Unit 1

Q.1. Define Iot an explain the characteristics of IoT

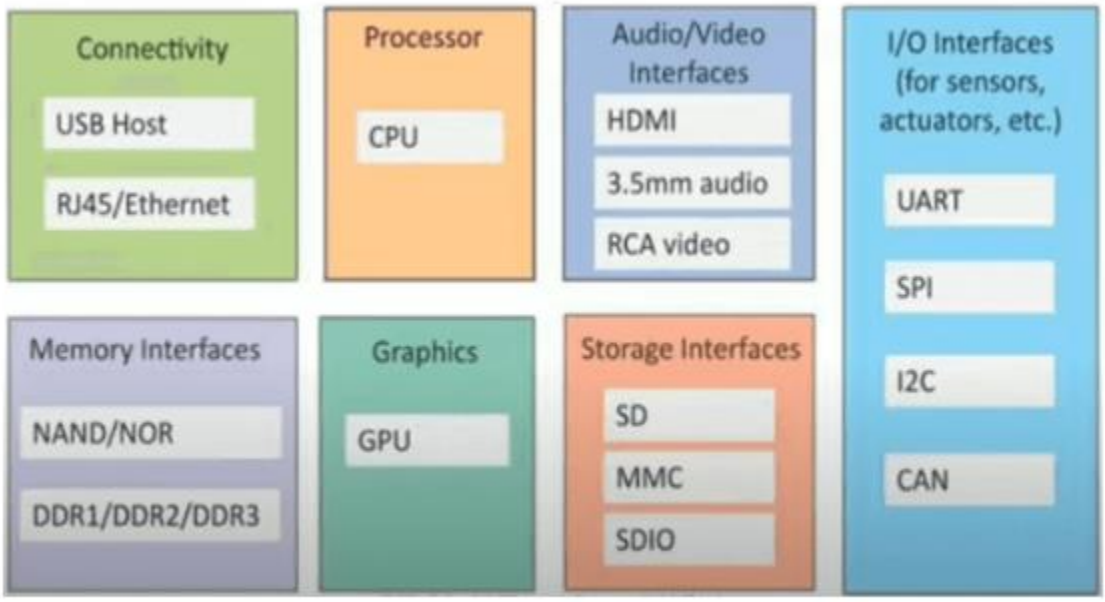
1. The Internet of things describes physical objects with sensors, processing ability, software and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks.
2. It is the way to interconnect with the help of internet devices that can be embedded to implement the functionality in everyday objects by enabling them to send and receive data.
3. IoT systems are typically composed of several components, including IoT devices, communication networks, gateways, and cloud-based data processing and storage systems.
4. IoT devices use sensors and other technologies to collect data, and then send that data to the cloud for analysis and storage.
5. The cloud also provides a centralized platform for managing and controlling IoT devices and networks.
6. **Characteristics:**
    a. Dynamic and Self-Adapting:
        i. IoT systems are designed to adapt and respond to changing conditions or requirements.
        ii. They can adjust their behavior, settings, or operations based on real-time data and feedback from the environment or users.
    b. Self-Configuring:
        i. IoT devices are capable of automatically configuring themselves and establishing connections with other devices or networks.
        ii. They can discover and join a network, obtain necessary network settings, and start communicating without requiring manual intervention.
    c. Interoperable Communication Protocols:
        i. IoT devices employ standardized communication protocols that allow them to interact and exchange data with each other seamlessly.
        ii. These protocols enable devices from different manufacturers or using different technologies to communicate effectively, fostering interoperability within the IoT ecosystem.
    d. Unique Identity:
        i. Each IoT device is assigned a unique identity or identifier, such as an IP address or a device ID.
        ii. This unique identifier helps in distinguishing and identifying individual devices within a network, facilitating secure communication and management.
    e. Integrated into Information Network:
        i. IoT devices are integrated into larger information networks, such as the internet or private networks.
        ii. This integration enables the seamless flow of data between devices, cloud platforms, and applications.

iii. It also allows for centralized monitoring, control, and management of IoT devices from a single interface.

7. These characteristics further emphasize the dynamic, intelligent, and interconnected nature of IoT systems, enabling them to adapt to changing conditions, communicate effectively, and be seamlessly integrated into existing information networks.

Q.2. Explain with suitable diagram physical design of IoT

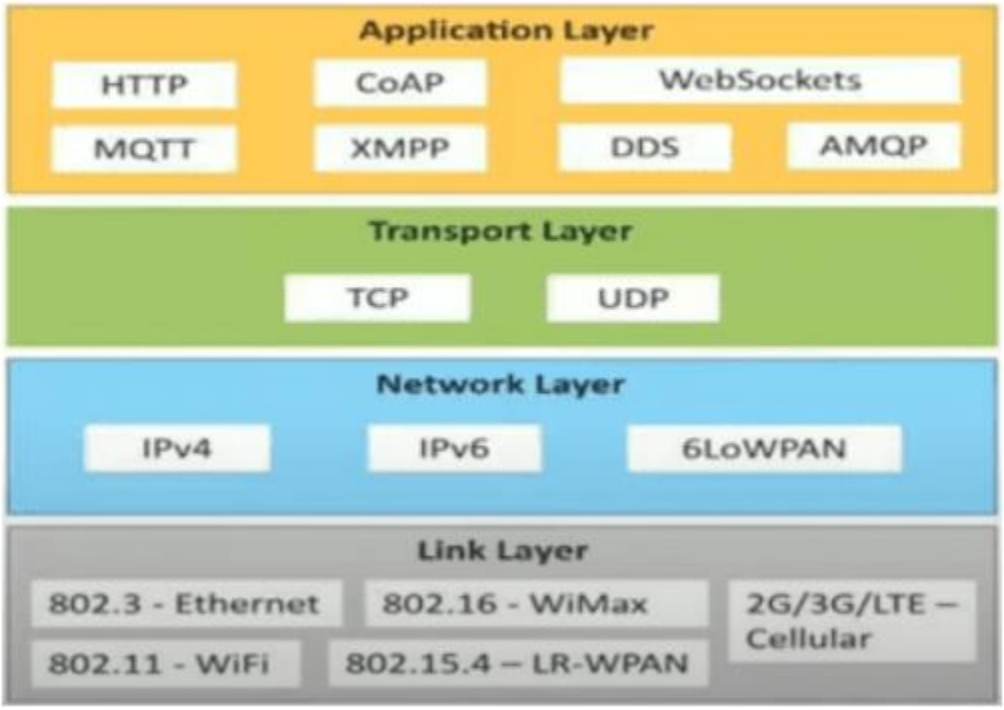
- The physical design of an IoT system is referred to as the Things/Devices and protocols that are used to build an IoT system.
- All these things/Devices are called Node Devices and every device has a unique identity that performs remote sensing, actuating, and monitoring work and the protocols that are used to establish communication between the Node devices and servers over the internet.
    - **Things in IoT:**
        - Things/Devices are used to build a connection, process data, provide interfaces, provide storage, and provide graphics interfaces in an IoT system.
        - All these generate data in a form that can be analyzed by an analytical system and program to perform operations.
        - For example, a temperature sensor that is used to analyze the temperature generates the data from a location and is then determined by algorithms.



devices in IoT(Internet of things)

    - **IoT Protocols-**
        - These protocols are used to establish communication between a node device and a server over the internet.

- It helps to send commands to an IoT device and receive data from an IoT device over the internet.
- We use different types of protocols that are present on both the server and client side and these protocols are managed by network layers like application, transport, network, and link layer.

