B K Birla Institute of Engineering & Technology Pilani, Rajasthan


# Internet of Things (7CS4-01)

Vikas Kumawat

Assistant Professor
Information Technology
Department

# Introduction - IOT

RAJASTHAN TECHNICAL UNIVERSITY, KOTA

**Scheme & Syllabus**

**IV Year- VII Semester: B. Tech. (Computer Science & Engineering)**

**Teaching & Examination Scheme**

**B.Tech. : Computer Science & Engineering**

**4th Year – VII Semester**

| SN | Category | Course | | Contact hrs/week | | | Marks | | | | Cr |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Code | Title | L | T | P | Exm Hrs | IA | ETE | Total | |
| **THEORY** | | | | | | | | | | | |
| 1 | PCC | 7CS4-01 | Internet of Things | 3 | 0 | 0 | 3 | 30 | 120 | **150** | **3** |
| 2 | OE | | Open Elective - I | 3 | 0 | 0 | 3 | 30 | 120 | **150** | **3** |
| | | | **Sub Total** | **6** | **0** | **0** | **6** | **60** | **240** | **300** | **6** |
| **PRACTICAL & SESSIONAL** | | | | | | | | | | | |
| 3 | PCC | 7CS4-21 | Internet of Things Lab | 0 | 0 | 4 | 2 | 60 | 40 | **100** | **2** |
| 4 | PCC | 7CS4-22 | Cyber Security Lab | 0 | 0 | 4 | 2 | 60 | 40 | **100** | **2** |
| 6 | PSIT | 7CS7-30 | Industrial Training | 1 | 0 | 0 | | | | **125** | **2.5** |
| 7 | PSIT | 7CS7-40 | Seminar | 2 | 0 | 0 | | | | **100** | **2** |
| 8 | SODECA | 7CS8-00 | Social Outreach, Discipline &Extra Curricular Activities | | | | | | | **25** | **0.5** |
| | | | **Sub- Total** | **0** | **0** | **10** | **4** | **120** | **80** | **450** | **9** |
| | | | **TOTAL OF VII SEMESTER** | **6** | **0** | **10** | **10** | **180** | **320** | **750** | **15** |

***L:*** *Lecture,* ***T:*** *Tutorial,* ***P:*** *Practical,* ***Cr:*** *Credits*

***ETE:*** *End Term Exam,* ***IA:*** *Internal Assessment*

# Course Content

## RAJASTHAN TECHNICAL UNIVERSITY, KOTA

**Scheme & Syllabus**

**IV Year- VII Semester: B. Tech. (Computer Science & Engineering)**

**7CS4-01: Internet of Things**

**Credit: 3**
**3L+0T+0P**

**Max. Marks: 150(IA:30, ETE:120)**
**End Term Exam: 3 Hours**

| SN | Contents | Hours |
|---|---|---|
| 1 | **Introduction:** Objective, scope and outcome of the course. | 01 |
| 2 | **Introduction to IoT:** Definition and characteristics of IoT, Design of IOT: Physical design of IOT, Logical Design of IOT- Functional Blocks, communication models, communication APIs, IOT enabling Technologies- Wireless Sensor Networks, Cloud computing, big data analytics, embedded systems. IOT Levels and deployment templates. | 08 |
| 3 | **IoT Hardware and Software:** Sensor and actuator, Humidity sensors, Ultrasonic sensor, Temperature Sensor, Arduino, Raspberry Pi, LiteOS, RIoTOS, Contiki OS, Tiny OS. | 07 |
| 4 | **Architecture and Reference Model:** Introduction, Reference Model and architecture, Representational State Transfer (REST) architectural style, Uniform Resource Identifiers (URIs). Challenges in IoT- Design challenges, Development challenges, Security challenges, Other challenges. | 08 |
| 5 | **IOT and M2M:** M2M, Difference and similarities between IOT and M2M, Software defined networks, network function virtualization, difference between SDN and NFV for IoT. | 08 |
| 6 | **Case study of IoT Applications:** Domain specific IOTs- Home automation, Cities, environment, Energy, Retail, Logistics, Agriculture, Industry, Health and Lifestyles. | 08 |
| | **Total** | 40 |

## Course Objectives

- To Learn Basic IoT Design and enable technologies
- To Learn IoT hardware and software
- To Learn about architecture and reference model
- To Learn the Concepts of Machine to machine and IoT
- To Learn case studies of IoT application

## Pre-requisites

- Fundamentals of Embedded Systems
- Concepts of Raspberry Pi
- Eager to Learn New Concepts

