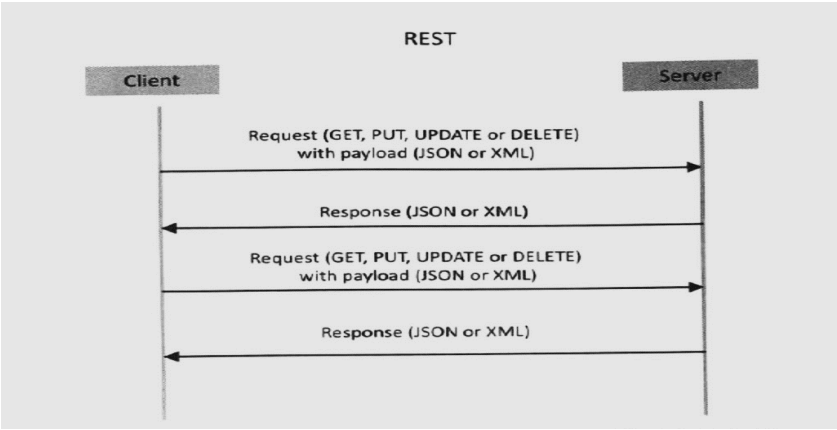
REST-based communication APIs:

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 resources and how system resource states are addressed and transferred.

REST APIs follow the request-response communication model. These are RESt architectural constraints applied to the components, connectors and data elements within a hypermedia system.

1. Client-Server: The principle behind the client-server constraints is the separation of concerns. Example- Clients should not concerned about storage. It is concern of service. Server should not concern about user interface. It is concern of clients. So this separation allows client and server to be independently developed and uploaded.
2. Stateless: Each communication should be independent of others. Each request from client should include all the information required to understand the request.
3. Cacheable: Cache constraints requires that the data within a response to a request be implicitly or explicitly labelled as cacheable or non-cacheable.
   If it is cacheable then the client is given right to reuse that response for later equivalent request.
4. Layered System: Layered system constraints the behaviour of components such that each component can't see beyond the intermediate layers during interaction.
5. Uniform Interface: Interface constraints requires that the method of communication between a client and a server must be uniform.
6. Code on demand (Optional): Servers can provide executable code or scripts for clients to execute in their context.

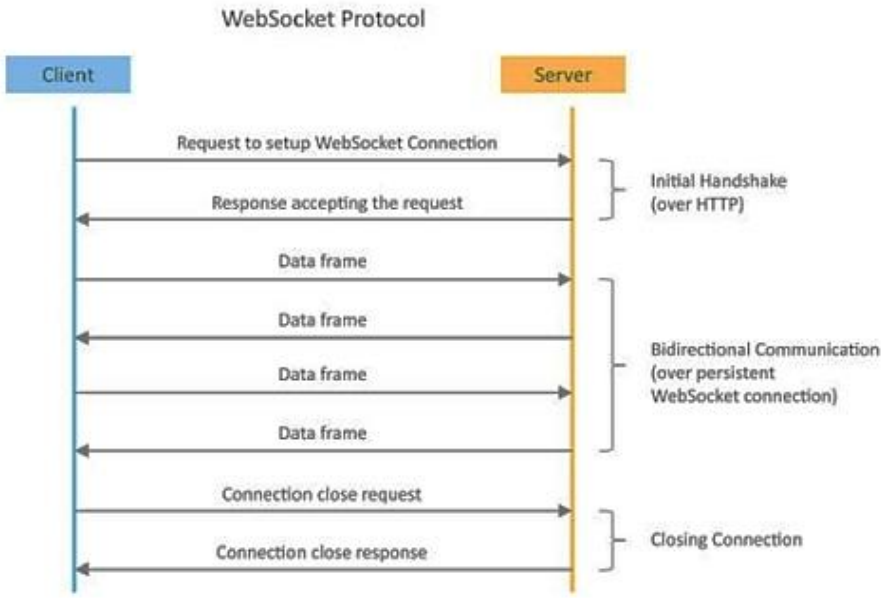
Resources are represented by URI. Client send request to those URIs using methods defined by HTTP protocol like GET, PUT, POST and DELETE.



Websocket based communication APIs

Websocket APIs allow bidirectional, full duplex communication between clients and servers. It follows Exclusive –Pair communication model. It doesn't require a new connection to be setup for each message sent.

Websocket suitable for IoT application that have low latency or high throughput requirement.



`

IoT Enabling Technologies

WSN:

- A WSN comprises of distributed devices with sensors which are used to monitor the environmental and physical conditions.
- A WSN consists of end nodes, routers and coordinator.
- End nodes have several sensors attached to them where the data is passed to coordinator with the help of routers.
- The coordinator also acts as gateway that connects WSN to Internet.

