Unit I

Introduction to Internet of things

1.1 Definition & Characteristics of IoT

1.2 Physical Design of IoT

1.3 Logical Design of IoT

1.4 IoT Enabling Technologies

1.5 IoT Levels and Deployment Templates.

1.1 Introduction

- The Internet of Things represents the whole way from collecting data, processing it, taking an action corresponding to the signification of this data to storing everything in the cloud. All this is made possible by the internet
- The Internet of things has become a very widely spread concept in the last few years. The reason for this is mainly the need to computerize and control most of the surrounding objects and have access to data in real time.
- Example: Parking sensors, about phones which can check the weather and so on

1.1.1 Definition & Characteristics of IoT

Definition:

A dynamic global n/w infrastructure with self configuring capabilities based on standard and interoperable communication protocols where physical and virtual –things‖ have identities, physical attributes and virtual personalities and use intelligent interfaces, and are seamlessly integrated into information n/w, often communicate data associated with users and their environments.

Characteristics of IoT

i) **Dynamic & Self Adapting**:

IoT devices and systems may have the capability to dynamically adapt with the changing contexts and take actions based on their operating conditions, user‘s context or sensed environment.

Eg: The surveillance system comprising of a number of surveillance cameras. The surveillance camera can adapt modes based on whether it is day or night. The surveillance system is adapting itself based on context and changing conditions.

ii) **Self Configuring:**

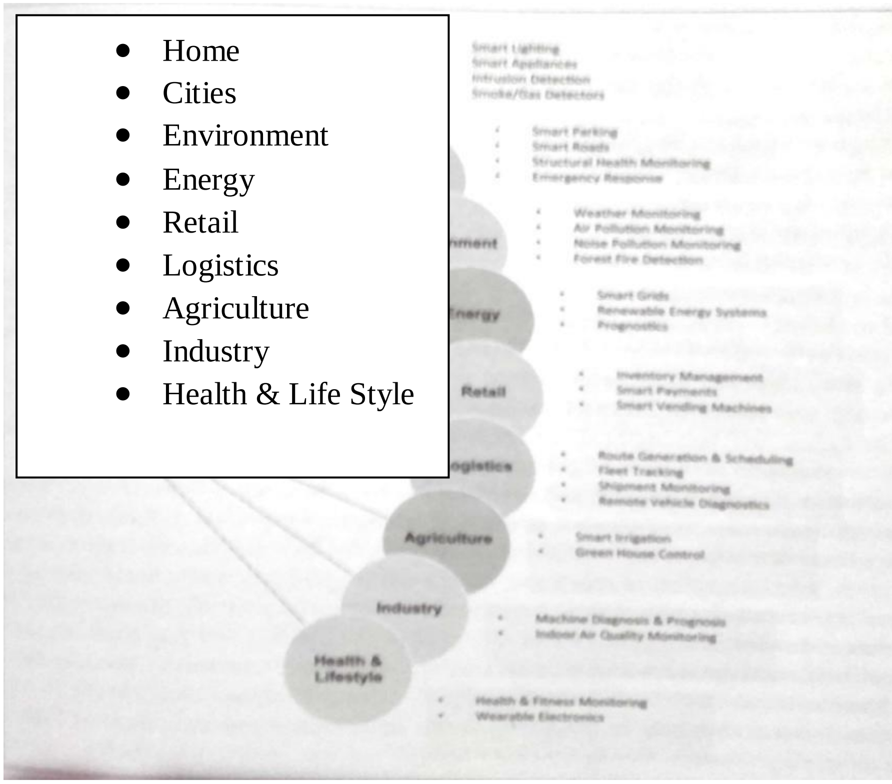
IOT devices have self configuring capability, allowing a large number of devices to work together to provide certain functionality. These devices have the ability configure themselves setup networking, and fetch latest software upgrades with minimal manual or user interaction.

iii) **Inter Operable Communication Protocols:** support a number of interoperable communication protocols and can communicate with other devices and also with infrastructure.

iv) **Unique Identity:** Each IoT device has a unique identity and a unique identifier(IP address).

v) **Integrated into Information Network:** that allow them to communicate and exchange data with other devices and systems.

Applications of IoT:



- Home
- Cities
- Environment
- Energy
- Retail
- Logistics
- Agriculture
- Industry
- Health & Life Style

1.2 **Physical Design of IoT**

1.2.1 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.

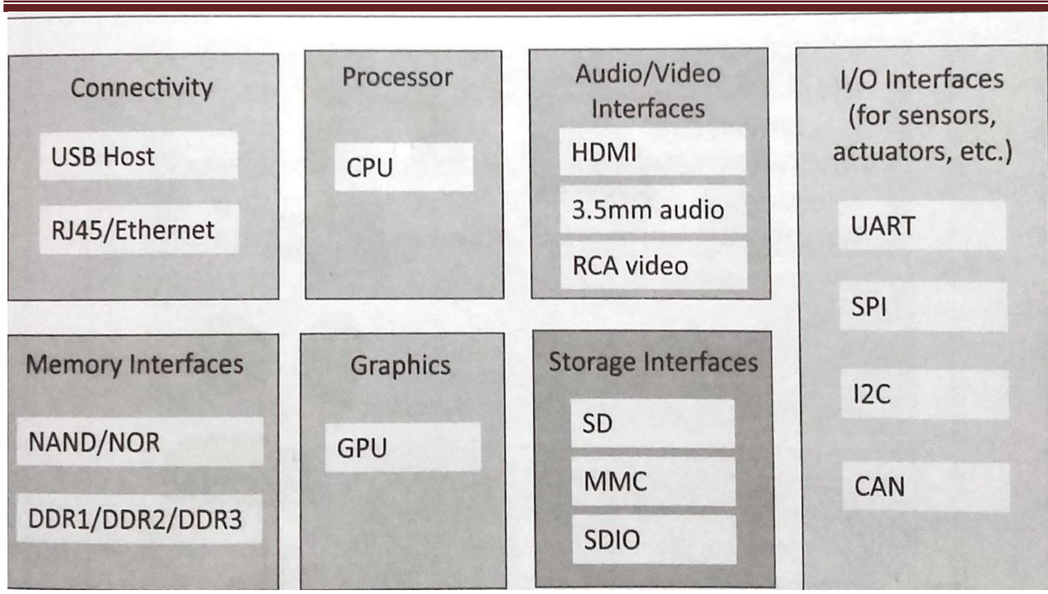
An IoT device may consist of several interfaces for communication to other devices both wired and wireless. These includes