# Course Outcome

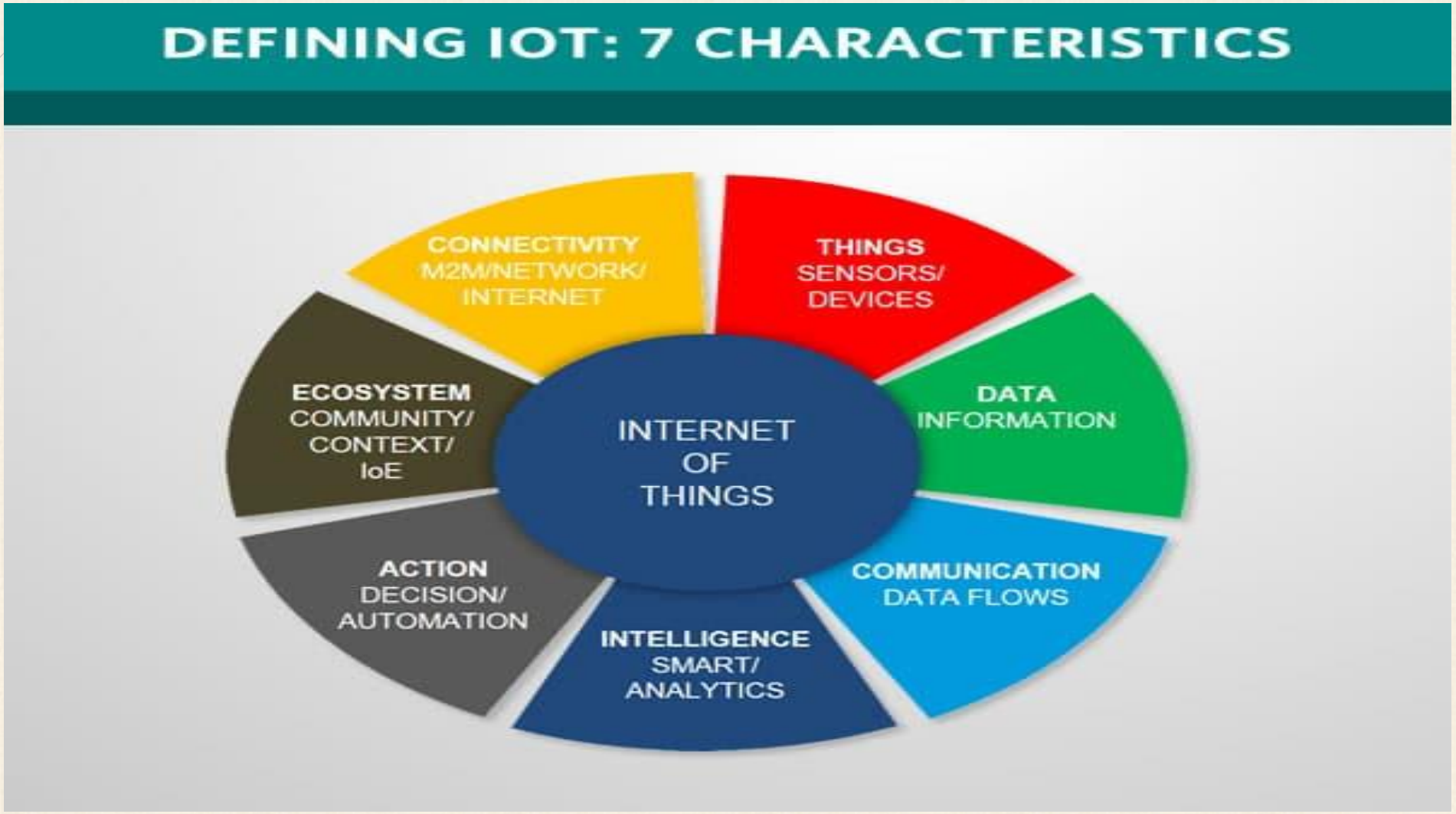
- CO1: Concepts of designing a IoT model.
- CO2: To understand about IoT hardware and software
- CO3: To understand the architecture and reference models
- CO4: To get the knowledge of Machine to machine intraction and networks
- CO5:Review on case study of IoT application.

# Definition IoT

**Definition 1**-The **Internet of things** (**IoT**) is a system of interrelated computing devices, mechanical and digital machines provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

**Definition 2-**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 virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network, often communicate data associated with users and their environments.

# Characteristics of IoT–



# Characteristics –

**Connectivity.** This doesn't need too much further explanation. With everything going on in IoT devices and hardware, with sensors and other electronics and connected hardware and control systems there needs to be a connection between various levels.

**Things**. Anything that can be tagged or connected as such as it's designed to be connected. From sensors and household appliances to tagged livestock. Devices can contain sensors or sensing materials can be attached to devices and items.

**Data**. Data is the glue of the Internet of Things, the first step towards action and intelligence.

**Communication**. Devices get connected so they can communicate data and this data can be analyzed. Communication can occur over short distances or over a long range to very long range. Examples: Wi-Fi, **LPWA** network technologies such as **LoRa** (Long Range) is a low-power wide-area network (LPWAN) or **NB-IoT** (**Narrowband IoT**).

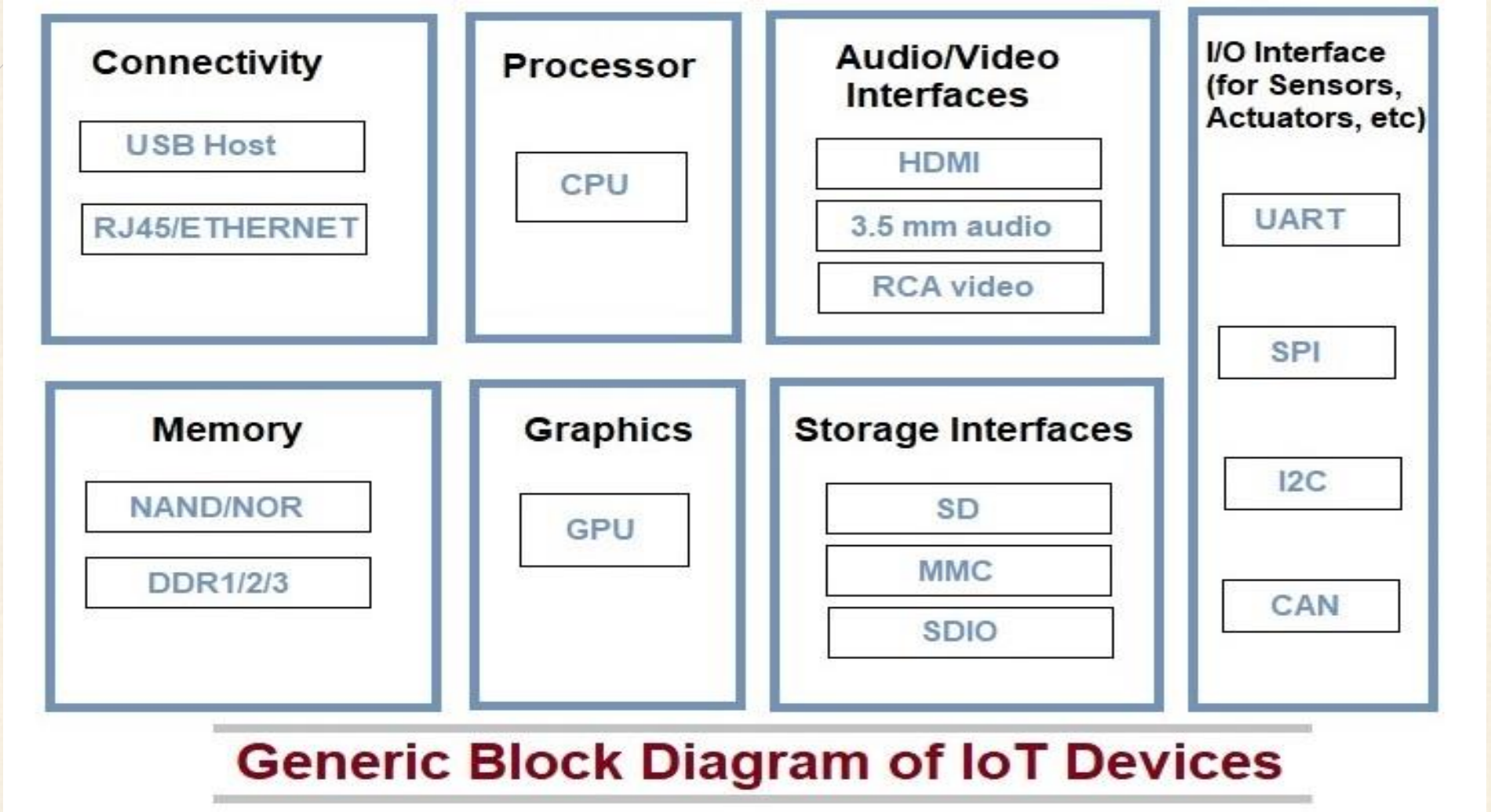
# Characteristics –

**Intelligence**. The aspect of intelligence as in the sensing capabilities in IoT devices and the intelligence gathered from big data analytics (also artificial intelligence).