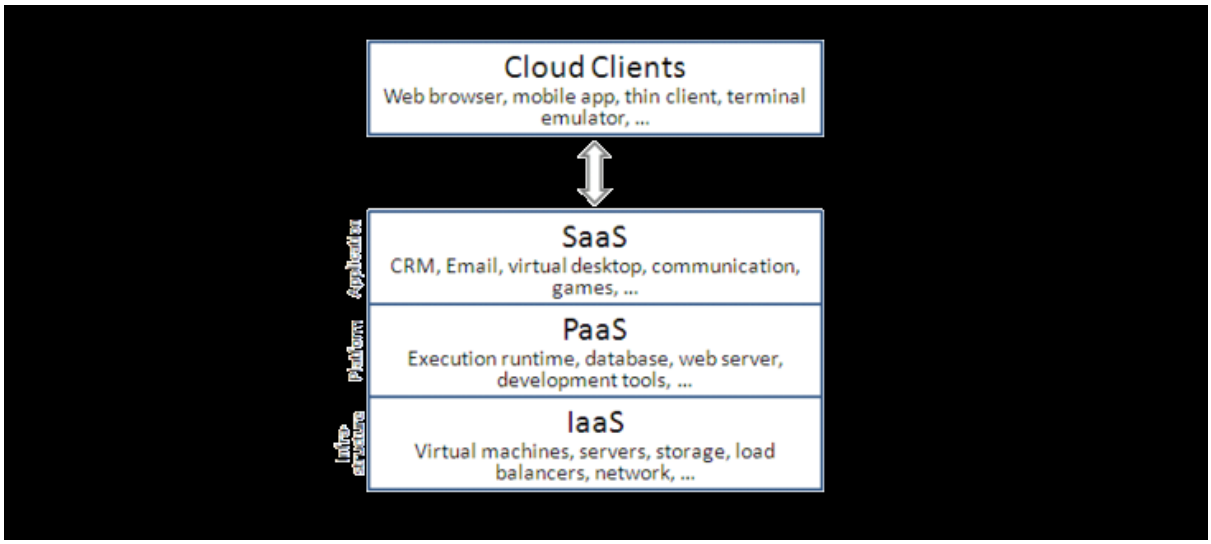
Example:

1. Soil Moisturizing System
2. Weather Monitoring system
3. Surveillance system
4. Health monitoring system.
5. Indoor Air quality monitoring system.

Cloud Computing:

Cloud computing is a trans-formative computing paradigm that involves delivering applications and services over the Internet Cloud computing involves provisioning of

computing, networking and storage resources on demand and providing these resources as metered services to the users, in a "pay as you go" model. C loud computing resources can be provisioned on demand by the users, without requiring interactions with the cloud service Provider. The process of provisioning resources is automated. Cloud computing resources can be accessed over the network using standard access mechanisms that provide platform independent access through the use of heterogeneous client platforms such as the workstations, laptops, tablets and smartphones.



Cloud computing services are offered to users in different forms:

Infrastructure as a Service (IaaS): hardware is provided by an external provider and managed for you

Platform as a Service (PaaS): in addition to hardware, your operating system layer is managed for you

Software as a Service (SaaS): further to the above, an application layer is provided and managed for you – you won't see or have to worry about the first two layers.

Big Data Analytics

Big Data analytics is the process of collecting, organizing and analyzing large sets of data (*called* Big Data) to discover patterns and other useful information. Big Data analytics can help organizations to better understand the information contained within the data and will also help identify the data that is most important to the business and future business decisions. Analysts working with Big Data typically want the *knowledge* that comes from analyzing the data.

Some examples of big data generated by IoT systems are described as follows:

- Sensor data generated by IoT system such as weather monitoring stations.
- Machine sensor data collected from sensors embedded in industrial and energy systems for monitoring their health and detecting Failures.
- Health and fitness data generated by IoT devices such as wearable fitness bands
- Data generated by ioT systems for location and tracking of vehicles
- Data generated by retail inventory monitoring systems Characteristics

Big data can be described by the following characteristics:

- **Volume –** The quantity of generated and stored data. The size of the data determines the value and potential insight, and whether it can be considered big data or not.
- **Variety –** The type and nature of the data. This helps people who analyze it to effectively use the resulting insight. Big data draws from text, images, audio, video; plus it completes missing pieces through data fusion.
- **Velocity –** In this context, the speed at which the data is generated and processed to meet the demands and challenges that lie in the path of growth and development. Big data is often available in real-time. Compared to small data, big data are produced more continually. Two kinds of velocity related to Big Data are the frequency of generation and the frequency of handling, recording, and publishing.