(i) I/O interfaces for sensors,

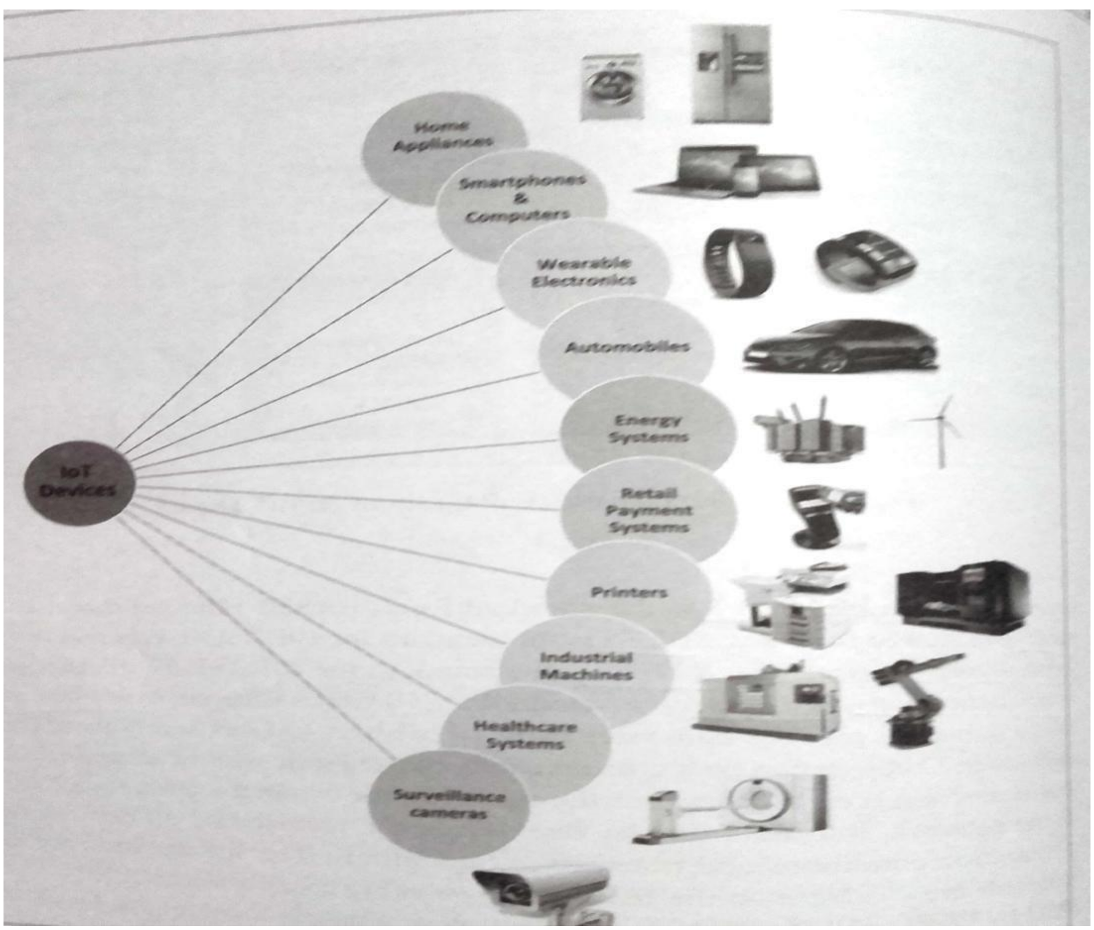
(ii) Interfaces for internet connectivity memory and storage interfaces

(iv) audio/video interfaces.

The following diagram shows the block diagram of a IOT device.



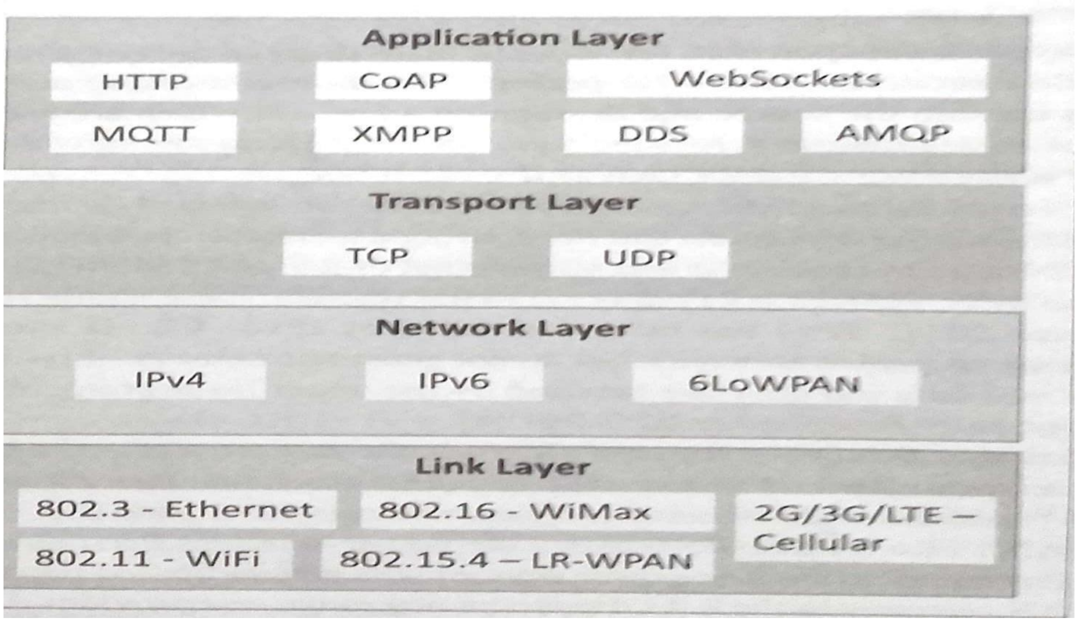
IOT devices can also be of varied types for example wearable sensors, smart watches, LED lights, automobile and industrial machines. The following diagram shows different types of IOT devices.



1.2.2 IoT Protocols.

The protocol consists of Link Layer, Network Layer, Transport Layer and Application Layer.

The following diagram shows the IoT protocol structure.



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 signaled 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 overfiber.
- **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 60Ghzband.
- **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:**(IPv6overLowpowerWirelessPersonalAreaNetwork)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.

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

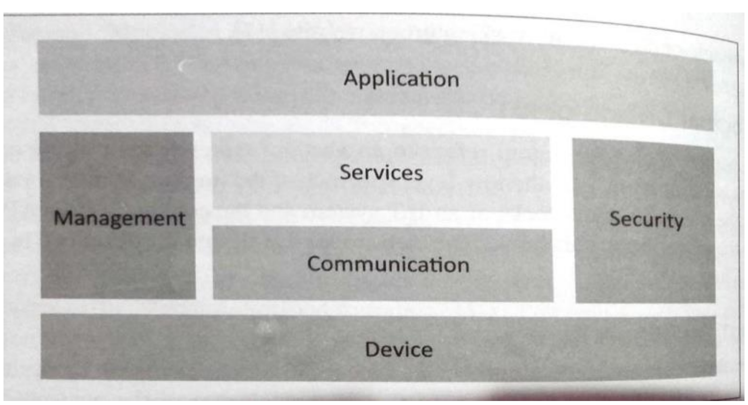
1.3Logical Design of IoT

The logical design of IoT refers to an abstract represent of entities and processes without going into the low level specifies of implementation.

i) IoT Functional Blocks ii) IoT Communication Models iii) IoT Comm. APIs

1.3.1 IoT Functional blocks

Provide the system the capabilities for identification, sensing, actuation, communication and management.