**Action**. The consequence of intelligence. This can be manual action, action based upon debates regarding phenomena (for instance in smart factory decisions) and automation, often the most important piece.

**Ecosystem**. The place of the Internet of Things from a perspective of other technologies, communities, goals and the picture in which the Internet of Things fits. The Internet of Everything dimension, the platform dimension and the need for solid partnerships.

# Physical Design of IoT –




Generic Block Diagram of IoT Devices

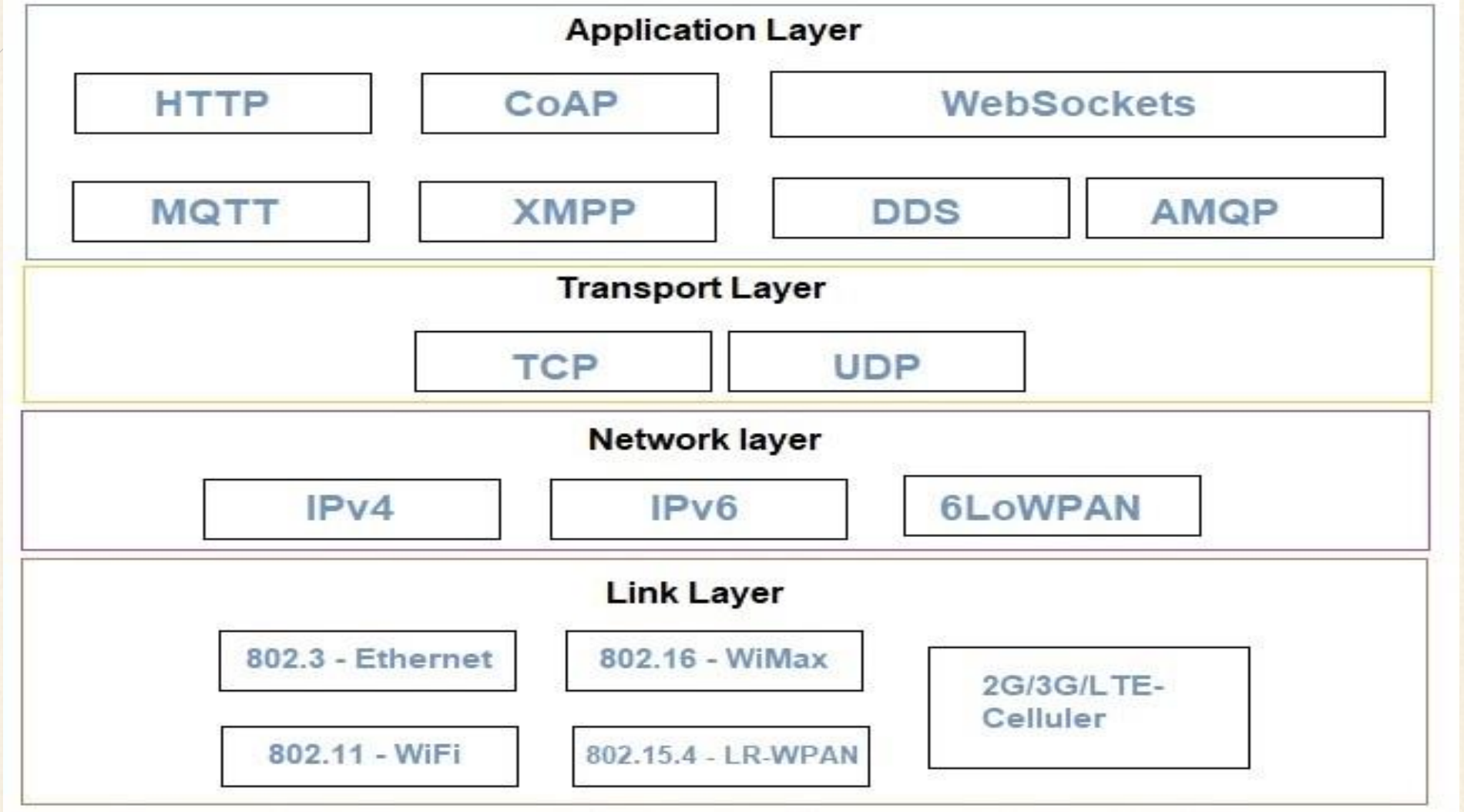
# Physical Design of IoT –

Basically Things refers to IoT Devices which have unique identities and can perform remote sensing, actuating and monitoring capabilities.