- **Veracity –** It is the extended definition for big data, which refers to the data quality and the data value. The data quality of captured data can vary greatly, affecting the accurate analysis.

Communication protocols

Communication protocols form the backbone of IoT systems and enable network connectivity and coupling to applications. Communication protocols allow devices to exchange data over the network. Multiple protocols often describe different aspects of a single communication. A group of protocols designed to work together are known as a protocol suite; when implemented in software they are a protocol stack.

Internet communication protocols are published by the Internet Engineering Task Force (IETF). The IEEE handles wired and wireless networking, and the International Organization for Standardization (ISO) handles other types. The ITU-T handles telecommunication protocols and formats for the public switched telephone network (PSTN). As the PSTN and Internet converge, the standards are also being driven towards convergence.

Embedded Systems

As its name suggests, Embedded means something that is attached to another thing. An embedded system can be thought of as a computer hardware system having software embedded in it. An embedded system can be an independent system or it can be a part of a large system. An embedded system is a controller programmed and controlled by a real-time operating system (RTOS) with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. Embedded systems control many devices in common use today. Ninety-eight percent of all microprocessors are manufactured to serve as embedded system component.

An embedded system has three components –

- It has hardware.
- It has application software.
- It has Real Time Operating system (RTOS) that supervises the application software and provide mechanism to let the processor run a process as per scheduling by following a plan to control the latencies. RTOS defines the way the system works. It sets the rules during the execution of application program. A small scale embedded system may not have RTOS.

IoT Levels and Deployment Templates

An IoT system comprises the following components: Device, Resource, Controller Service, Database, Web service, Analysis Component and Application.

Device: An IoT device allows identification, remote sensing, remote monitoring capabilities.

Resource: Software components on the IoT device for -accessing, processing and storing sensor information, controlling actuators connected to the device. - enabling network access for the device.

Controller Service: Controller service is a native service that runs on the device and interacts with the web services. • It sends data from the device to the web service and receives commands from the application (via web services) for controlling the device.

Database : Database can be either local or in the cloud and stores the data generated by the IoT device.

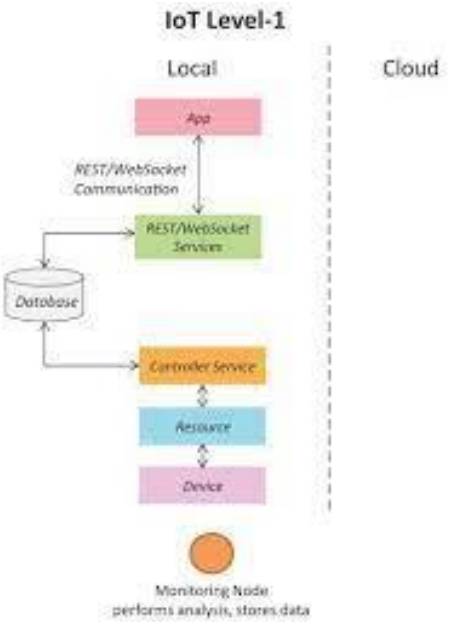
Web Service: Web services serve as a link between the IoT device, application, database and analysis components. •It can be implemented using HTTP and REST principles (REST service) or using the WebSocket protocol (WebSocket service).

Analysis Component: Analysis Component is responsible for analyzing the IoT data and generating results in a form that is easy for the user to understand.

Application: IoT applications provide an interface that the users can use to control and monitor various aspects of the IoT system. • Applications also allow users to view the system status and the processed data.

IoT Level-1

A level-1 IoT system has a single node/device that performs sensing and/or actuation, stores data, performs analysis and hosts the application.

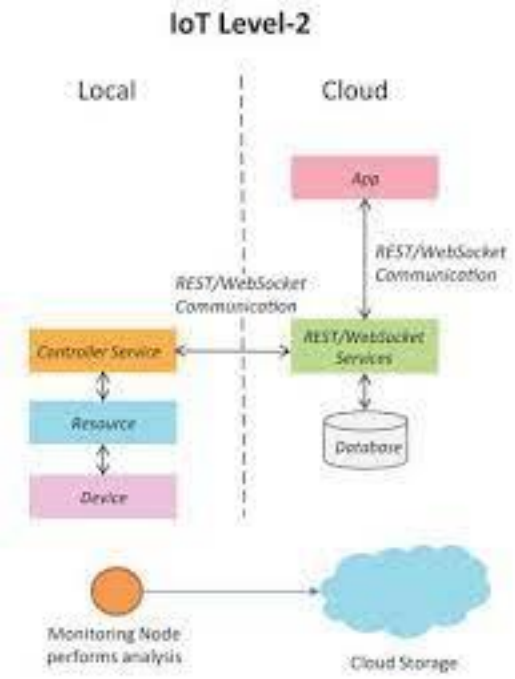


Level-1 IoT systems are suitable for modelling low- cost and low-complexity solutions where the data involved is not big and the analysis requirements are not computationally intensive.