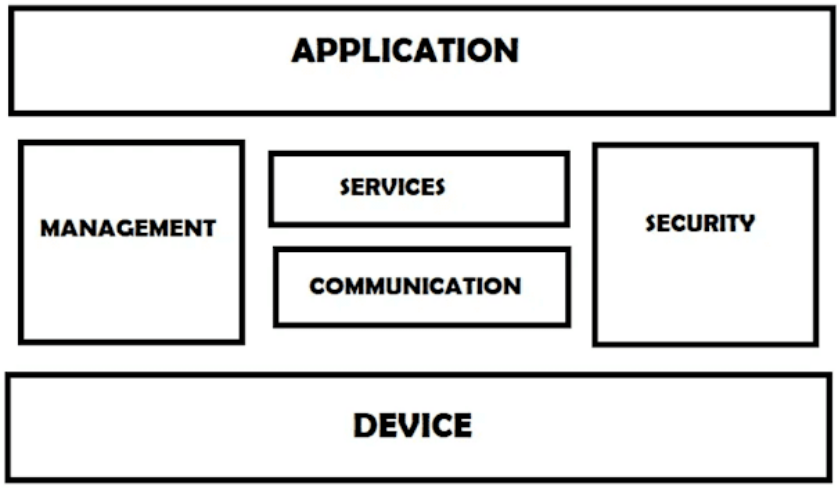


- **Application Layer protocol-**In this layer, protocols define how the data can be sent over the network with the lower layer protocols using the application interface. These protocols include HTTP, WebSocket, XMPP, MQTT, DDS, and AMQP protocols.
- **Transport Layer-** This layer is used to control the flow of data segments and handle error control. Also, these layer protocols provide end-to-end message transfer capability independent of the underlying network.
- **Network Layer-** This layer is used to send datagrams from the source network to the destination network. We use IPv4 and IPv6 protocols as host identification that transfers data in packets.
- **Link Layer-** Link-layer protocols are used to send data over the network's physical layer. it also determines how the packets are coded and signaled by the devices.

Q.3. Explain IoT Functional blocks in logical design of IoT

a. The logical design of an IoT system refers to an abstract representation of entities and processes without going into the low-level specifications of implementation.
b. It uses Functional Blocks, Communication Models, and Communication APIs to implement a system.

i. **IoT Functional blocks-** An IoT system consists of a number of functional blocks like Devices, services, communication, security, and application that provide the capability for sensing, actuation, identification, communication, and management.



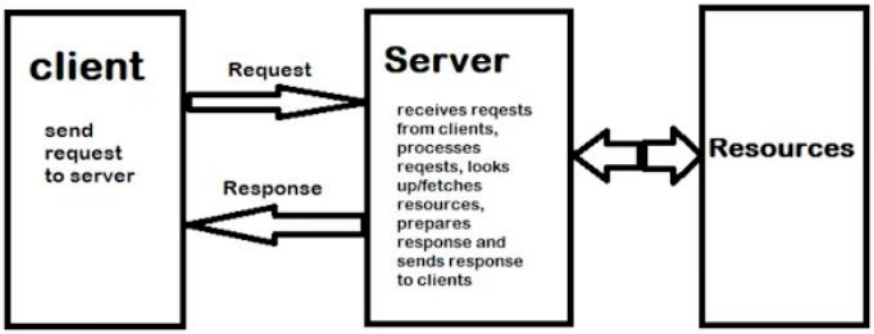
IoT functional blocks

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

Q.4. Write short Notes on IoT Communication Models

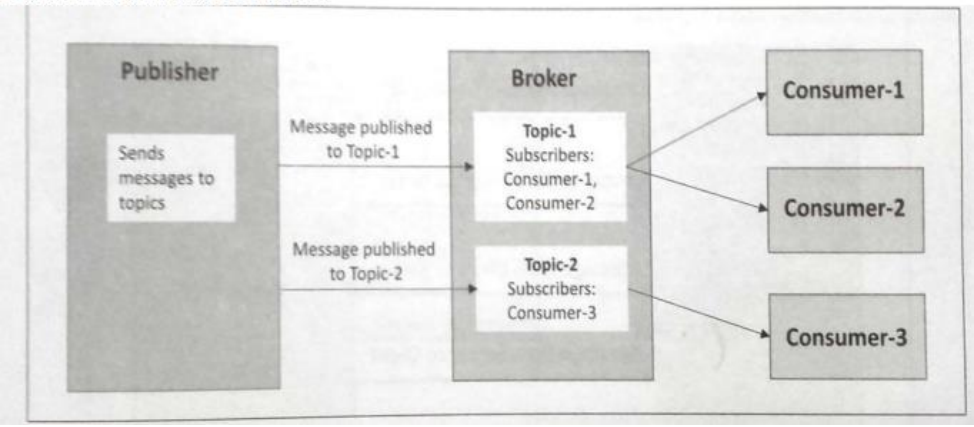
IoT Communication Models- There are several different types of models available in an IoT system that is used to communicate between the system and server like the request-response model, publish-subscribe model, push-pull model, exclusive pair model, etc.

1. **Request-Response Communication Model-** This model is a communication model in which a client sends the request for data to the server and when the server receives the request it fetches the data, retrieves the resources and prepares the response, and then sends the data back to the client.

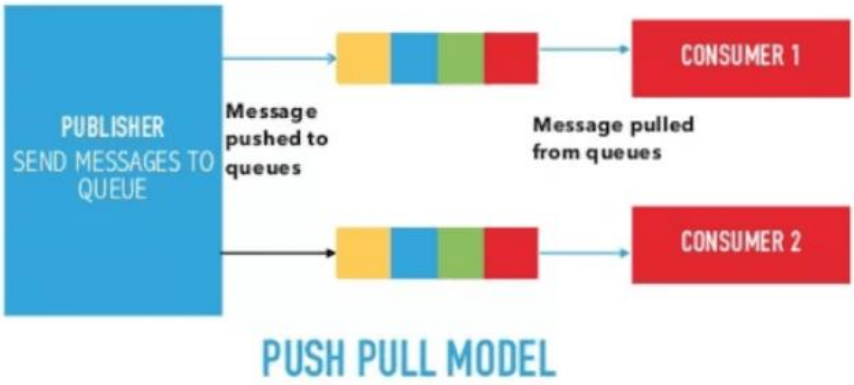


Request-Response Communication Model

2. **Publish-Subscribe Communication Model-** This communication model includes broker, publisher and the consumer. Publishers are the source of data but they are not aware of consumers. They send the data, which is managed by the brokers. In addition, when a consumer subscribes to a topic, which is managed by the broker, the broker sends the data received by the publisher to all the subscribed consumers.

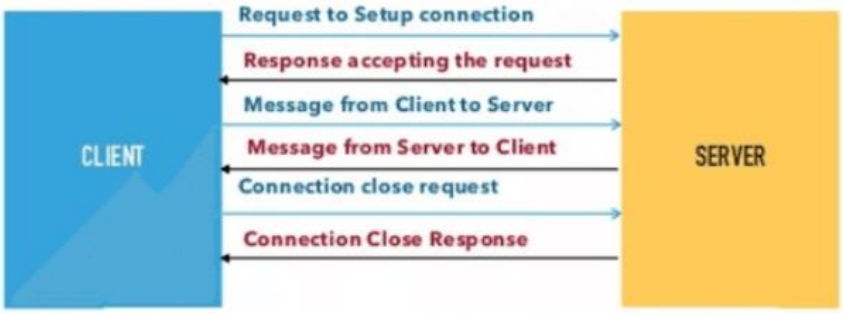


3. **Push-Pull Communication Model-** It is a communication model in which the data is pushed by the producers in a queue and the consumers can pull the data from the queues. Here producers are not aware of the consumers.



PUSH PULL MODEL

4. **Exclusive Pair:** It is bi-directional, full duplex communication model that uses a persistent connection between the client and server. Once the connection is set up, it remains open until the client sends a request to close the connection. It is a tasteful communication model and the server is aware of all the open connections.