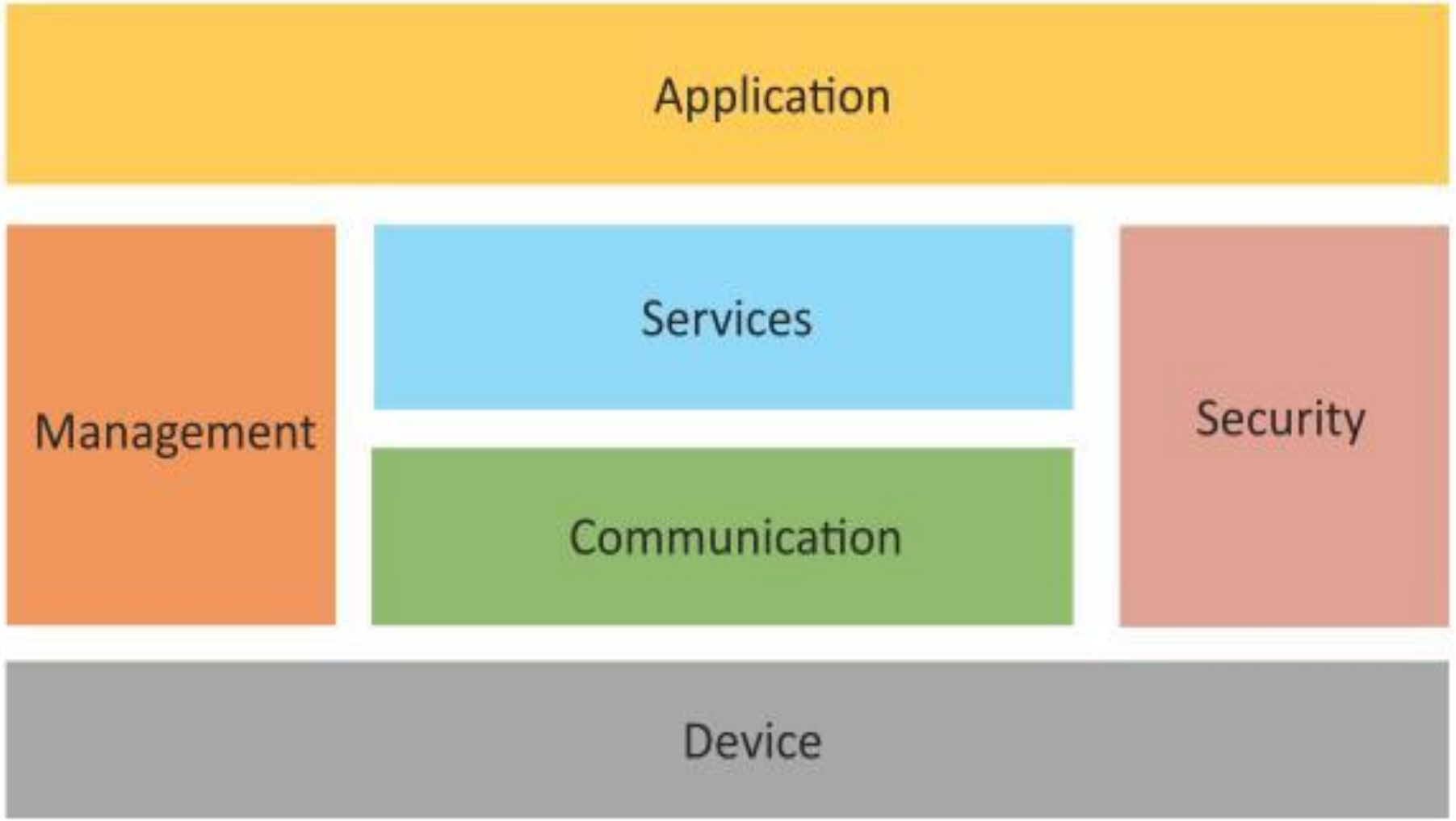
Things are is main part of IoT Application. IoT Devices can be various type, Sensing Devices, Smart Watches, Smart Electronics appliances, Wearable Sensors, Automobiles, and industrial machines.

These devices generate data in some forms or the other which when processed by data analytics systems leads to useful information to guide further actions locally or remotely.

# IoT Protocols–



# Logical Design of IoT–



## Logical Design of IoT –

**Logical Design of IoT:**

In Logical design of Internet of things. Logical design of IoT system refers to an abstract representation of the entities & processes without going into the low-level specifies of the implementation. For understanding Logical Design of IoT, we describe given below terms.

- IoT Functional Blocks
- IoT Communication Models
- IoT Communication APIs

## Logical Design of IoT –

**IoT Functional Blocks**

**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 blocks 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. Application also allow users to view the system status and view or analyze the processed data.

## Logical Design of IoT –

### IoT Communication Models

### Request-Response Model

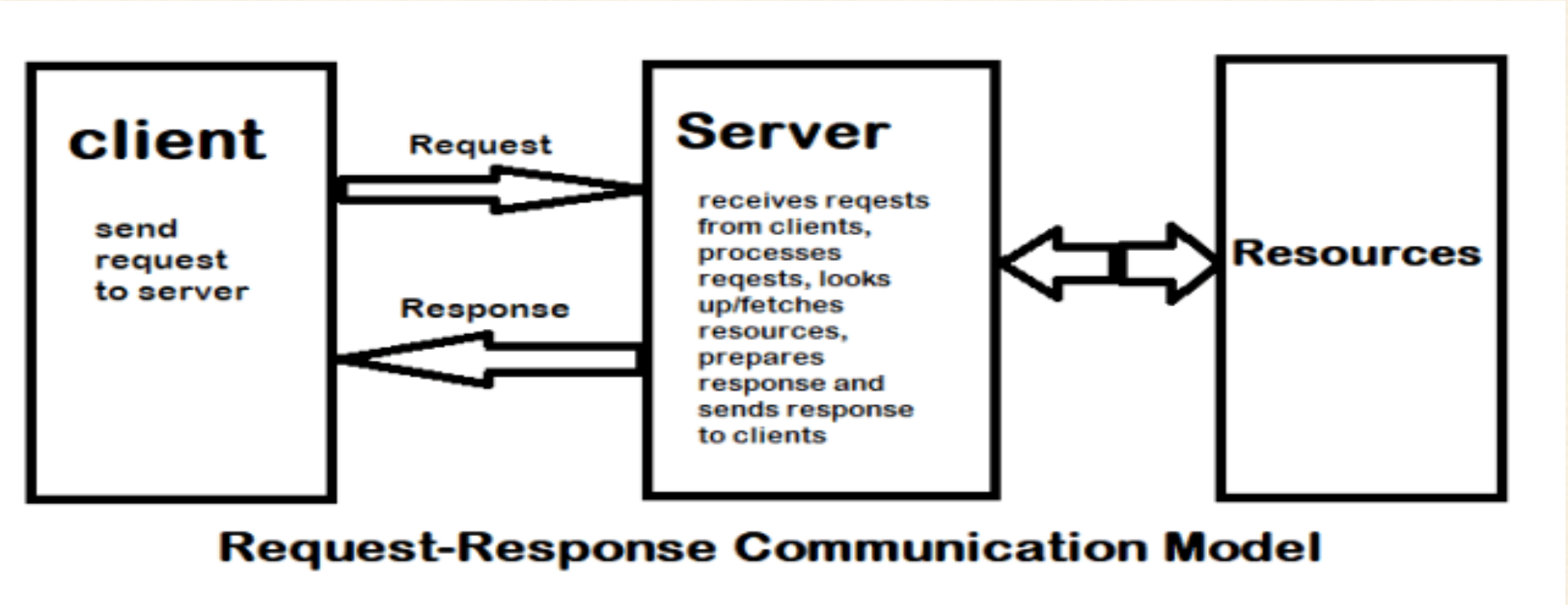
Request-response model is communication model in which the client sends requests to the server and the server responds to the requests. When the server receives a request, it decides how to respond, fetches the data, retrieves resource representation, prepares the response, and then sends the response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

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.