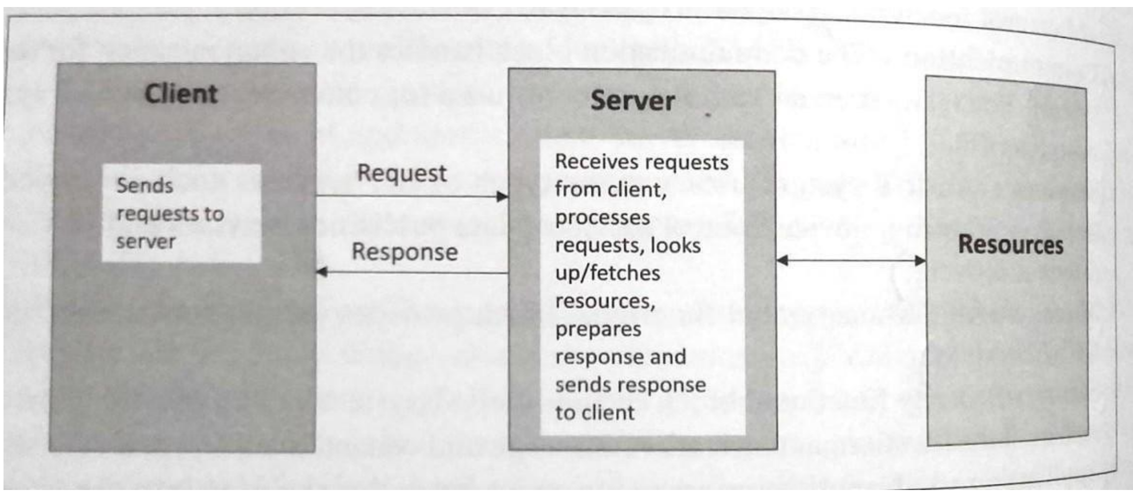


- **Device:** An IoT system comprises of devices that provide sensing, actuation, monitoring and control functions.
- **Communication:** handles the communication for IoT system.
- **Services:** for device monitoring, device control services, data publishing services and services for device discovery.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

1.3.2 IoT Communication Models

i) Request-Response ii) Publish-Subscribe iii)Push-Pull iv) Exclusive Pair

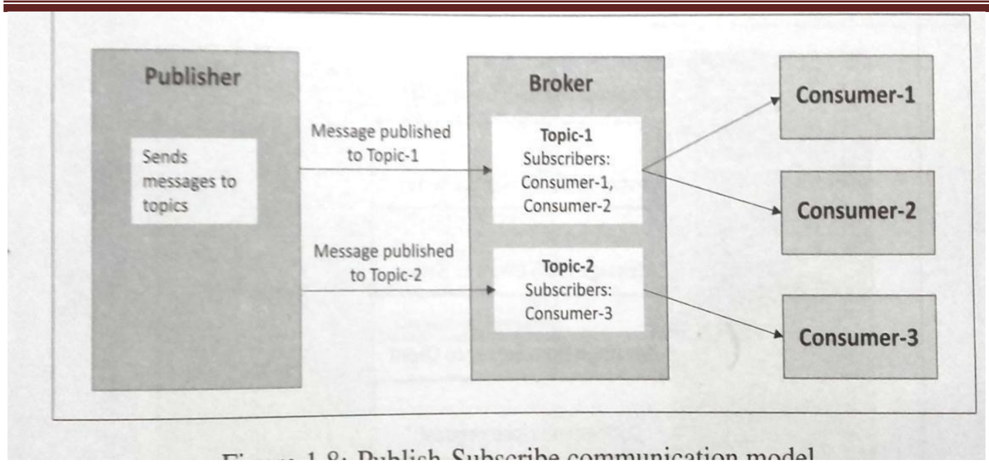
i) Request-Response Model:



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.

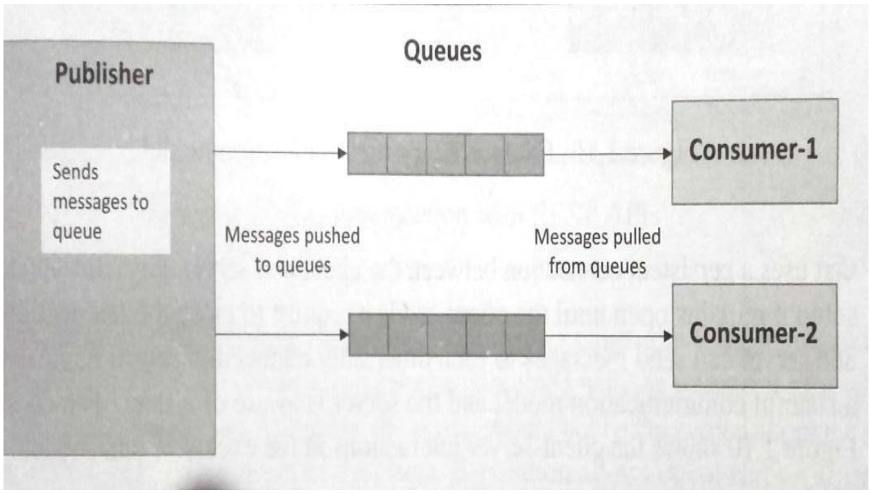
ii) Publish-Subscribe Model:

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.



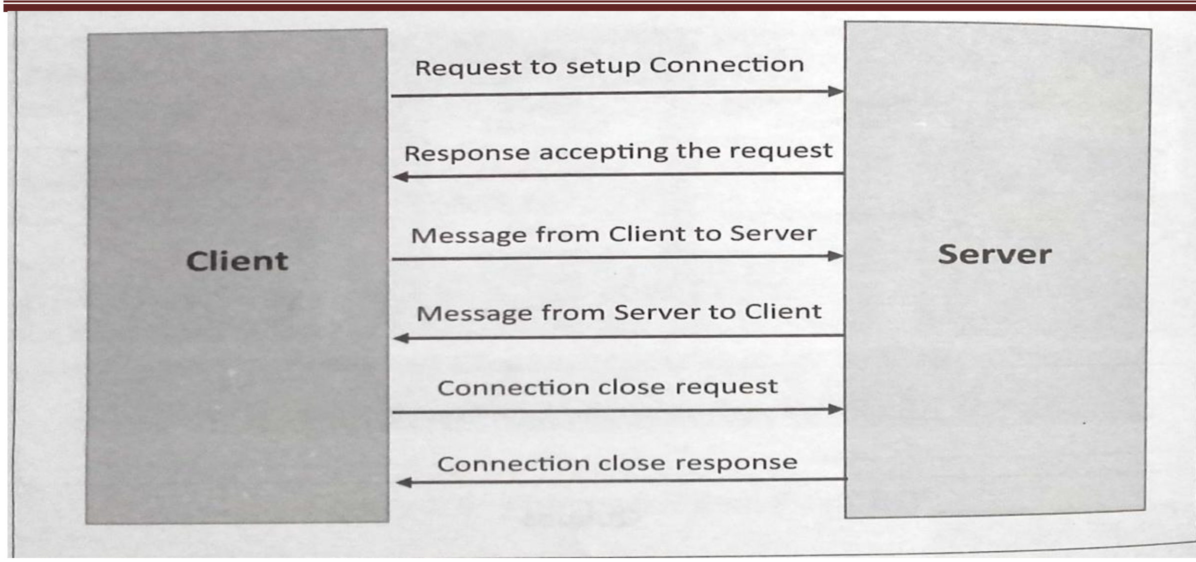
iii) Push-Pull Model:

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.



iv)Exclusive Pair:

It is bi-directional, fully duplex communication model that uses a persistent connection between the client and server. 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.