EXCLUSIVE PAIR COMMUNICATION MODEL

Q.5. Explain the two Communication APIs used by IoT system

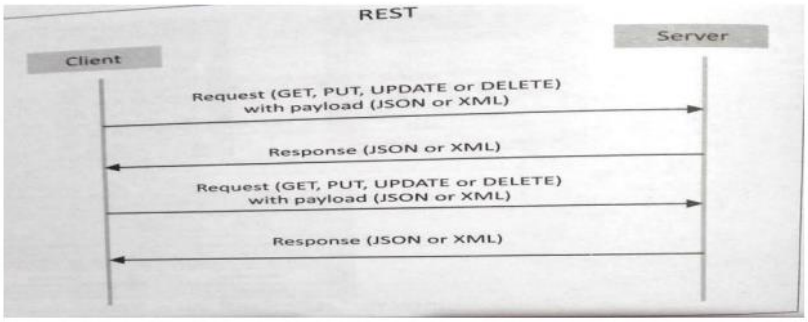
a. IoT communication APIs- These APIs like REST and Web Socket are used to communicate between the server and system in IoT.

i. REST-based communication APIs-

1. Representational State Transfer (REST) is an architectural style that provides a set of principles for designing networked applications, including web services.
2. REST APIs focus on resources and how their states are transferred between clients and servers using the HTTP protocol.
3. Key characteristics and constraints of REST-based communication APIs include:
    a. **Client-server architecture:** REST APIs separate the client (consumer of the API) and the server (provider of the API), allowing them to evolve independently.

b. **Stateless:** Each request from the client to the server must contain all necessary information for the server to understand and process it, without relying on any previous interaction.
c. **Cacheable:** REST APIs can leverage caching mechanisms to improve performance and reduce the load on the server.
d. **Layered system:** Intermediaries, such as proxies or gateways, can be placed between the client and server without affecting the overall communication.
e. **Uniform interface:** REST APIs utilize a uniform set of predefined operations (HTTP verbs like GET, POST, PUT, DELETE) to interact with resources.

Request-Response model used by REST:



b. Web Socket-based communication API-

i. Web Socket is a communication protocol that enables full duplex, bidirectional communication between a client and a server over a single, long-lived connection.

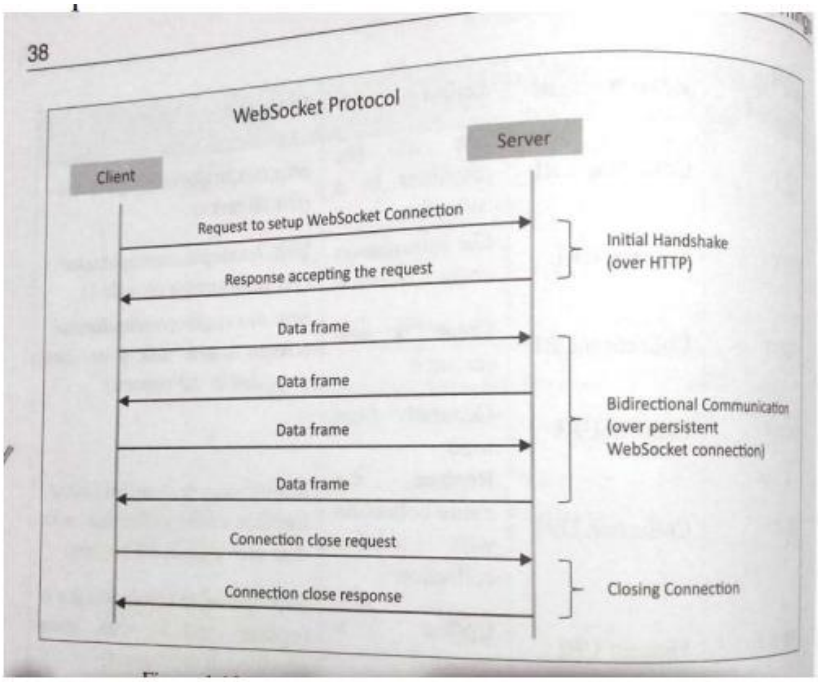
ii. Unlike traditional HTTP-based communication, where the client must initiate a new request for each interaction, Web Socket allows continuous communication in both directions.

iii. Key characteristics of Web Socket-based communication APIs include:

1. **Bi-directional communication:** Web Socket APIs enable real-time, simultaneous communication between the client and server. Both parties can send and receive messages at any time without the need for explicit requests.
2. **Full-duplex communication:** Web Socket connections allow data to flow in both directions simultaneously, eliminating the need for separate request-response cycles.
3. **Persistent connection:** Once established, Web Socket connections remain open until explicitly closed, enabling efficient and low-latency communication.

4. **Event-driven model:** Web Socket APIs are built on an event-driven model, where messages or events trigger actions or responses on the client or server side.

iv. Web Socket-based communication APIs are often used in scenarios where real-time, interactive, and continuous communication between IoT devices and servers is required, such as streaming data, live updates, or collaborative applications.



Both REST-based and Web Socket-based communication APIs have their strengths, are used in different IoT applications, depending on the specific requirements, and use cases.

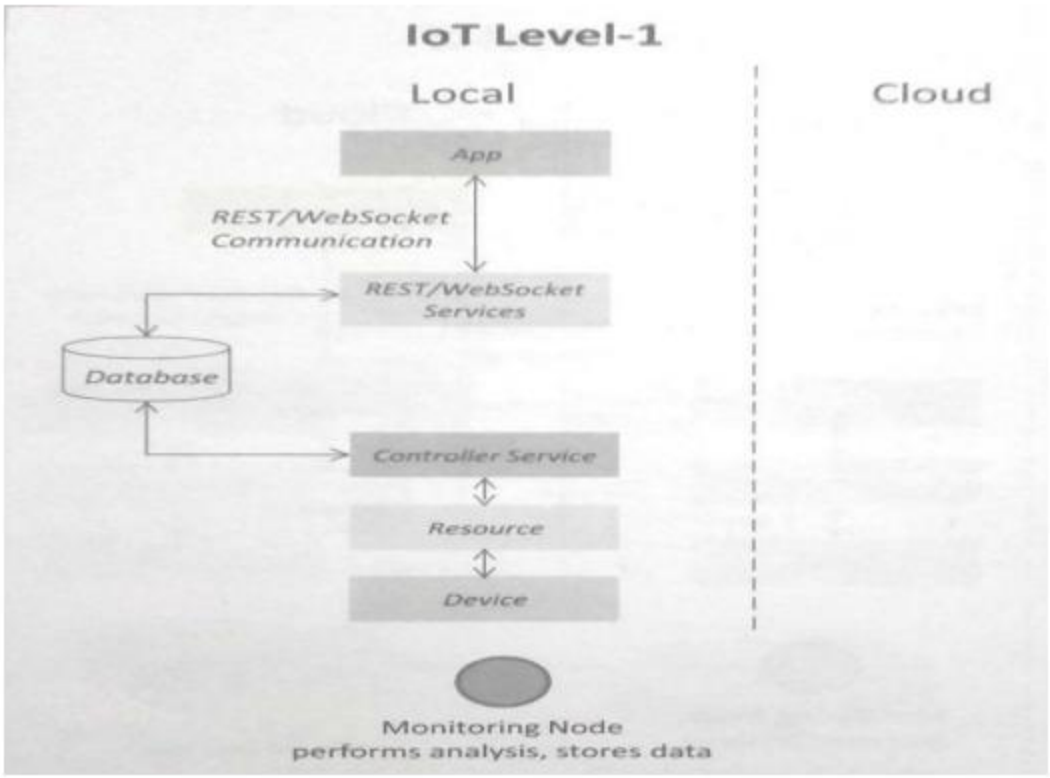
Q.6. Describe IoT level 1, level 2, level 3