IoT – Level 1 Example : Home Automation System

IoT Level-2

A level-2 IoT system has a single node that performs sensing and/or actuation and local analysis. Data is stored in the cloud and the application is usually cloud-based.

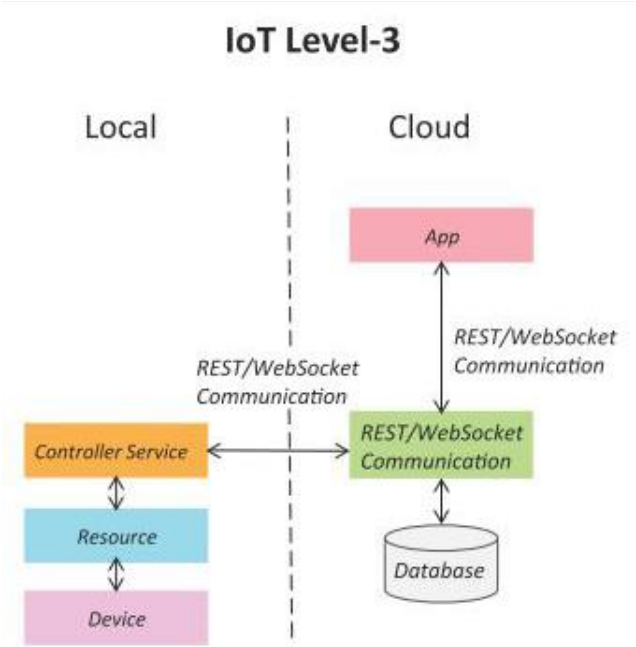


Level-2 IoT systems are suitable for solutions where the data involved is big; however, the primary analysis requirement is not computationally intensive and can be done locally.

IoT – Level 2 Example: Smart Irrigation

IoT Level-3

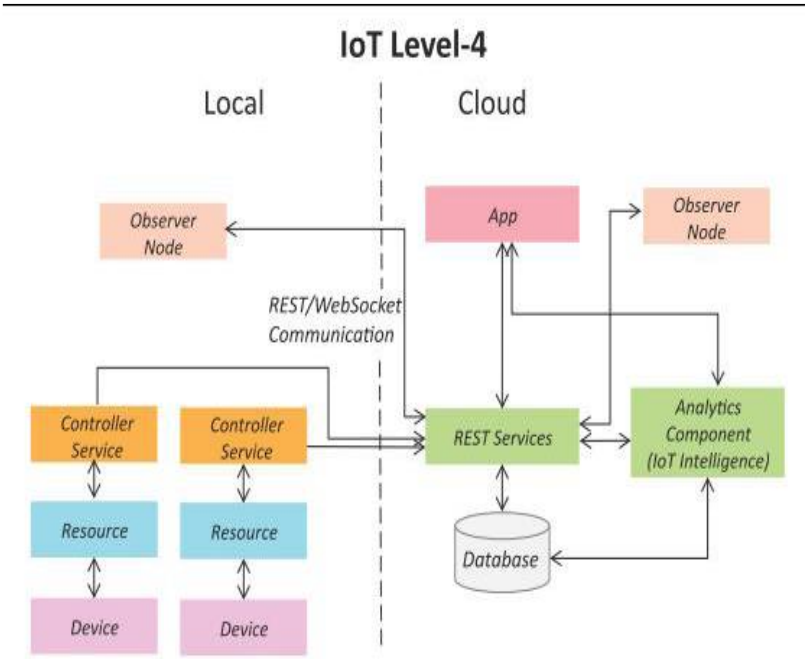
A level-3 IoT system has a single node. Data is stored and analyzed in the cloud and the application is cloud- based. Level-3 IoT systems are suitable for solutions where the data involved is big and the analysis requirements are computationally intensive.



IoT – Level 3 Example: Tracking Package Handling Sensors used Gives orientation infosense movement or vibrations Websocket service is used because sensor data can be sent in real time. Accelrometer Gyroscope

IoT Level-4

A level-4 IoT system has multiple nodes that perform local analysis. Data is stored in the cloud and the application is cloud-based. Level-4 contains local and cloud- based observer nodes which can subscribe and receive information collected in the cloud from IoT devices.