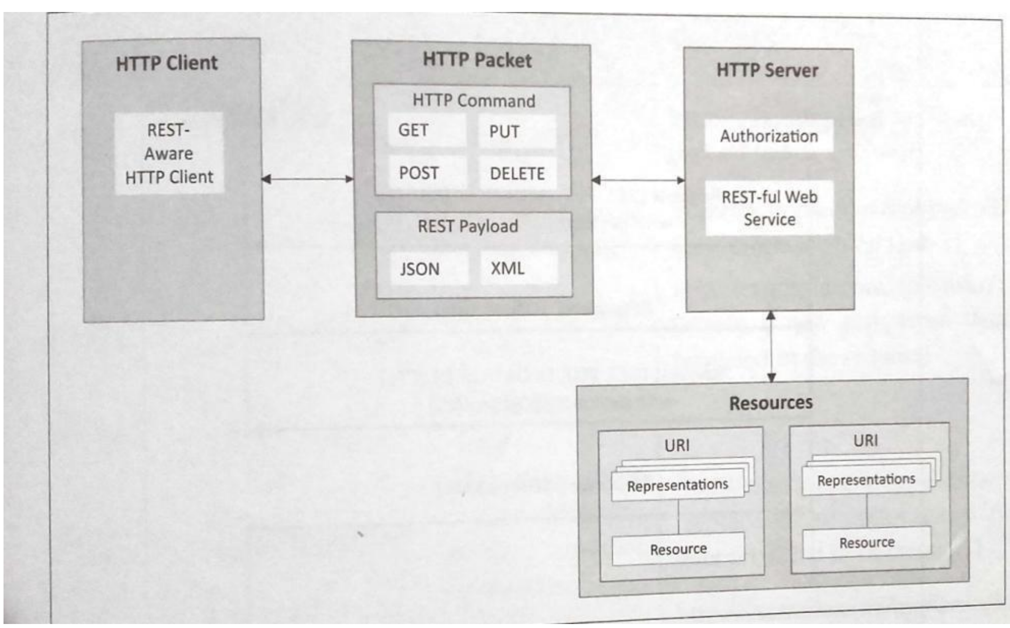
1.3.3 IoT Communication APIs.

i) REST based communication APIs(Request-Response BasedModel)

ii) WebSocket based Communication APIs(Exclusive PairBasedModel)

i) REST based communication APIs: Representational State Transfer(REST) is a set of architectural principles by which we can design web services and web APIs that focus on a system's resources and have resource states are addressed and transferred.

The REST architectural constraints are as follows: The below figure shows the communication between client server with REST APIs



Client-Server: The principle behind client-server constraint is the separation of concerns. Separation allows client and server to be independently developed and updated.

Stateless: Each request from client to server must contain all the info. Necessary to understand the request, and cannot take advantage of any stored context on the server.

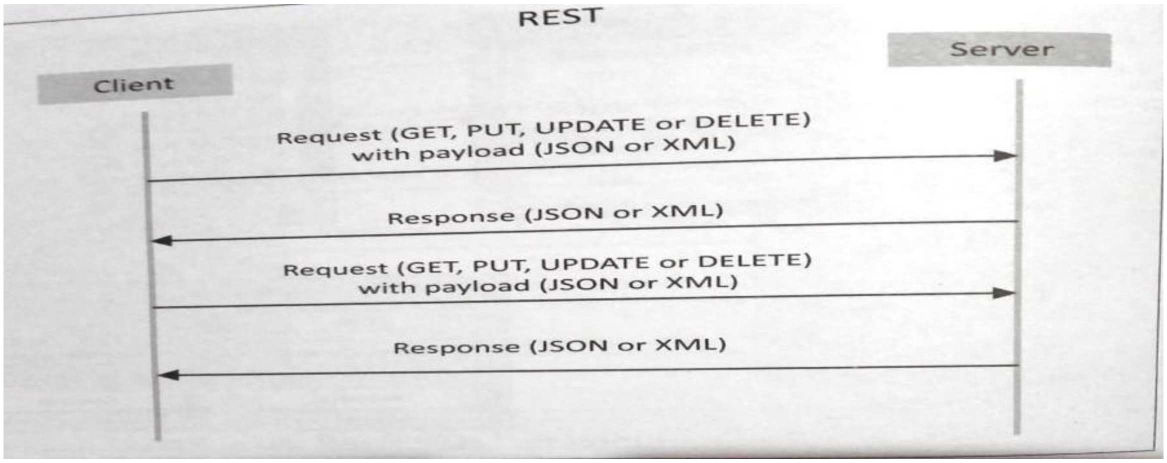
Cache-able: Cache constraint requires that the data within a response to a request be implicitly or explicitly labeled as cache-able or non-cacheable. If a response is cache-able, then a client cache is given the right to reuse that response data for later, equivalent requests.

Layered System: constraints the behavior of components such that each component cannot see beyond the immediate layer with which they are interacting.

User Interface: constraint requires that the method of communication between a client and a server must be uniform.

Code on Demand: Servers can provide executable code or scripts for clients to execute in their context. This constraint is the only one that is optional.

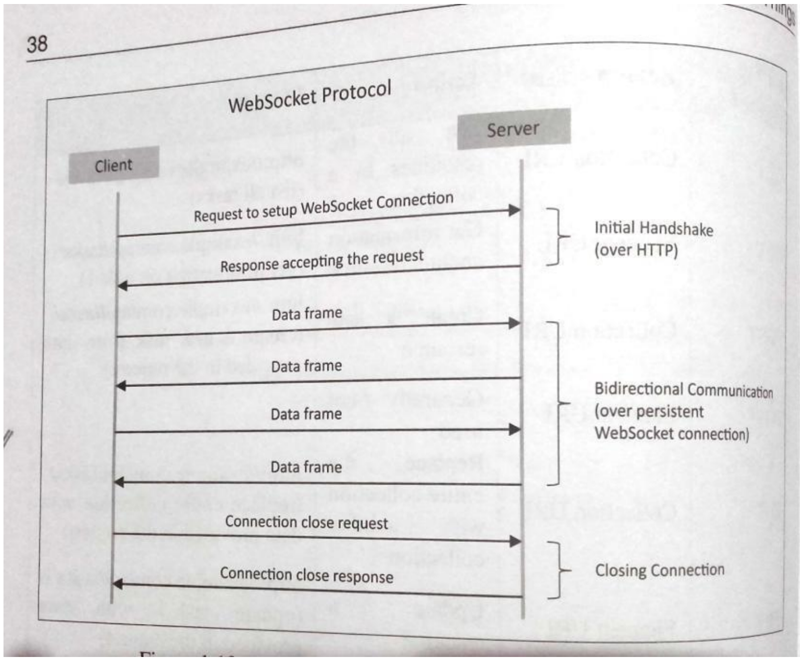
The Request-Response model used by REST:



RESTful web service is a collection of resources which are represented by URIs. RESTful web API has a base URI(e.g: http://example.com/api/tasks/). The clients and requests to these URIs using the methods defined by the HTTP protocol(e.g: GET, PUT, POST or DELETE). A RESTful web service can support various internet media types.

B) WebSocket Based Communication APIs

WebSocket APIs allow bi-directional, full duplex communication between clients and servers. WebSocket APIs follow the exclusive pair communication model.



1.4 **IoT Enabling Technologies**

IoT is enabled by several technologies including Wireless Sensor Networks, Cloud Computing, Big Data Analytics, Embedded Systems, Security Protocols and architectures, Communication Protocols, Web Services, Mobile internet and semantic search engines.