a. **IOT Level 1:**

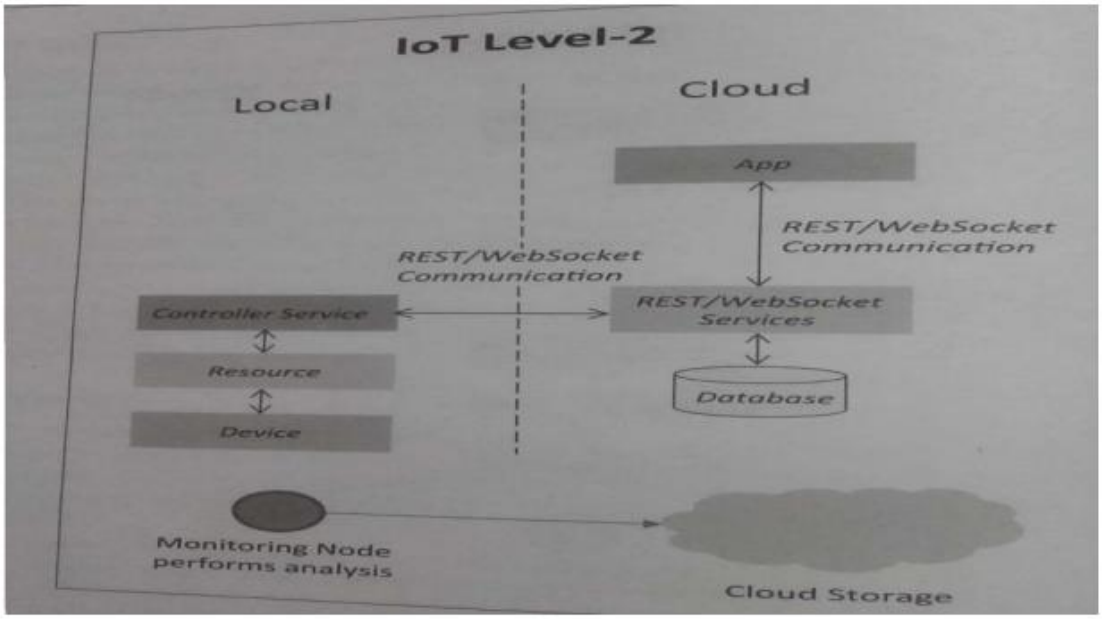
i. In Level 1, the IoT system consists of a single node that performs sensing and/or actuation, stores data, performs analysis, and hosts the application.

ii. This level is suitable for low-cost and low-complexity solutions where the data involved is not extensive, and the analysis requirements are not computationally intensive.

iii. Home automation is an example of a Level 1 IoT system, where a single smart home device controls various appliances and collects data for simple analysis.



b. **IoT Level2:**
   i. Level two IoT systems have a single node that performs sensing, actuation, and local analysis.
   ii. However, the data is stored in the cloud, and the application is usually cloud-based.
   iii. Level two systems are suitable for solutions where the data involved is more significant, but the primary analysis requirement can be done locally without heavy computational resources.
   iv. An example is a smart irrigation system, where sensors collect data on soil moisture and local analysis is performed to determine watering needs, with data being sent to the cloud for storage and additional analysis.



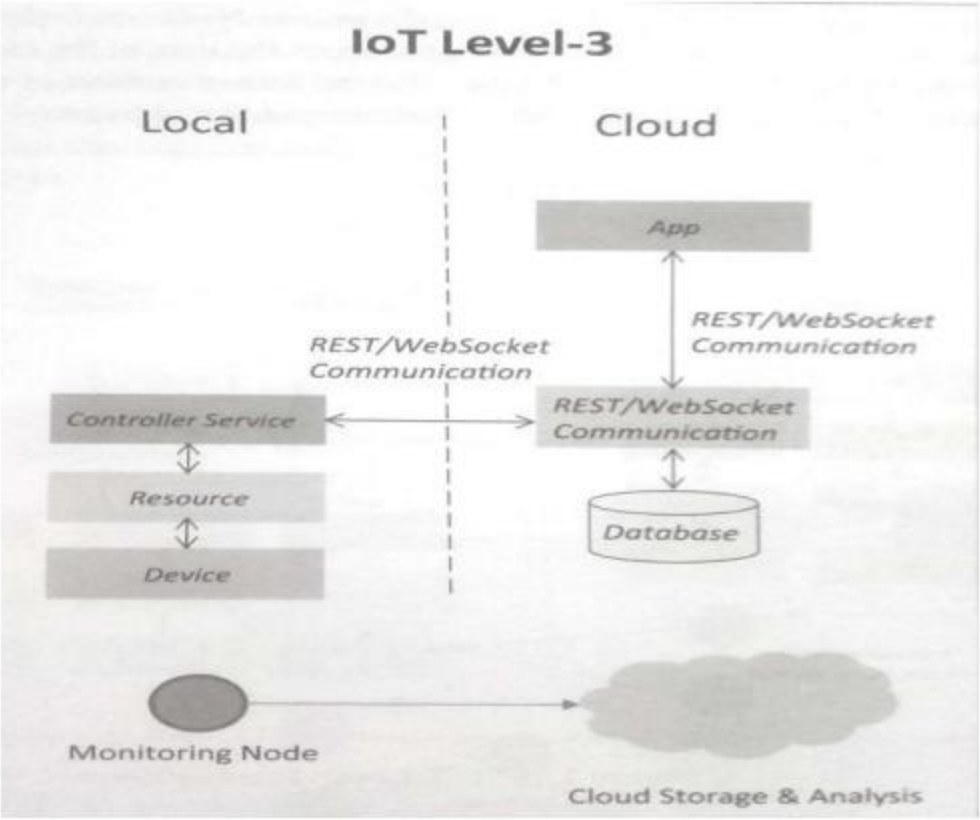
c. **IoT Level3:**

i. Level three IoT systems also have a single node, but data storage and analysis occur in the cloud, and the application is cloud-based.

ii. Level three systems are designed for solutions with extensive data and computationally intensive analysis requirements.

iii. An example could be a package handling tracking system, where sensors on packages collect data on location, temperature, and handling.

iv. The collected data is sent to the cloud for storage and advanced analysis to track the package's journey, monitor handling conditions, and optimize logistics.



Q.7. what is M2M Communication? Differentiate between IoT and M2M

1. Machine-to-Machine (M2M) communication refers to direct communication between devices or machines without human intervention.
2. It is a form of communication where machines exchange data and information with each other, enabling them to interact, collaborate, and make decisions without the need for human involvement.
3. M2M communication relies on various technologies and protocols to enable seamless connectivity and data exchange between devices.
4. These devices can include sensors, actuators, meters, industrial machinery, vehicles, and other IoT devices.
5. M2M communication allows devices to collect data, share information, and trigger actions based on predefined rules or algorithms.
6. M2M communication has a wide range of applications across industries, including industrial automation, transportation and logistics, healthcare, smart cities, agriculture, and energy management.
7. Examples of M2M communication include remote monitoring and control of equipment, asset tracking, smart grid management, vehicle telematics, and automated healthcare systems.

Basis of	IoT	M2M
Abbreviation	Internet of Things	Machine to Machine
Intelligence	Devices have objects that are responsible for decision making	Some degree of intelligence is observed in this.
Connection type used	The connection is via Network and using various communication types.	The connection is a point to point
Communication protocol used	Internet protocols are used such as HTTP, FTP, and Telnet.	Traditional protocols and communication technology techniques are used
Data Sharing	Data is shared between other applications that are used to improve the end-user experience.	Data is shared with only the communicating parties.
Internet	Internet connection is required for communication	Devices are not dependent on the Internet.
Type of Communication	It supports cloud communication	It supports point-to-point communication.
Computer System	Involves the usage of both Hardware and Software.	Mostly hardware-based technology
Scope	A large number of devices yet scope is large.	Limited Scope for devices.

Unit 2

Q.1. Explain ETSI M2M Architecture

1. ETSI (European Telecommunications Standards Institute) in 2009 formed a Technical Committee (TC) on M2M topics aimed at producing a set of standards for communication among machines from an end - to-end viewpoint.