## Logical Design of IoT –

### IoT Communication Models

### Request-Response Model




Request-Response Communication Model

## Logical Design of IoT –

### IoT Communication Models

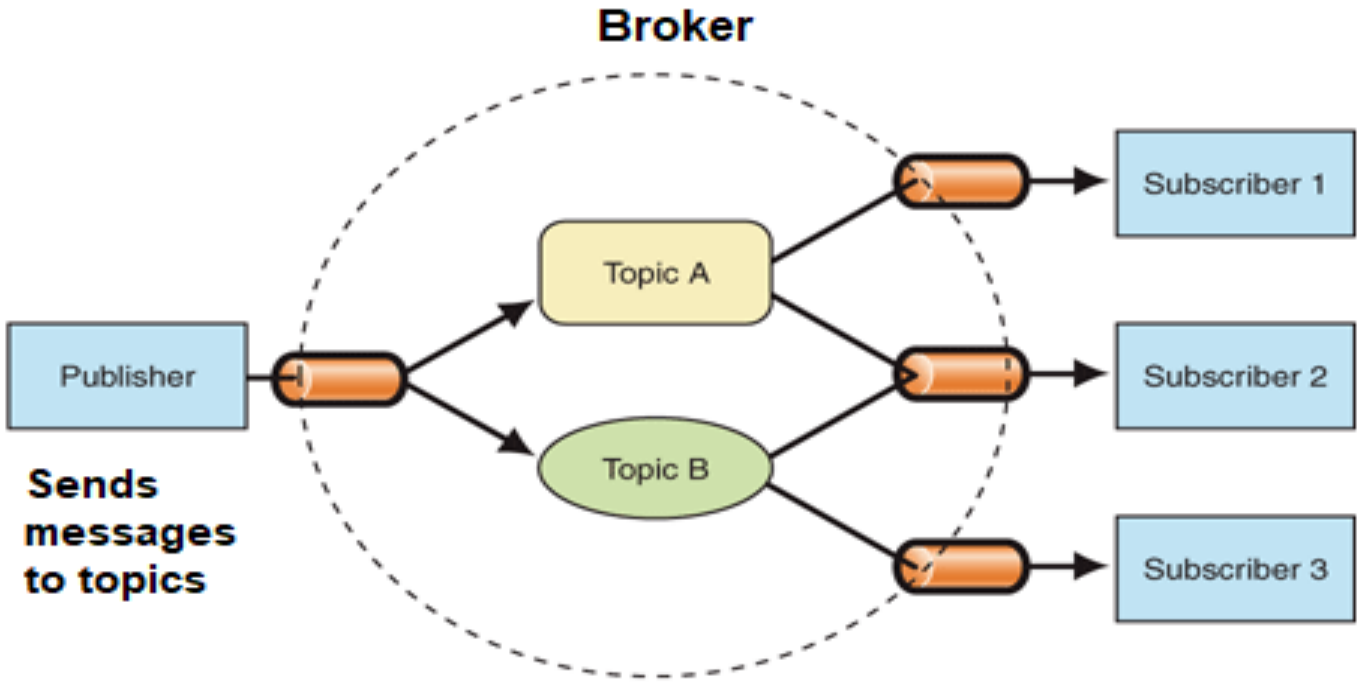
### Publish-Subscribe Model

Publish-Subscribe is a communication model that involves publishers, brokers and consumers. Publishers are the source of data. Publishers send the 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 receive data for a topic from the publisher, it sends the data to all the subscribed consumers.