1.4.1 Wireless Sensor Networks

A wireless sensor network comprises of distributed devices with sensors which are used to monitor the environmental and physical conditions. A WSN consist of a number of end nodes and routers and a co-ordinator. The coordinator collects the data from all the nodes. Coordinator also acts as a gateway that connects the WSN to the internet.

WSNs used in IoT systems are described as follows:

- Weather Monitoring System: in which nodes collect temp, humidity and other data, which is aggregated and analyzed.
- Indoor air quality monitoring systems: to collect data on the indoor air quality and concentration of various gases.
- Soil Moisture Monitoring Systems: to monitor soil moisture at various locations.

- Surveillance Systems: use WSNs for collecting surveillance data(motion data detection).
- Smart Grids : use WSNs for monitoring grids at various points.
- Structural Health Monitoring Systems: Use WSNs to monitor the health of structures(building, bridges) by collecting vibrations from sensor nodes deployed at various points in the structure.

WSNs are enabled by wireless communication protocols such as IEEE 802.15.4. Zig Bee is one of the most popular wireless technologies used by WSNs .Zig Bee specifications are based on IEEE 802.15.4. Zig Bee operates 2.4 GHz frequency and offers data rates upto 250 KB/s and range from 10 to 100meters.

1.4.2 Cloud Computing

Cloud computing is a transformative 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”. Cloud 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 services are offered to users in different forms.

- ***<u>Infrastructure-as-a-service(IaaS):</u>***Provides users the ability to provision computing and storage resources. These resources are provided to the users as a virtual machine instances and virtual storage.
- ***<u>Platform-as-a-Service(PaaS):</u>*** Provides users the ability to develop and deploy application in cloud using the development tools, APIs, software libraries and services provided by the cloud service provider.
- ***<u>Software-as-a-Service(SaaS):</u>*** Provides the user a complete software application or the user interface to the application itself. The cloud service provider manages the underlying cloud infrastructure including servers, network, operating systems, storage, and application software.

1.4.3 Big data Analysis

Big data is defined as collections of data sets whose volume , velocity or variety is so large that it is difficult to store, manage, process and analyze the data using traditional databases and data processing tools.

Some examples of big data generated by IoT are

- Sensor data generated by IoT systems.
- Machine sensor data collected from sensors established in industrial and energy systems.
- Health and fitness data generated IoT devices.
- Data generated by IoT systems for location and tracking vehicles.
- Data generated by retail inventory monitoring systems.

The underlying characteristics of Big Data are

Volume: There is no fixed threshold for the volume of data for big data. Big data is used for massive scale data.

Velocity: Velocity is another important characteristics of Big Data and the primary reason for exponential growth of data.

Variety: Variety refers to the form of data. Big data comes in different forms such as structured or unstructured data including test data, image , audio, video and sensor data .

1.4.4 Communication Protocols:

Communication Protocols form the back-bone of IoT systems and enable network connectivity and coupling to applications.

- Allow devices to exchange data over network.
- Define the exchange formats, data encoding addressing schemes for device and routing of packets from source to destination.
- It includes sequence control, flow control and retransmission of lost packets.

1.4.5 Embedded Systems:

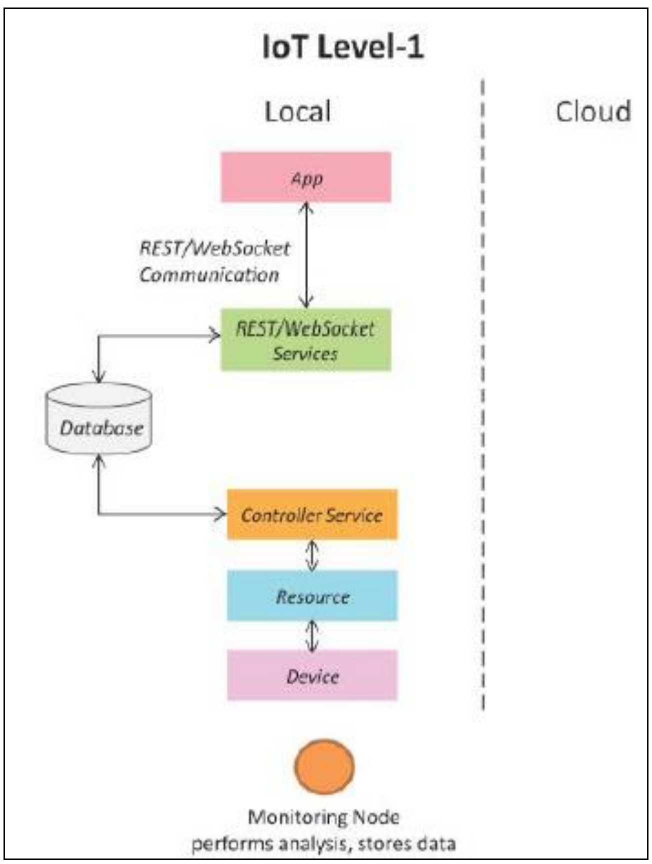
Embedded Systems is a computer system that has computer hardware and software embedded to perform specific tasks. Key components of embedded system include microprocessor or micro controller, memory (RAM, ROM, Cache), networking units (Ethernet Wi-Fi Adaptor), input/output units (Display, Keyboard, etc..,) and storage (Flash memory). Embedded System range from low cost miniaturized devices such as digital watches to devices such as digital cameras, POS terminals, vending machines, appliances etc.,

1.5 IOT Levels and Deployment Templates.

1.5.1 IoT Level-1

Level-1 IoT systems has a single node that performs sensing and/or actuation, stores data, performs analysis and host the application. Suitable for modeling low cost and low complexity solutions where the data involved is not big and analysis requirement are not computationally intensive. An e.g., of IoT Level1 is Home automation.

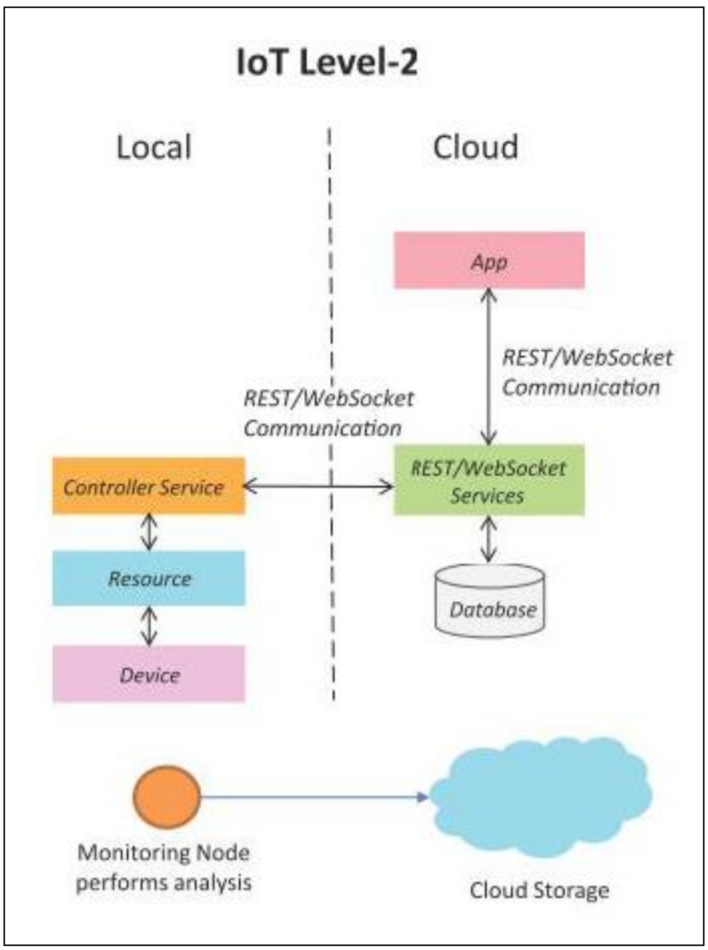
The system consist of a single node that allows controlling the lights and appliances in a home the device used in this system interfaces with the lights and appliances using electronic rely switches. The status information of each light or appliances is maintained in a local database. REST services deployed locally allow retrieving and updating the state of each lighter appliance in the status database. The controller service continuously monitors the state of each light or appliance by retrieving the light from the database.