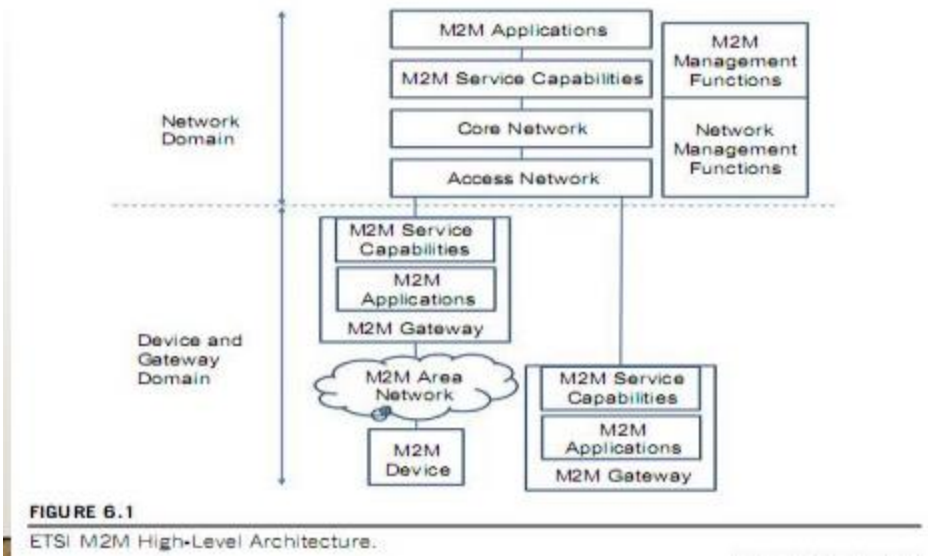


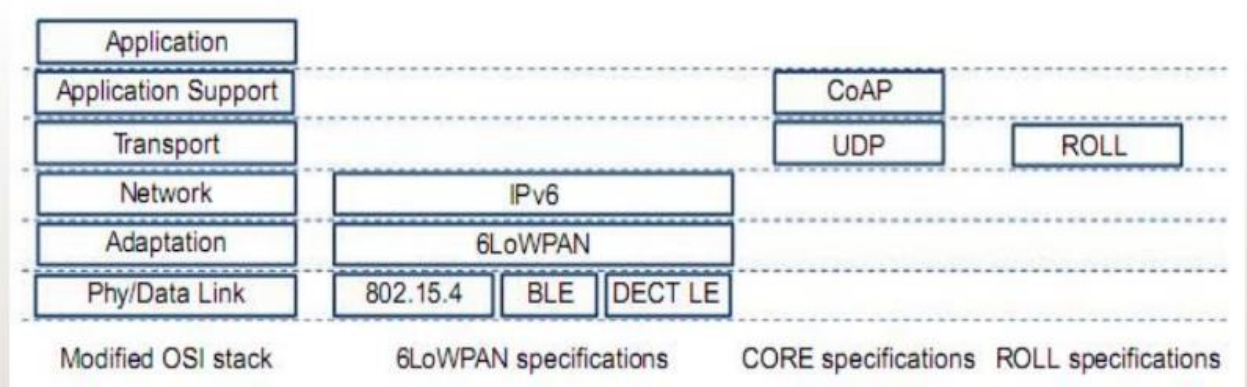
FIGURE 6.1
ETSI M2M High-Level Architecture.

2. **Key Elements:**
    a. **M2M Device:**
        i. This refers to the device participating in an M2M scenario, such as a device with a temperature sensor.
        ii. It contains M2M Applications and M2M Service Capabilities.
        iii. The M2M Device can connect to the Network Domain either directly or through an M2M Gateway.
    b. **Direct Connection:**
        i. This means that the M2M Device is capable of performing registration, authentication, authorization, management, and provisioning to the Network Domain.
        ii. It also implies that the device has the necessary physical layer to communicate directly with the Access Network.
    c. **M2M Area Network:**
        i. This network provides connectivity between M2M Devices and M2M Gateways. It can be a local area network (LAN) or a Personal Area Network (PAN).
        ii. Various networking technologies like Bluetooth, ZigBee, 6LoWPAN, MBUS, KNX (wired or wireless), and PLC can be used in M2M Area Networks.
    d. **M2M Gateway:**
        i. The M2M Gateway facilitates connectivity for M2M Devices in the M2M Area Network to the Network Domain.
        ii. It contains M2M Applications and M2M Service Capabilities.
        iii. The M2M Gateway may also provide services to legacy devices that are not directly visible to the Network Domain.
    e. **M2M Service Capabilities:**
        i. These are functions exposed to different M2M Applications through open interfaces.
        ii. They utilize underlying Core Network functions and aim to simplify applications by abstracting network functions.

f. **M2M Applications:**
    i. These are specific M2M applications, such as smart metering, that utilize the M2M Service Capabilities through the open interfaces.
    ii. M2M Applications leverage the functionality provided by the M2M Service Capabilities to enable communication and interaction between devices.
g. **Network Management Functions:**
    i. These functions are responsible for managing the Access and Core Network, including tasks like provisioning and fault management.
h. **M2M Management Functions:**
    i. These functions are specifically designed to manage the M2M Service Capabilities on the Network Domain.
    ii. Specific M2M Service Capabilities typically perform the management of an individual M2M Device or Gateway.

3. ETSI's Technical Committee on M2M was established to develop a set of standards that facilitate end-to-end communication among machines.
4. These standards help ensure interoperability, security, and efficiency in M2M deployments across various domains and applications.

Q2. Explain IETF Architecture for IoT



1. 6LoWPAN (IPv6 over Low-power WPAN), Core (Constrained RESTful Environments), and ROLL (Routing over Low power and Loss networks) are three working groups within the IETF that address different aspects of communication for constrained devices in IoT.
2. The architecture introduces an Application Support Layer that combines the Presentation and Session Layers from the traditional OSI model. It includes protocols like the IETF Constrained Application Protocol (Cap), which provides reliability and RESTful operation support for IoT applications.
3. An intermediate layer called the Adaptation Layer is positioned between the Physical/Data Link and the Network Layer. Its main function is to adapt network layer packets to the physical or link layer packets required by constrained devices. For example, the 6LoWPAN layer adapts IPv6 packets to specific link layer technologies such as IEEE 802.15.4, Bluetooth Low Energy (BLE), or DECT Low Energy.

4. The IETF Cap specification defines the Transport and Application Support Layers. It specifies transport packet formats, reliability support on top of UDP, and a RESTful application protocol with methods similar to HTTP (ARE, PUT, POST, and DELETE). Cap clients operate on Cap server resources.
5. Cap servers can be hosted on constrained devices themselves, and the term "server" does not necessarily imply a powerful machine.
6. The Core Link Format specification provides a discovery method for Cap server resources. For example, by sending a GET request to a specific well-defined server resource (./well-known/core), a Cap client can receive a response listing available Cap resources and their capabilities.
7. The Core interface specification defines interface types and the expected behavior of RESTful methods for interacting with resources.
8. The IETF IoT architecture does not currently include profile specifications similar to those in other IoT technologies like ZigBee. Profile specifications describe mappings between profile names, protocol stack behavior, information models, and serialization formats over communication media.
9. The Resource Directory (RD) is a Cap server resource (/rd.) that maintains a list of resources, their server contact information, types, interfaces, and other related information. It serves as a rendezvous mechanism for Cap server resource descriptions, allowing devices to publish their available resources and for clients to locate specific resource types.
10. A Mirror Server (MS) is another Cap server resource (/MS) that maintains a list of resources and their cached representations. It acts as a mirror of the original server and allows clients to retrieve the latest updated representations even when the original server is not directly accessible.
11. The IETF Core workgroup includes guidelines for mapping between HTTP and Cap, addressing schemes, response codes, media types, etc., to facilitate interworking between the two protocols. This includes the use of an HTTP-to-Cap proxy for translation between the protocols.

Q.3. Explain ITU Architecture for IoT

International Telecommunication Union-Telecommunication Sector View

1. The International Telecommunication Union (ITU-T) Telecommunication Sector has been actively involved in IoT standardization since 2005.
2. The ITU-T has established this activity to focus on standardization efforts for IoT. It was initially named Joint Coordination Activity on Network Aspects of Identification Systems (JCA-NID) before being renamed.
3. The ITU-T's IoT domain model encompasses physical devices that directly connect to a communication network or through gateway devices.
4. This connectivity enables the exchange of information between devices, services, and applications.