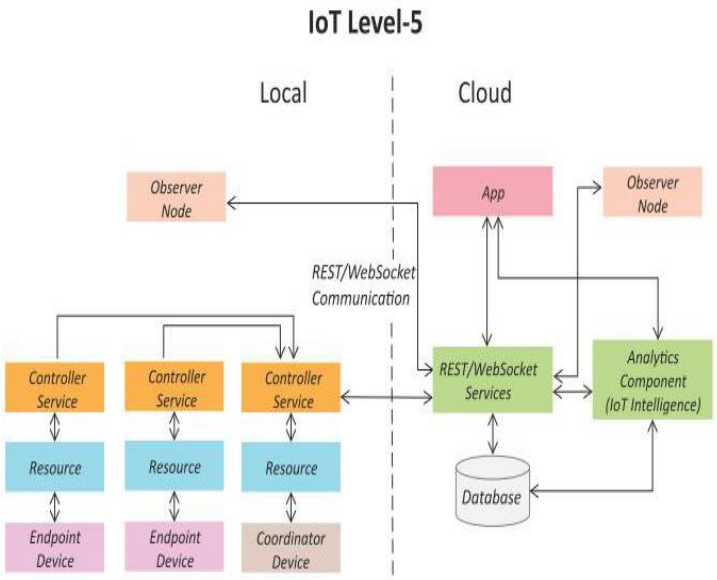
Level-4 IoT systems are suitable for solutions where multiple nodes are required, the data involved is big and the analysis requirements are computationally intensive.

IoT – Level 4 Example: Noise Monitoring Sound Sensors are used

IoT Level-5

Level-5 IoT systems are suitable for solutions based on wireless sensor networks, in which the data involved is big and the analysis requirements are computationally intensive.

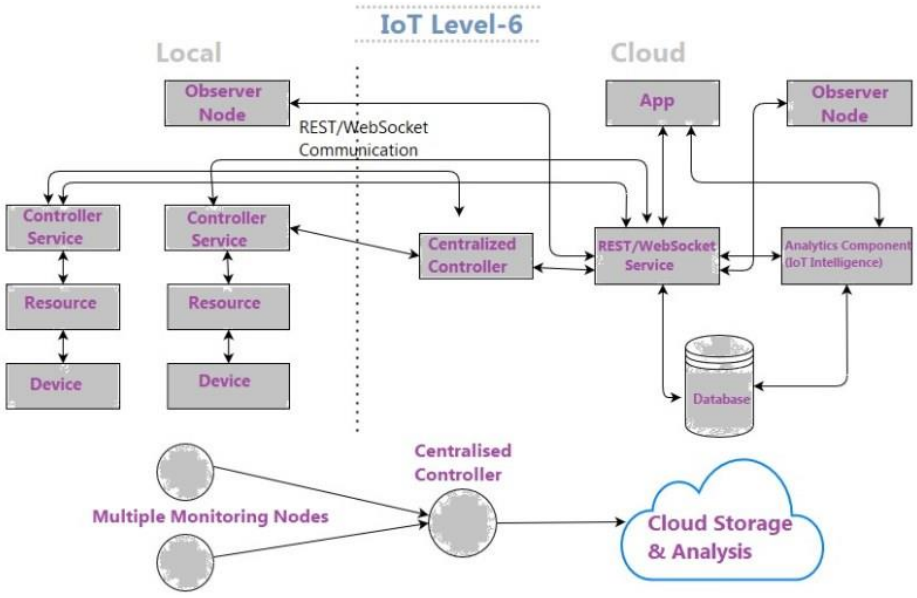
•A level-5 IoT system has multiple end nodes and one coordinator node. •The end nodes perform sensing and/or actuation. •The coordinator node collects data from the end nodes and sends it to the cloud. •Data is stored and analyzed in the cloud and the application is cloud- based.



- IoT – Level 5 Example: Forest Fire Detection Detect forest fire in early stages to take action while the fire is still controllable. Sensors measure the temperature, smoke, weather, slope of the earth, wind speed, speed of fire spread, flame length

IoT Level-6

•A level-6 IoT system has multiple independent end nodes that perform sensing and/or actuation and send data to the cloud. •Data is stored in the cloud and the application is cloud-based. •The analytics component analyses the data and stores the results in the cloud database. •The results are visualized with the cloud-based application. •The centralized controller is aware of the status of all the end nodes and sends control commands to the nodes.



IoT – Level 6 Example: Weather Monitoring System Wind speed and direction Solar radiation Temperature (air, water, soil) Relative humidity Sensors used Precipitation Snow depth Barometric pressure Soil moisture