1.5.2 IoT Level-2

IoT Level2 has a single node that performs sensing and/or actuating and local analysis as shown in fig. Data is stored in cloud and application is usually cloud based. Level2 IoT systems are suitable for solutions where data are involved is big, however, the primary analysis requirement is not computationally intensive and can be done locally itself. An e,g., of Level2 IoT system for Smart Irrigation.

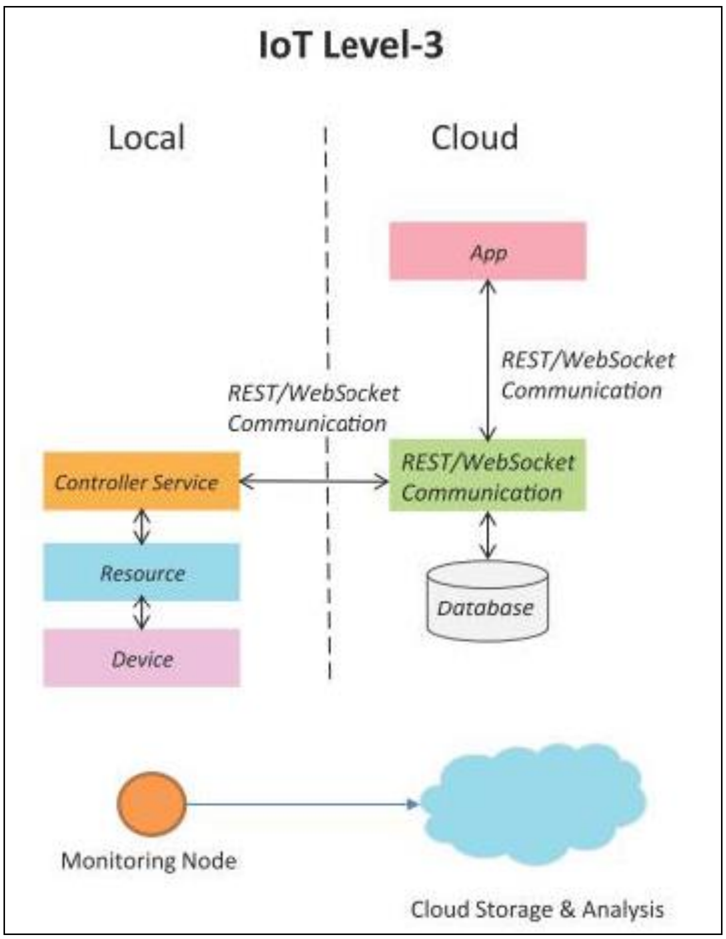
The system consists of a single node that monitors the soil moisture level and controls the irrigation system. The device used system collects soil moisture data from sensors. The controller service continuously monitors the moisture level. A cloud based REST web service is used for storing and retrieving moisture data which is stored in a cloud database. A cloud based application is used for visualizing the moisture level over a period of time which can help in making decision about irrigation schedule.



1.5.3 IoT Level -3

This System has a single node. Data is stored and analyzed in the cloud application is cloud based as shown in fig. Level3 IoT systems are suitable for solutions where the data involved is big and analysis requirements are computationally intensive.

The system consists of a single node that monitors the vibration levels for the package being shipped . The device in this system uses accelerometer and gyroscope sensor for monitoring vibration levels. The controller serves in the sensor data to the cloud in a real time using a websocket service. The data is stored in the cloud and also visualizing the cloud based applications . The analysis components in the cloud can trigger alerts if the vibration level becomes greater than the threshold.

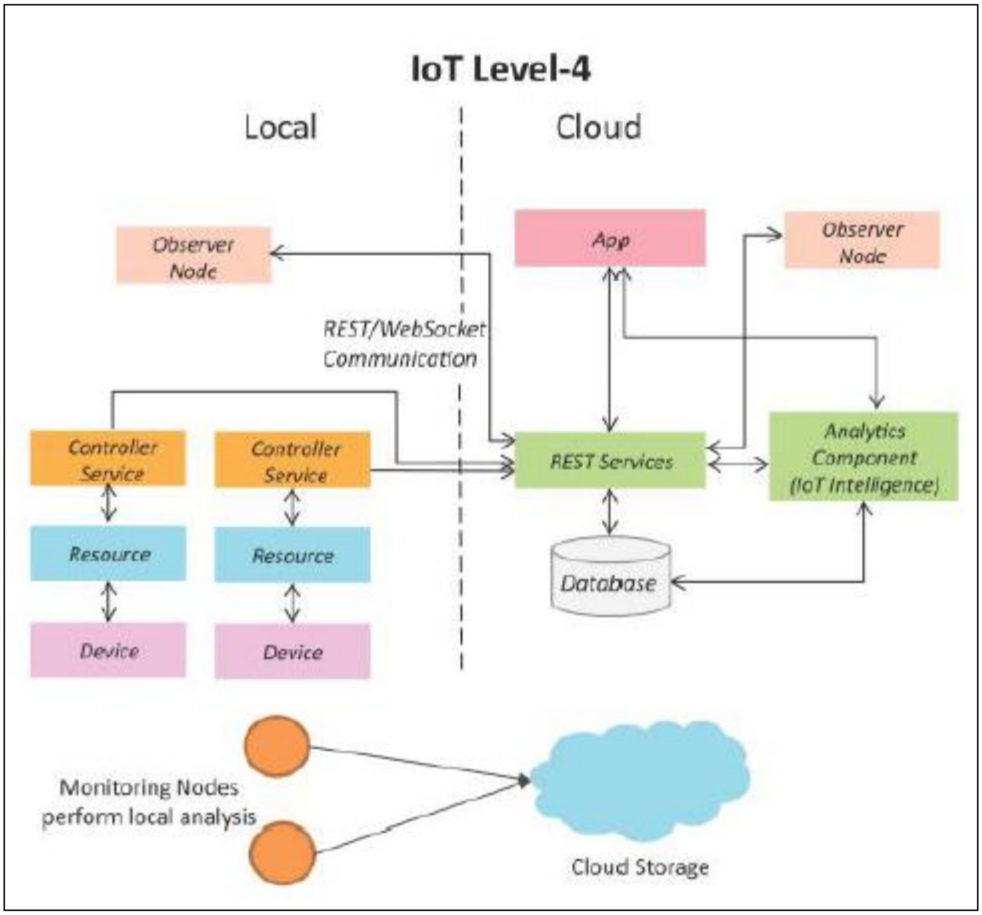


1.5.4 IoT Level-4

This System has multiple nodes that perform local analysis. Data is stored in the cloud and application is cloud based as shown in fig. Level4 contains local and cloud based observer nodes which can subscribe to 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 in big and the analysis requirements are computationally intensive.