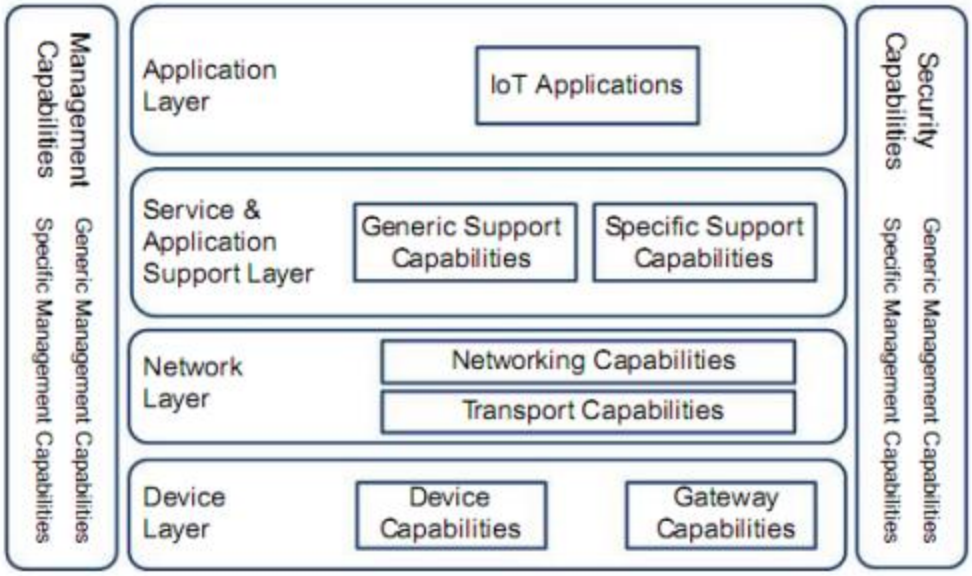


FIGURE 6.6

ITU-T IoT Reference Model.

(ITU-T 2013)

a. **Application Layer:** This layer within the ITU-T IoT model hosts specific IoT applications. For example, remote patient monitoring could be an application running at this layer.
b. **Service & Application Support Layer:** This layer consists of generic service capabilities utilized by all IoT applications, such as data processing and storage. It also includes specific service capabilities tailored to particular application domains, such as e-health or telematics.
c. **Network Layer:** The network layer provides networking capabilities for IoT. This includes functions like mobility management, authentication, authorization, accounting (AAA), and transport capabilities for IoT service data.
d. **Device Layer:** The device layer comprises both device capabilities and gateway capabilities.
    i. **Device Capabilities:** This encompasses the direct interaction of devices with the communication network, utilizing the capabilities provided by the network layer. It also includes ad hoc networking capabilities and low-power operation features that impact communication, such as the ability to sleep and wake up.
    ii. **Gateway Capabilities:** Gateways play a role in the IoT ecosystem by supporting multiple protocols and performing protocol conversions. They bridge the network layer capabilities with the communication capabilities of devices.

5. The ITU-T's standardization efforts aim to establish common frameworks, protocols, and interfaces for IoT, ensuring interoperability and seamless communication among devices, services, and applications.

Q.4. Explain Open Geospatial Consortium architecture

Open Geospatial Consortium Architecture

1. The Open Geospatial Consortium (OGC 2013) is an international industry consortium of a few hundred companies, government agencies, and universities that develops publicly available standards that provide geographical information support to the Web, and wireless and location- based services.
2. OGC includes, among other working groups,
    a. the Sensor Web Enablement (SWE) (OGC SWE 2013) domain working group, which develops standards for sensor system models (e. g. Sensor Model Language, or Sensor ML),
    b. sensor information models (Observations & Measurements , or O&M), and
    c. Sensor services that follow the Service-Oriented Architecture (SOA) paradigm, as is the case for all OGC-standardized services.

The functionality that is targeted by OGC SWE includes

• Discovery of sensor systems and observations that meet an application's criteria.
• Discovery of a sensor's capabilities and quality of measurements.
• Retrieval of real-time or time-series observations in standard encodings.
• Tasking of sensors to acquire observations.
• Subscription to, and publishing of, alerts to be issued by sensors or sensor services based upon certain criteria.

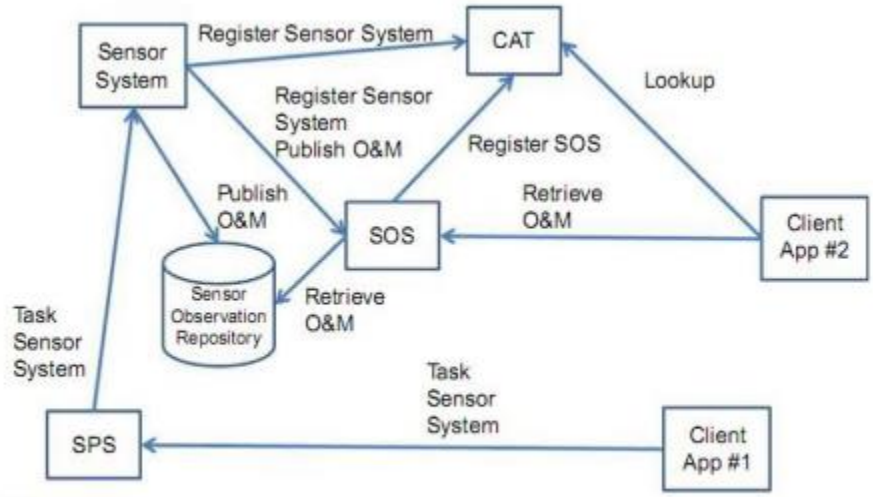


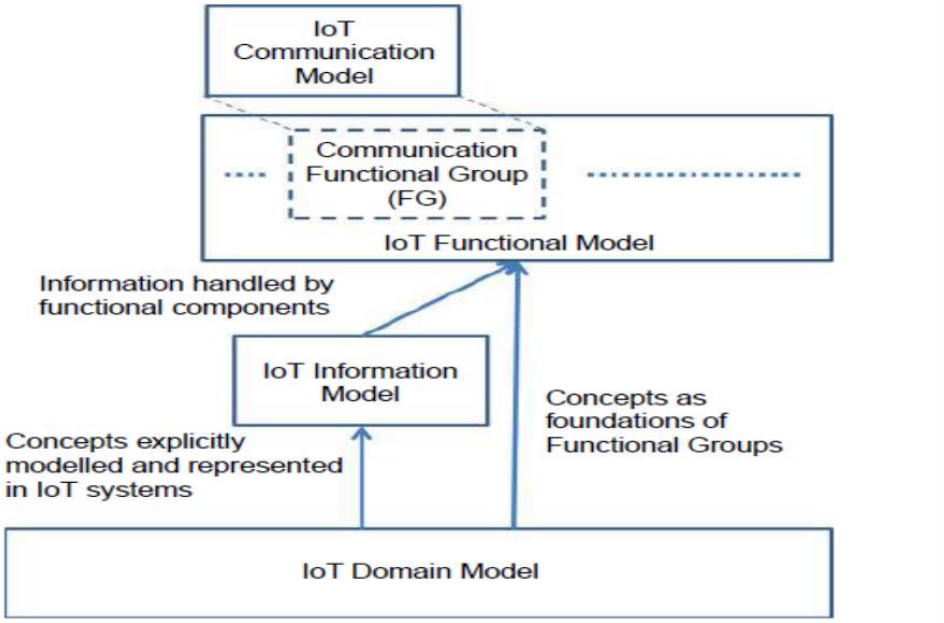
FIGURE 6.10
OGC functional architecture and interactions.

OGC SWE includes the following standards:

a. **Sensor and Transducer Model Language (TML):** These standards provide a model and XML schema for describing sensor and actuator systems and processes. Sensor enables the description of sensors and their capabilities, while TML focuses on transducers and their measurement processes.
b. **Observations and Measurements (O&M):** O&M is a model and XML schema for describing observations and measurements made by sensors. It allows for standardized representation and exchange of sensor data.

c. **SWE Common Data Model:** This standard defines a common data model for describing low-level data models, such as serialization in XML, in the messages exchanged between different functional entities within the OGC SWE framework.
d. **Sensor Observation Service (SOS):** SOS is a service that facilitates the requesting, filtering, and retrieval of observations and sensor system information. It acts as an intermediary between clients (requesting applications) and observation repositories or near real-time sensor channels.
e. **Sensor Planning Service (SPS):** SPS is a service that handles user-defined requests for sensor observations and measurements. It enables applications to specify the desired acquisition parameters and acts as an intermediary between the application and the sensor collection system.
f. **PUCK:** PUCK defines a protocol for retrieving sensor metadata from serial port (RS232) or Ethernet-enabled sensor devices. It provides a standardized approach for accessing sensor information.