## Logical Design of IoT –

### IoT Communication Models

### Publish-Subscribe Model



## Logical Design of IoT –

### IoT Communication Models

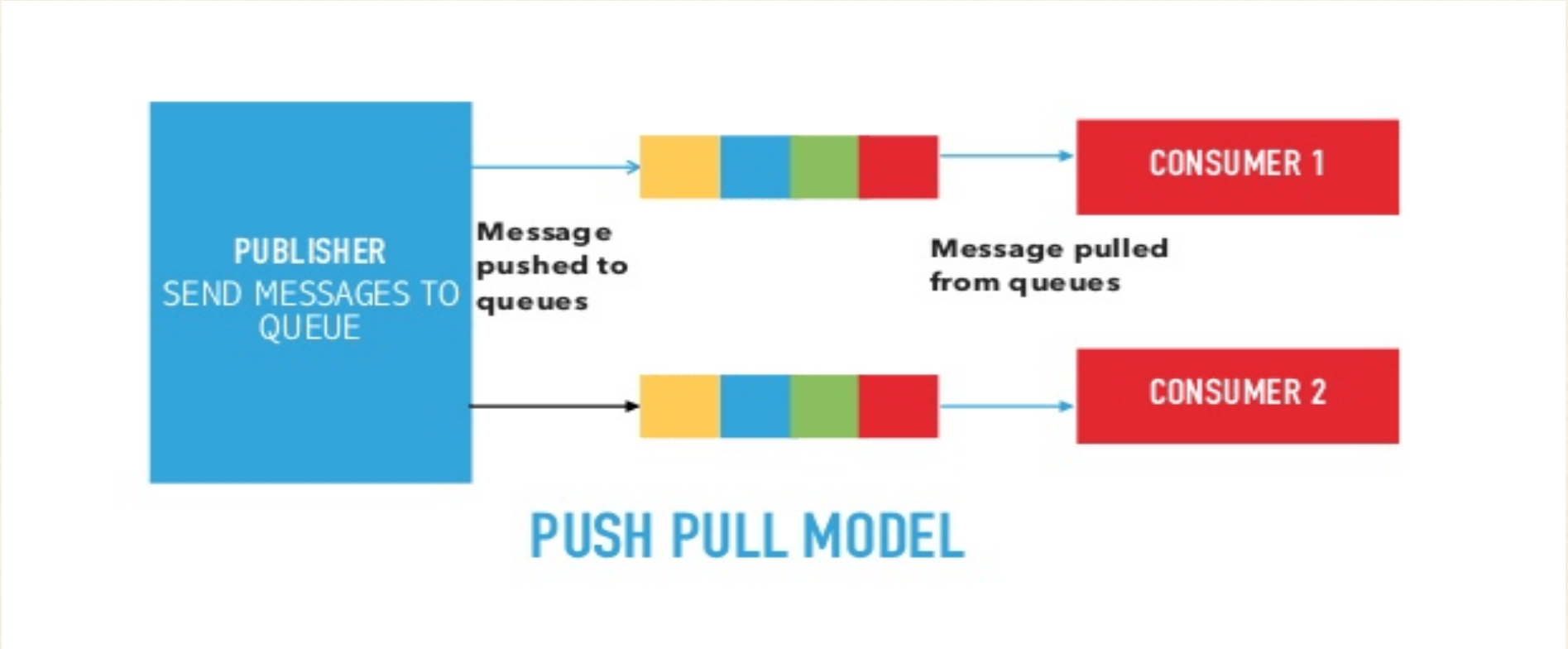
### Push-Pull Model

Push-Pull is a communication model in which the data producers push the data to queues and the consumers Pull the data from the Queues. Producers do not need to be aware of the consumers. Queues help in decoupling the messaging between the Producers and Consumers. Queues also act as a buffer which helps in situations when there is a mismatch between the rate at which the producers push data and the rate rate at which the consumer pull data.

## Logical Design of IoT –

### IoT Communication Models

### Push-Pull Model



## Logical Design of IoT –

### IoT Communication Models

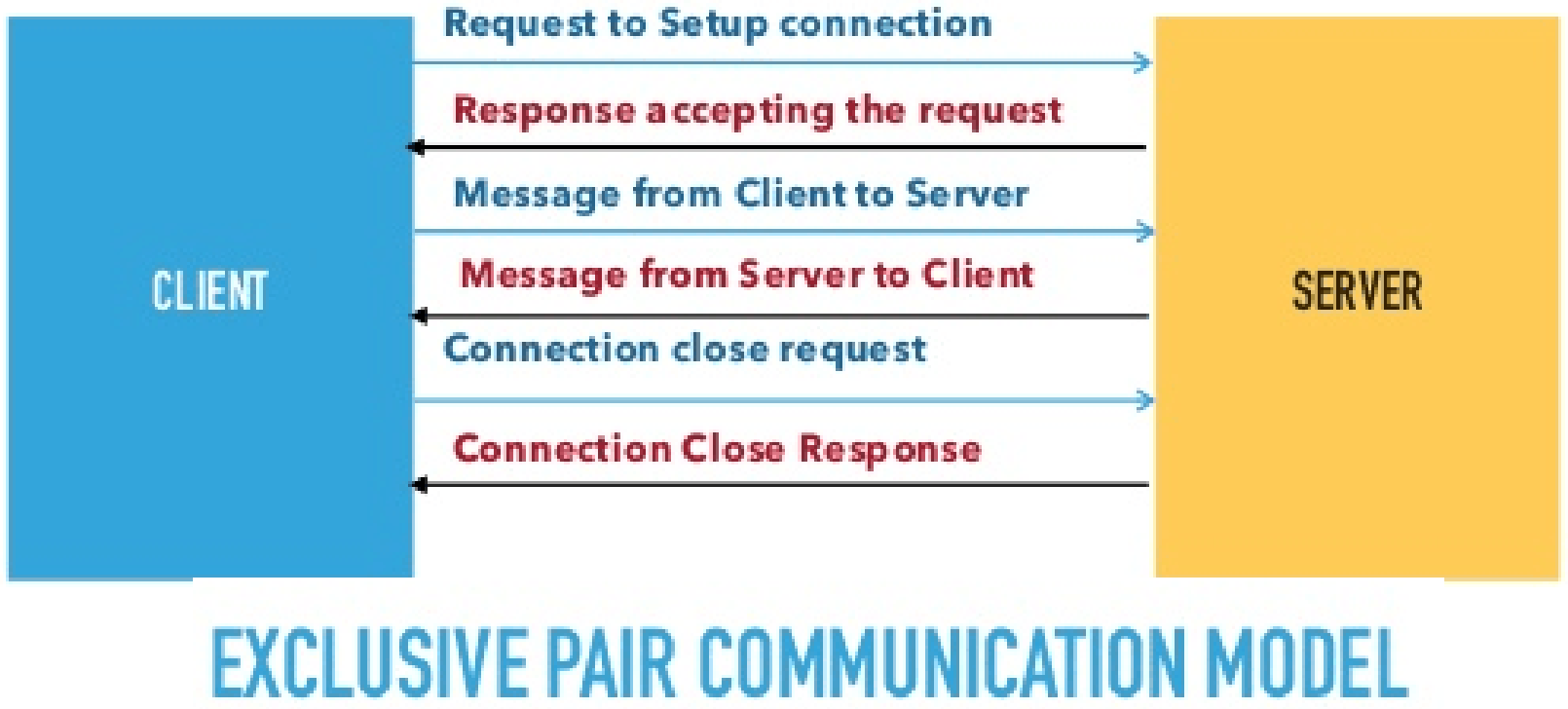
### Exclusive Pair Model

Exclusive Pair is a bidirectional, fully duplex communication model that uses a persistent connection between the client and server. Connection is setup it remains open until the client sends a request to close the connection. Client and server can send messages to each other after connection setup. Exclusive pair is stateful communication model and the server is aware of all the open connections.

## Logical Design of IoT –

### IoT Communication Models

### Exclusive Pair Model



## IoT Communication APIs-

### IoT Communication APIs

Generally we used Two APIs For IoT Communication. These IoT Communication APIs are:

- REST-based Communication APIs
- WebSocket-based Communication APIs

# IoT Communication APIs-

## IoT Communication APIs

### REST-based Communication APIs

Representational state transfer (REST) is a set of architectural principles by which you can design Web services the Web APIs that focus on systems resources and how resource states are addressed and transferred. REST APIs that follow the request response communication model, the rest architectural constraint applies to the components, connector and data elements, within a distributed hypermedia system. The rest architectural constraint are as follows:

## IoT Communication APIs-

**Client-server –** The principle behind the client-server constraint is the separation of concerns. for example clients should not be concerned with the storage of data which is concern of the serve. Similarly the server should not be concerned about the user interface, which is concern of the clien. Separation allows client and server to be independently developed and updated.

**Stateless –** Each request from client to server must contain all the information necessary to understand the request, and cannot take advantage of any stored context on the server. The session state is kept entirely on the client.

**Cache-able –** Cache constraints requires that the data within a response to a request be implicitly or explicitly leveled as cache-able or non-cache-able. If a response is cache-able, then a client cache is given the right to reuse that response data for later, equivalent requests. caching can partially or completely eliminate some instructions and improve efficiency and scalability.