Example : IoT System for Noise Monitoring.

The system consists of multiple nodes placed in different locations for monitoring noise levels in an area. The nodes in this example are equipped with sound sensors. Nodes are independent of each other. Each nodes runs its owner controller service that sends the data to the cloud . The data is stored in cloud database. The analysis of data collected from a number of nodes is done in the cloud. A cloud based application is used for visualizing the aggregated data.

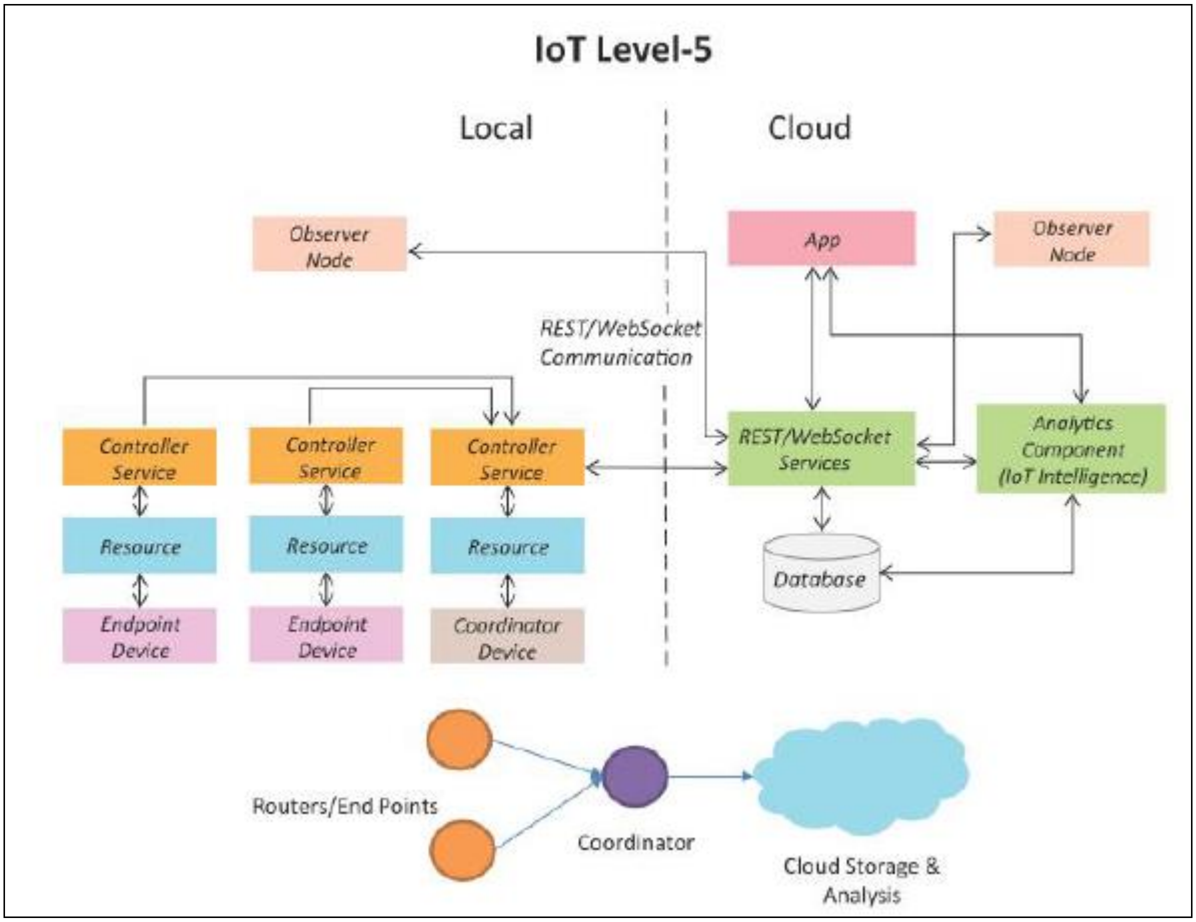


1.5.5 IoT Level-5

System has multiple end nodes and one coordinator node as shown in fig. The end nodes that perform sensing and/or actuation. Coordinator node collects data from the end nodes and sends to the cloud. Data is stored and analyzed in the cloud and application is cloud based. Level5 IoT systems are suitable for solution based on wireless sensor network, in which data are high intensive.

Example :IoT system for Forest Fire Detection.

The system consists of multiple nodes placed in different locations for monitoring temperature, humidity and CO_2 levels in a forest. The end nodes in this example are equipped with various sensors such as temperature, humidity and CO_2. The coordinator node collects the data from the end nodes and act as a gateway that provides internet connectivity to the IoT system. The controller service on the coordinator device sends the collected data to the cloud. The data is stores in a cloud database. The analysis of data is done in the computing cloud to aggregate the data and make predictions. A cloud based applications is used for visualizing the data

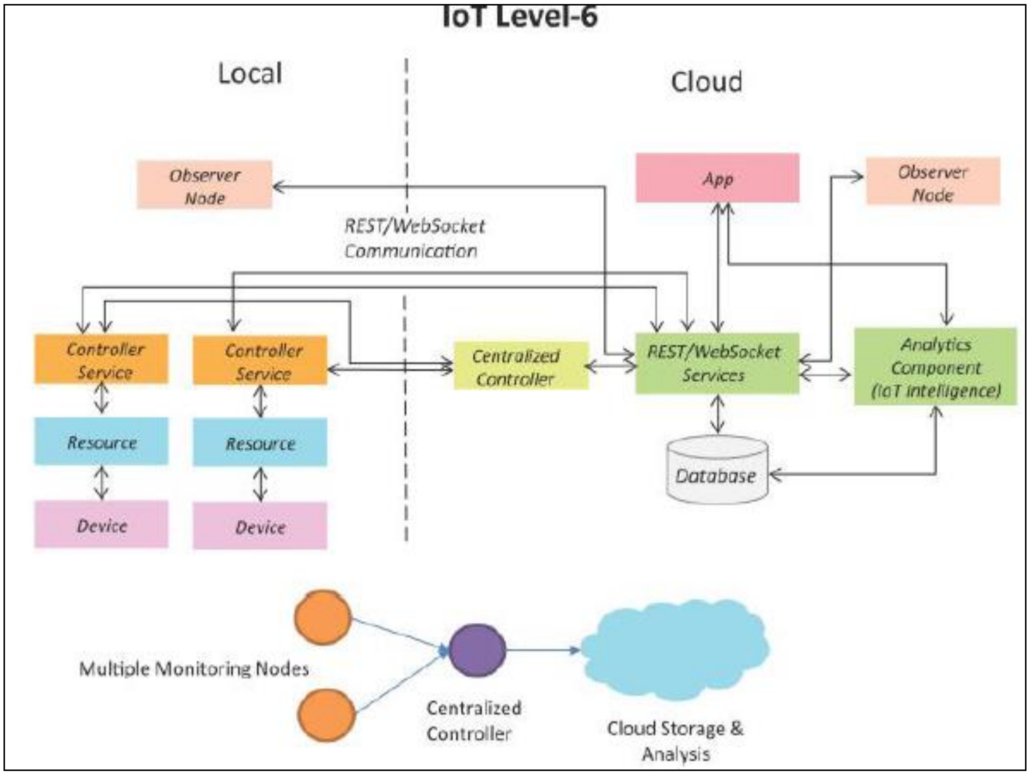


1.5.6 IoT Level -6.

System has multiple independent end nodes that perform sensing and/or actuation and sensed data to the cloud. Data is stored in the cloud and application is cloud based as shown in fig. The analytics component analyses the data and stores the result in the cloud data base. The results are visualized with the cloud based applications. The centralized controller is aware of the status of all endnodes and sends control commands to the nodes.