Q.6. Explain IoT Reference Model

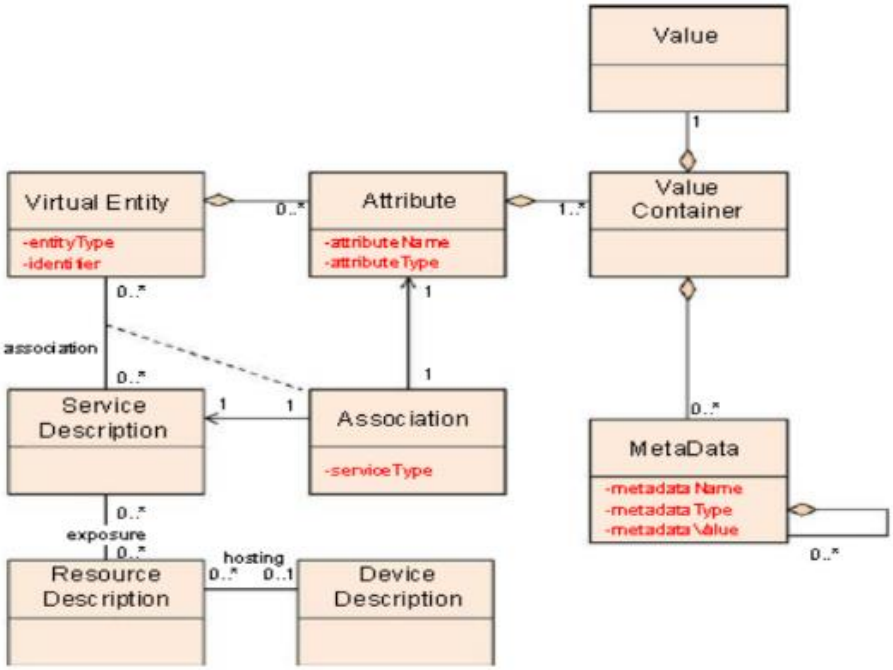


1. **Domain Model:**
    a. The domain model captures the basic attributes of the main concepts and the relationship between these concepts.
    b. A domain model also serves as a tool for human communication between people working in the domain in question and between people who work across different domains.
    c. The domain model is an important part of any reference model since it includes a definition of the main abstract concepts (abstractions), their responsibilities, and their relationships.
    d. Based on the IoT Domain Model, the IoT Information Model, Functional Model and Communication Model has been developed.
2. **Information Model:**

a. The Information Model is derived from the IoT domain model and focuses on capturing and processing information about the main entities and their interactions within the IoT system.
b. It defines the structure, attributes, and relationships of the information that flows within the system.
c. The Information Model helps in understanding the data requirements, data formats, and data flows within the IoT system.

3. **Functional Model:**
   a. The Functional Model describes the concepts and entities specific to the working system or application within the IoT domain.
   b. It identifies the functions, capabilities, and operations that the IoT system needs to perform to achieve its objectives.
   c. The Functional Model focuses on the behavior and functionality of the system, including how it interacts with the entities defined in the Information Model.
4. **Communication Model:**
   a. The Communication Model captures the communication interactions between the entities within an IoT system.
   b. It defines the protocols, messaging formats, communication patterns, and technologies used for communication between devices, sensors, gateways, and other components.
   c. The Communication Model ensures that the IoT system's entities can exchange information effectively and reliably.

Q.7. Explain IoT Functional Model
Q.8. Explain IoT Information Model



1. The IoT Information Model plays a crucial role in capturing the details of a **Virtual Entity-centric model** within the Internet of Things (IoT) domain. It enriches the data associated with Virtual Entities by providing the right context, allowing queries about who, what, where, and when to be answered.
2. Similar to the IoT Domain Model, the IoT Information Model is typically presented using **Unified Modeling Language (UML) diagrams.**
3. UML provides a standardized notation for modeling the structure, relationships, and behavior of the entities in the information model.
4. The IoT Information Model includes an association class, which contains specific information about the association between a Virtual Entity and a related Service.
5. This **association class** helps define the relationship between Virtual Entities and the services they interact with.
6. The IoT Information Model focuses on maintaining the necessary information about **Virtual Entities** and their properties or attributes.
7. These attributes can be either static or dynamic in nature. Static attributes describe the characteristics that do not change frequently, while dynamic attributes capture information that varies over time.
8. The information about these attributes can enter the system through various means, such as manual data entry or reading data from sensors attached to the Virtual Entity.
9. The attributes of Virtual Entities in the IoT Information Model can have one or more values annotated with **meta-information or metadata.**
10. Metadata provides additional information about the attributes, such as their data types, units of measurement, allowed value ranges, or semantic meaning.
11. Metadata plays a crucial role in understanding and interpreting the data associated with Virtual Entities.
12. By capturing the details of Virtual Entities and their attributes, annotated with metadata, the IoT Information Model provides a comprehensive representation of the information flow and structure within the IoT system.
13. It facilitates understanding, communication, and effective utilization of data within the IoT ecosystem.

Q.9. Describe safety, privacy, security and trust model of IoT

1. **Safety:**
    a. System safety in IoT is highly dependent on the specific application or domain.
    b. In cases where IoT systems involve actuators that can potentially harm humans or animals, ensuring safety is crucial.
    c. While not all hazards or mitigation steps may involve IoT technology directly, system designers can incorporate safety assertions at relevant points in the interaction between users, services, resources, and devices to mitigate risks.
2. **Privacy:**
    a. Protecting user privacy is of utmost importance in IoT systems, particularly due to the involvement of human interactions with the physical world.

b. The IoT-A Privacy Model emphasizes functional components such as identity management, authentication, authorization, and trust and reputation to safeguard user privacy and ensure that personal information is handled securely.

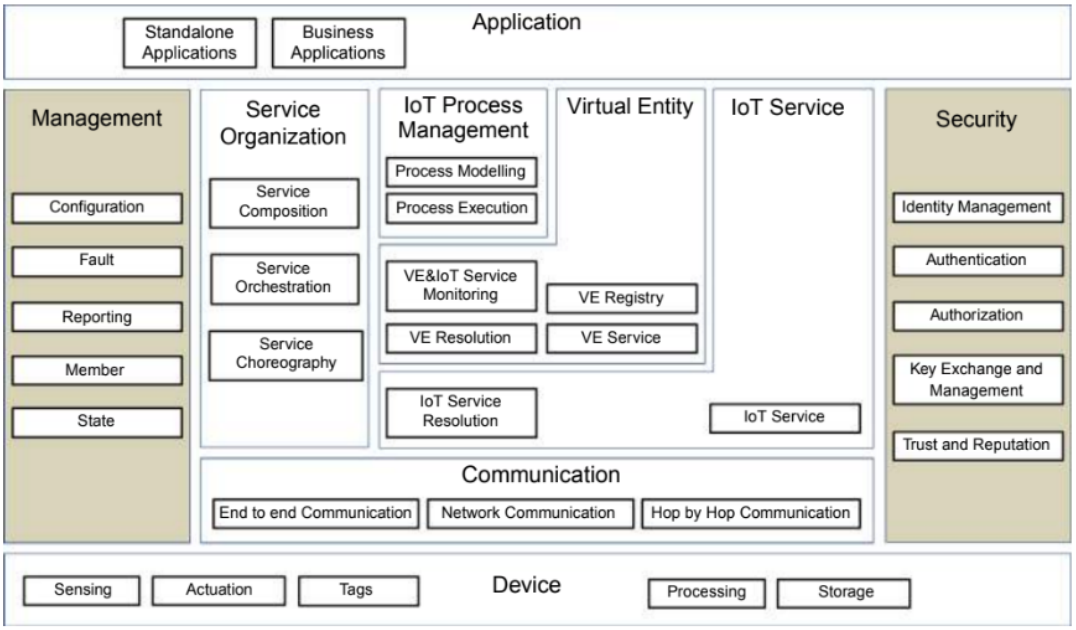
3. **Trust:**
    a. Trust models in IoT systems represent the dependencies and expectations among interacting entities.
    b. Trust models are essential for establishing reliable relationships and enabling secure interactions.
    c. IoT-A identifies trust model domains, trust evaluation mechanisms, trust behavior policies, trust anchors, and federation of trust as necessary aspects of a trust model within IoT systems.
4. **Security:**
    a. The security model for IoT focuses on ensuring communication security and protecting the confidentiality and integrity of interacting entities.
    b. It encompasses various functional components such as identity management, authentication, authorization, and trust and reputation.
    c. Communication security measures, such as encryption and secure protocols, are implemented to prevent unauthorized access, data breaches, and tampering.