## IoT Communication APIs-

**Layered system** – layered system constraints, constrains the behavior of components such that each component cannot see beyond the immediate layer with they are interacting. For example, the client cannot tell whether it is connected directly to the end server or two an intermediary along the way. System scalability can be improved by allowing intermediaries to respond to requests instead of the end server, without the client having to do anything different.

**Uniform interface** – uniform interface constraints requires that the method of communication between client and server must be uniform. Resources are identified in the requests (by URIsin web based systems) and are themselves is separate from the representations of the resources data returned to the client. When a client holds a representation of resources it has all the information required to update or delete the resource you (provided the client has required permissions). Each message includes enough information to describe how to process the message.

**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.

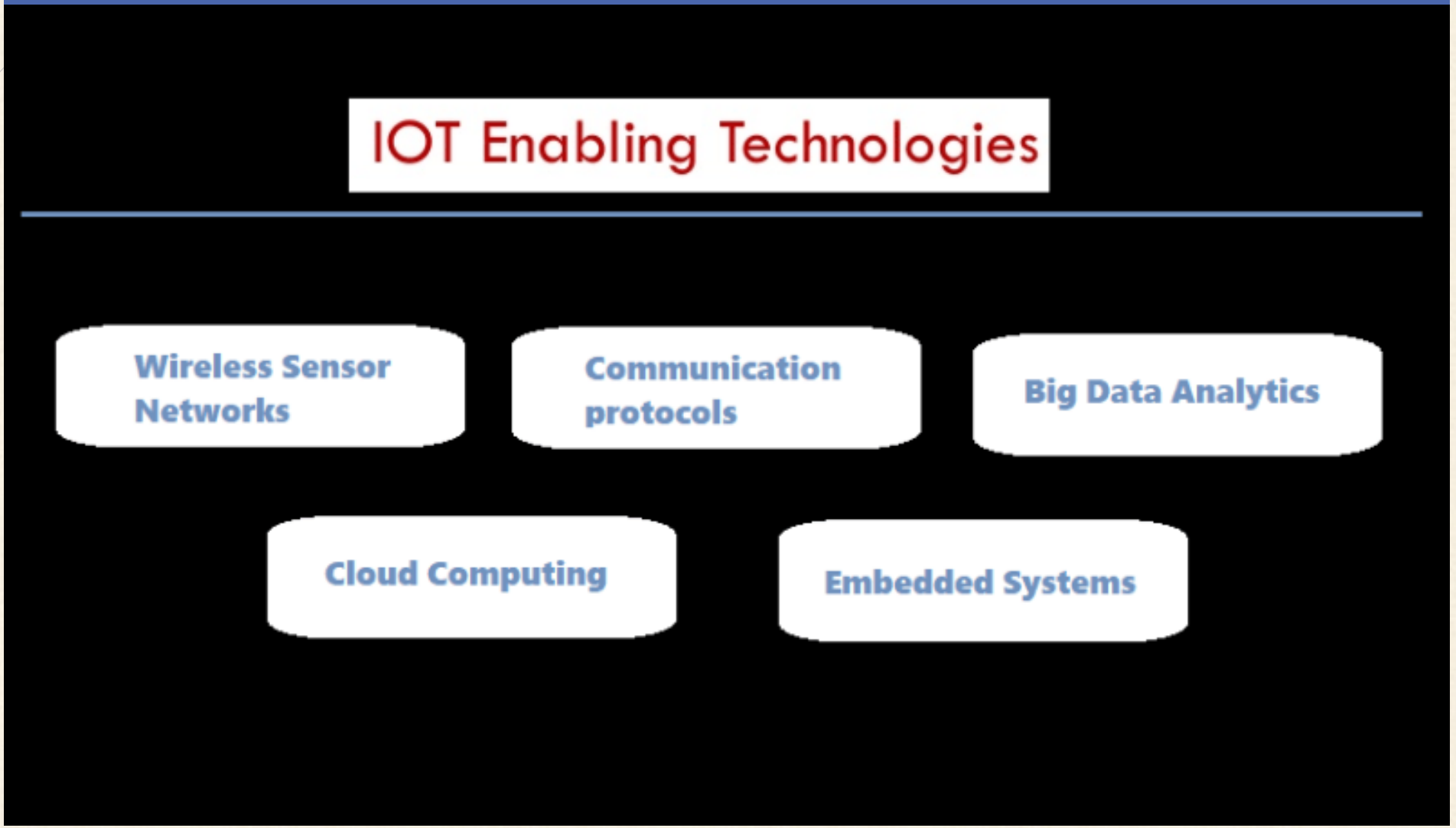
# WebSocket based communication API-

## WebSocket based communication API

Websocket APIs allow bi-directional, full duplex communication between clients and servers. Websocket APIs follow the exclusive pair communication model. Unlike request-response model such as REST, the Websocket APIs allow full duplex communication and do not require new connection to be setup for each message to be sent. Websocket communication begins with a connection setup request sent by the client to the server. The request (called websocket handshake) is sent over HTTP and the server interprets it is an upgrade request. If the server supports websocket protocol, the server responds to the websocket handshake response. After the connection setup client and server can send data/messages to each other in full duplex mode. Websocket API reduce the network traffic and latency as there is no overhead for connection setup and termination requests for each message. Websocket suitable for IoT applications that have low latency or high throughput requirements. So Web socket is most suitable IoT Communication APIs for IoT System.