Example weather monitoring system

The system consists of multiple nodes placed in different locations for monitoring temperatures,humidity and pressure in an area.the end nodes are equipped with various sensors(such as temperature,humidity and pressure).the end nodes send the data to the cloud real-time using a websocket service.the data is stored in a cloud database. The analysis of data is done in a cloud to aggregate a data and make predictions.a cloud based application is used for visualizing the data.



Part A-Multiple Choice Questions

[Separately discussed]

Part B-7 Marks

1. Define IOT and explain the characteristics of IoT.
2. Make note on IoT Functional Blocks
3. Make note on IoT communication APIs.
4. Elaborate the functions of IoT Level3 with diagram
5. Explain the functions of IoT Level5 with diagram
6. Describe the functions IoT Level6 with block diagram.

Part C- 16 Marks

1. Elaborate the functions of IoT Protocols with block diagram.
2. Describe the IoT enabling technologies in detail.

2.7 IoT Physical Devices and Endpoints: IoT device

- A "Thing" in Internet of Things (IoT) can be any object that has a unique identifier and which can send/receive data (including user data) over a network (e.g., smart phone, smartTV, computer, refrigerator, car, etc.).
- IoT devices are connected to the Internet and send information about themselves or about their surroundings (e.g. information sensed by the connected sensors) over a network (to other devices or servers/storage) or allow actuation upon the physical entities/environment around them remotely.

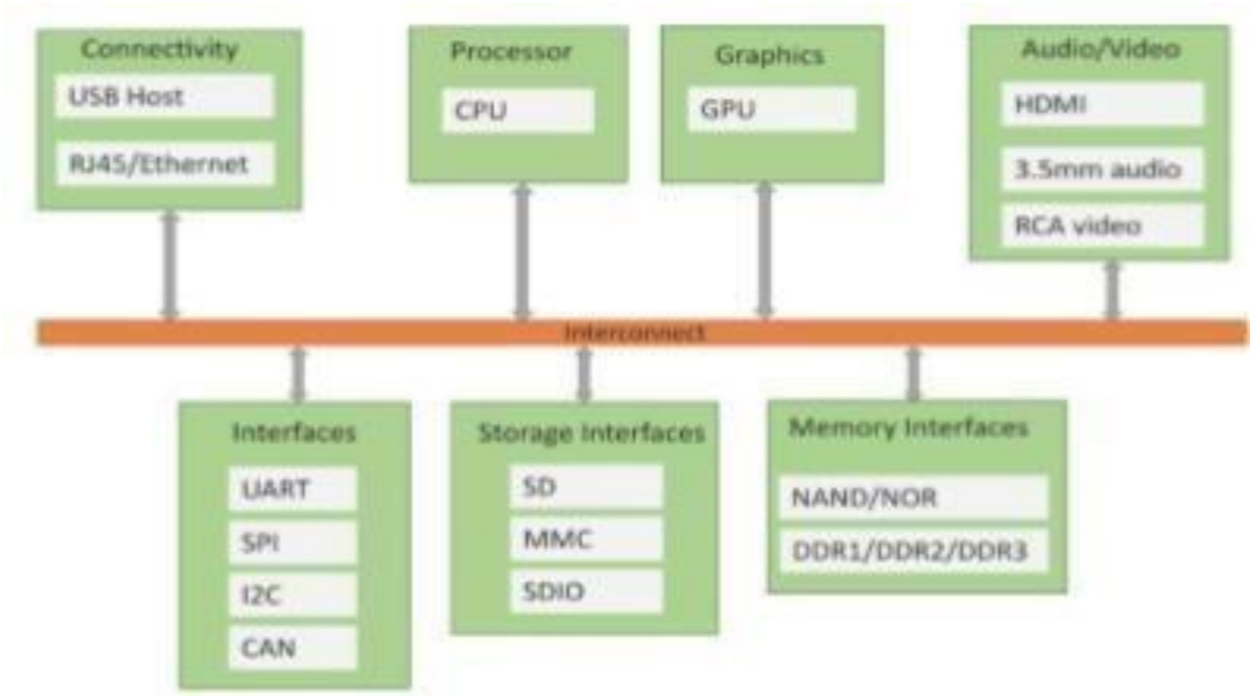
IoT Device Examples

- A home automation device that allows remotely monitoring the status of appliances and controlling the appliances.
- An industrial machine which sends information abouts its operation and health monitoring data to a server.
- A car which sends information about its location to a cloud-based service.
- A wireless-enabled wearable device that measures data about a person such as the number of steps walked and sends the data to a cloud-based service.

2.8 Basic Building Blocks of an IoT Device.

- **Sensing:** Sensors can be either on-board the IoT device or attached to the device . IoT device can collect various types of information from the on board or attached sensors such as temperature, humidity, light intensity, etc
- **Actuation:** IoT devices can have various types of actuators attached that allow taking actions upon the physical entities in the vicinity of the device.
  Example: A Relay switch connected to an IoT device can turn an appliance on/off based on the commands sent to the device.
- **Communication:** Communication modules are responsible for sending collected data to other devices or cloud-based servers/storage and receiving data from other devices and commands from remote applications.
- **Analysis & Processing:** Analysis and processing modules are responsible for making sense of the collected data

Block Diagram of an IoT Device



Expansions

- USB Host-Universal Serial Bus Host
- RJ 45/Ethernet- Component /Port
- CPU- Central Processing Unit
- GPU- Graphical Processor Unit
- HDMI-High-Definition Multimedia Interface Splitter
- RCA Video-Radio Corporation of America Community
- UART- Universal Asynchronous Receiver Transmitter
- SPI-Serial Peripheral Interface
- I2C-Inter Integrated Circuit bus
- CAN-Controller Area Network
- SD-Secondary Storage
- MMC-Multimedia Memory Cards.

- SDIO-Secure Digital Input Output
- NAND/ NOR- Logic Gates
- DDR1/DDR2/DDR3-Double Data Rate

Part A-Multiple Choice Questions

[Separately discussed]

Part B- 8 Marks

1. Elaborate the functions of IoT Level1 with diagram
2. Discuss the functions of IoT Level2 with diagram
3. Explain the functions of IoT Level5 with diagram
4. Describe the functions IoT Level6 with block diagram.

Part C- 16 Marks

1. Elaborate the functions of IoT Level3 with diagram
2. Explain the functions IoT Level4 with block diagram.
3. Describe the basic building blocks of an IoT Device with diagram.