# WebSocket based communication API-

## IoT Enabling Technologies-

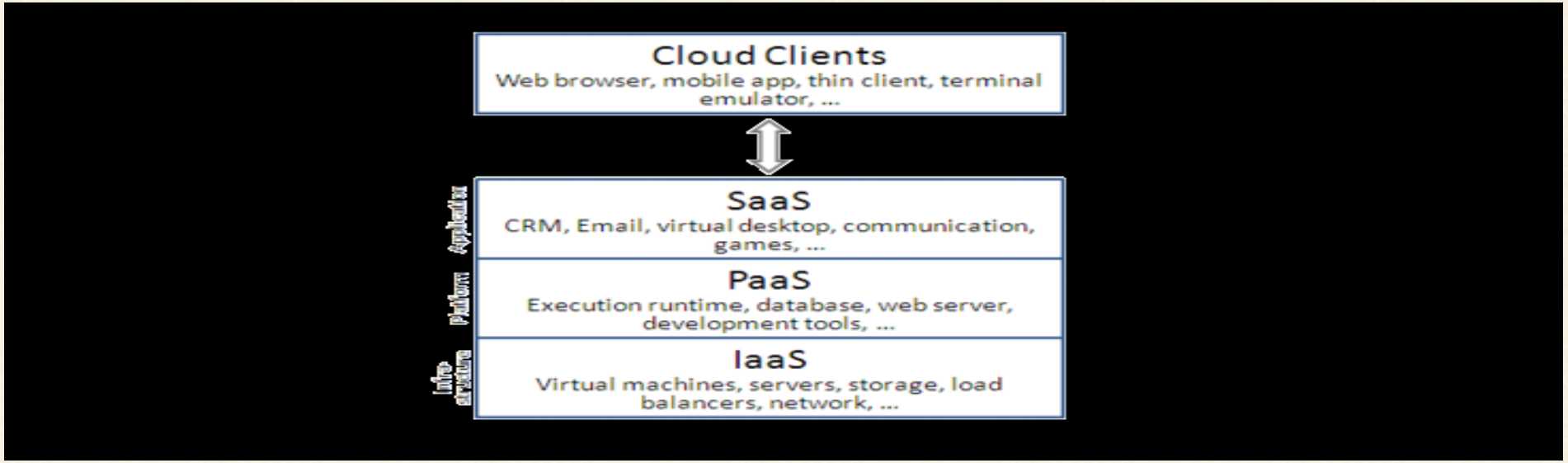


# IoT Enabling Technologies-

- **Wireless Sensor Networks**

A wireless sensor network comprises of distributed device with sensor which are used to monitor the environmental and physical conditions. A WSN consists of a number of end-nodes and routers and a coordinator. End Nodes have several sensors attached to them in node can also act as routers. Routers are responsible for routing the data packets from end-nodes to the coordinator.

- **Cloud Computing**



# IoT Enabling Technologies-

- 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.

- 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.

# IoT Enabling Technologies-

- 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

- **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

# IoT Levels and Deployment Templates-

- 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.

# IoT Levels and Deployment Templates-

**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 templateof 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 Levels and Deployment Templates-

- **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 info sense movement or vibrations Websocket service is used because sensor data can be sent in real time. **Accelerometer Gyroscope**

# IoT Levels and Deployment Templates-

## 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 Levels and Deployment Templates-

## 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 analyzes 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

# IoT security challenges-

## Secure constrained devices

Many IoT devices have limited amounts of storage, memory, and processing capability and they often need to be able to operate on lower power, for example, when running on batteries.

Security approaches that rely heavily on encryption are not a good fit for these constrained devices, because they are not capable of performing complex encryption and decryption quickly enough to be able to transmit data securely in real-time.

## Authorize and authenticate devices

With so many devices offering potential points of entry within an IoT system, device authentication and authorization is critical for securing IoT systems.

Devices must establish their identity before they can access gateways and upstream services and apps. However, there are many IoT devices that fall down when it comes to device authentication, for example, by using weak basic password authentication or using passwords unchanged from their default values.

# IoT security challenges-

- **Manage device updates**

  Applying updates, including security patches, to firmware or software that runs on IoT devices and gateways presents a number of challenges. For example, you need to keep track of which updates are available, apply them synchronously across distributed environments with heterogeneous devices that communicate through a range of different networking protocols. You have to implement a graceful rollback-strategy in case the update fails.

- **Secure communication**

  Once the devices themselves are secured, the next IoT security challenge is to ensure that communication across the network between devices and cloud services or apps is secure.

  Many IoT devices don't encrypt messages before sending them over the network. However, best practice is to use transport encryption and to adopt standards like TLS. Using separate networks to isolate devices also helps with establishing secure, private communication, so that data transmitted remains confidential. Common measures include using Firewalls, restricting physical access to gateway devices, using randomly generated one-time-password, and turn off the OS features that aren't required by the device.

# IoT security challenges-

**Ensure data privacy and integrity**

It is also important that wherever the data ends up after it has been transmitted across the network, it is stored and processed securely. Implementing data privacy includes redacting or anonymizing sensitive data before it is stored or using data separation to decouple personally identifiable information from IoT data payloads. Data that is no longer required should be disposed of securely, and if data is stored, maintaining compliance with legal and regulatory frameworks is also an important challenge. There has been an increased interest in developing systems that enable federated machine learning, where the data stays local to the device and the machine learning work happens at edge, with only the learnings/insights being shared with the cloud. The essence of such systems is that the algorithm should come to the data instead of the conventional methods where all the data goes to the algorithm.

# IoT security challenges-

- **Secure web, mobile, and cloud applications**

  Web, mobile, and cloud apps and services are used to manage, access, and process IoT devices and data, so they must also be secured as part of a multi-layered approach to IoT security. All the standard practices to secure mobile and web apps should be followed.

- **Ensure high availability**

  As we come to rely more on IoT within our day-to-day lives, IoT developers must consider the availability of IoT data and the web and mobile apps that rely on that data as well as our access to the physical things managed by IoT systems. The potential for disruption as a result of connectivity outages or device failures, or arising as a result of attacks like denial of service attacks, is more than just inconvenience. In some applications, the impact of the lack of availability could mean loss of revenue, damage to equipment, or even loss of life.

# IoT security challenges-

- **Prevent incidents by detecting vulnerabilities**

  Despite best efforts, security vulnerabilities and breaches are inevitable. How do you know if your IoT system has been compromised? In large scale IoT systems, the complexity of the system in terms of the number of devices connected, and the variety of devices, apps, services, and communication protocols involved, can make it difficult to identify when an incident has occurred. Strategies for detecting vulnerabilities and breaches include monitoring network communications and activity logs for anomalies, engaging in penetration testing and ethical hacking to expose vulnerabilities, and applying security intelligence and analytics to identify and notify when incidents occur.

- **Manage vulnerabilities**

  The complexity of IoT systems also makes it challenging to assess the repercussions of a vulnerability or the extent of a breach in order to manage its impact. Challenges include identifying which devices were affected, what data or services were accessed or compromised and which users were impacted, and then taking immediate actions to resolve the situation.

# IoT security challenges-

- **Predict and preempt security issues**

  A longer-term IoT security challenge is to apply security intelligence not only for detecting and mitigating issues as they occur, but also to predict and proactively protect against potential security threats. Threat modeling is one approach used to predict security issues. Other approaches include applying monitoring and analytics tools to correlate events and visualize unfolding threats in real-time, as well as applying AI to adaptively adjust security strategies applied based on the effectiveness of previous actions. The essence is to minimize human intervention and offload as much work as possible to algorithms since continuous manual examination is almost impossible.