Overall, safety, privacy, trust, and security are interconnected and critical aspects of designing and operating IoT systems. By addressing these aspects appropriately, IoT systems can be made more reliable, protect user privacy, establish trust relationships, and ensure the secure exchange of data and information.

Q.10. Explain Functional view, Information view and Deployment operation view of IoT Reference architecture

a. The Reference Architecture is a starting point for generating concrete architectures and actual systems.
b. A Reference Architecture serves as a guide for one or more concrete system architects.
c. The concept of views for the presentation of an architecture in IoT Reference Architecture are useful for reducing the complexity of the Reference Architecture blueprints by addressing groups of concerns one group at a time.
    i. Functional View: Description of what the system does, and its main functions.
    ii. Information View: Description of the data and information that the system handles.
    iii. Deployment and Operational View: Description of the main real world components of the system such as devices, network routers, servers, etc.
d. <u>Functional view:</u>
    i. The functional view for the IoT Reference Architecture is adapted from Iota-A (Cares et al. 2013).

ii. It consists of the Functional Groups (FGs) presented in the IoT Functional Model, each of which includes a set of Functional Components (FCs).



iii. Communication functional group: The Communication FG contains following components:

1. Hop-by-Hop Communication: The hop-by-hop FC is responsible for transmission and reception of physical and MAC layer frames to/from other devices. This FC has two main interfaces: (a) one “southbound” to/from the actual radio on the device, and (b) one “northbound” to/from the Network FC in the Communication FG.
2. Network Communication: This FC is responsible for message routing & forwarding and the necessary translations of various identifiers and addresses.
3. End-to-End Communication: This FC is responsible for end-to-end transport of application layer messages through diverse network and MAC/PHY layers.

iv. Service Organization FG:

1. The Service Organization FG acts as a coordinator between different Services offered by the system.
2. It consists of the following Functional Components:
    a. The Service Composition FC manages the descriptions and execution environment of complex services consisting of simpler dependent services.
    b. The Service Orchestration FC resolves the requests coming from IoT Process Execution FC or User into the concrete IoT services that fulfill the requirements.

c. The Service Choreography FC is a broker for facilitating communication among Services using the Publish/Subscribe pattern.

v. Security FG:
   1. The Security FG contains the necessary functions for ensuring the security and privacy of an IoT system.
   2. It consists of the following FCs:
      a. The Identity Management FC manages the different identities of the involved Services or Users in an IoT system in order to achieve anonymity by the use of multiple pseudonyms.
      b. The Authentication FC verifies the identity of a User and creates an assertion upon successful verification.
      c. The Key Exchange & Management is used for setting up the necessary security keys between two communicating entities in an IoT system.
      d. The Trust & Reputation FC manages reputation scores of different interacting entities in an IoT system and calculates the service trust levels.

vi. Management FG:
   1. The Management FG contains system-wide management functions that may use individual FC management interfaces.
   2. It consists of the following FCs:
      a. The Configuration FC maintains the configuration of the FCs and the Devices in an IoT system.
      b. The Fault FC detects, logs, isolates, and corrects system-wide faults if possible.
      c. The Member FC manages membership information about the relevant entities in an IoT system.

e. Information view:

i. The information view consists of:
   1. the description of the information handled in the IoT System,
   2. the way this information is handled in the system; in other words, the information lifecycle and flow (how information is created, processed, and deleted),
   3. the information handling components

ii. Information Description:

1. The pieces of information handled by an IoT system complying to an ARM such as the Iota-A (Cares et al. 2013) are the following :

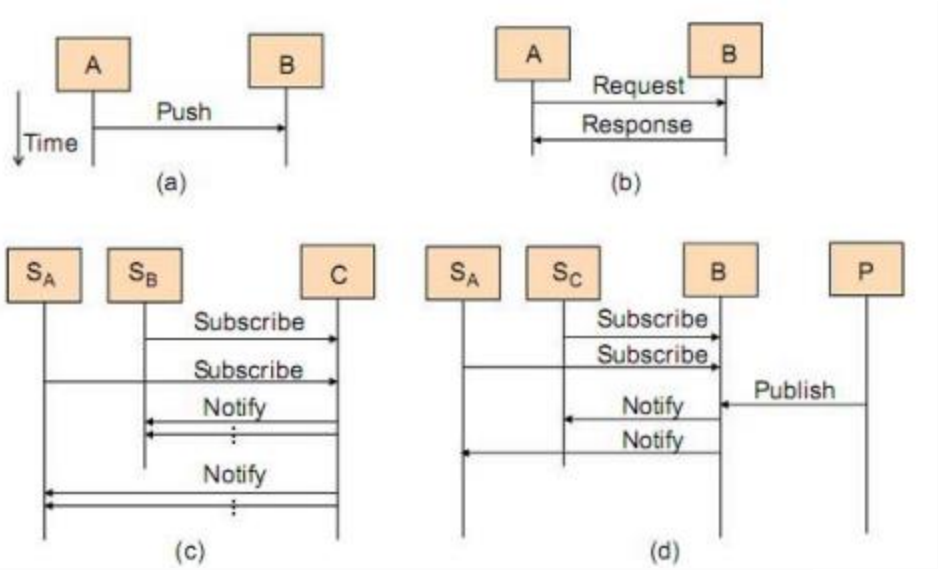
a. Virtual Entity context information, i.e. the attributes (simple or complex) as represented by parts of the IoT Information model (attributes that have values and metadata such as the temperature of a room). This is one of the most important pieces of information that should be captured by an IoT system, and represents the properties of the associated Physical Entities or Things.
b. IoT Service output itself is another important part of information generated by an IoT system. For example , this is the information generated by interrogating a Sensor or a Tag Service.
c. Virtual Entity descriptions in general, which contain not only the attributes coming from IoT Devices (e.g. ownership information)

iii. Information flow and lifecycle:

1. The flow of information in an IoT system follows two main directions:
    a. From devices that produce information such as sensors and tags, information follows a context-enrichment process until it reaches the consumer application or part of the larger system,
    b. From the application or part of a larger system information, it follows a context reduction process until it reaches the consumer types of devices (e.g. actuators).

iv. Information handling:

1. An IoT system is typically deployed to monitor and control Physical Entities.
2. Mainly the Devices, Communication, IoT Services, and Virtual Entity FGs in turn perform monitoring and controlling Physical Entities in the functional view.
3. Certain FCs of these FGs, as well as the rest of the FGs, play a supporting role for the main FGs in the Reference Architecture, and therefore in the flow of information.
4. The presentation of information handling in an IoT system assumes that FCs exchange and process information.
5. The exchange of information between FCs follows the interaction patterns below.



6. Push: An FC A pushes the information to another FC B if the contact information of component B is already configured in component A, and component B listens for such information pushes.
7. Request/Response: An FC A sends a request to another FC B and receives a response from B after A serves the request.
8. Subscribe/Notify: Multiple subscriber components (S, S) can subscribe for information to a component C, and C will notify the relevant subscribers when the requested information is ready.

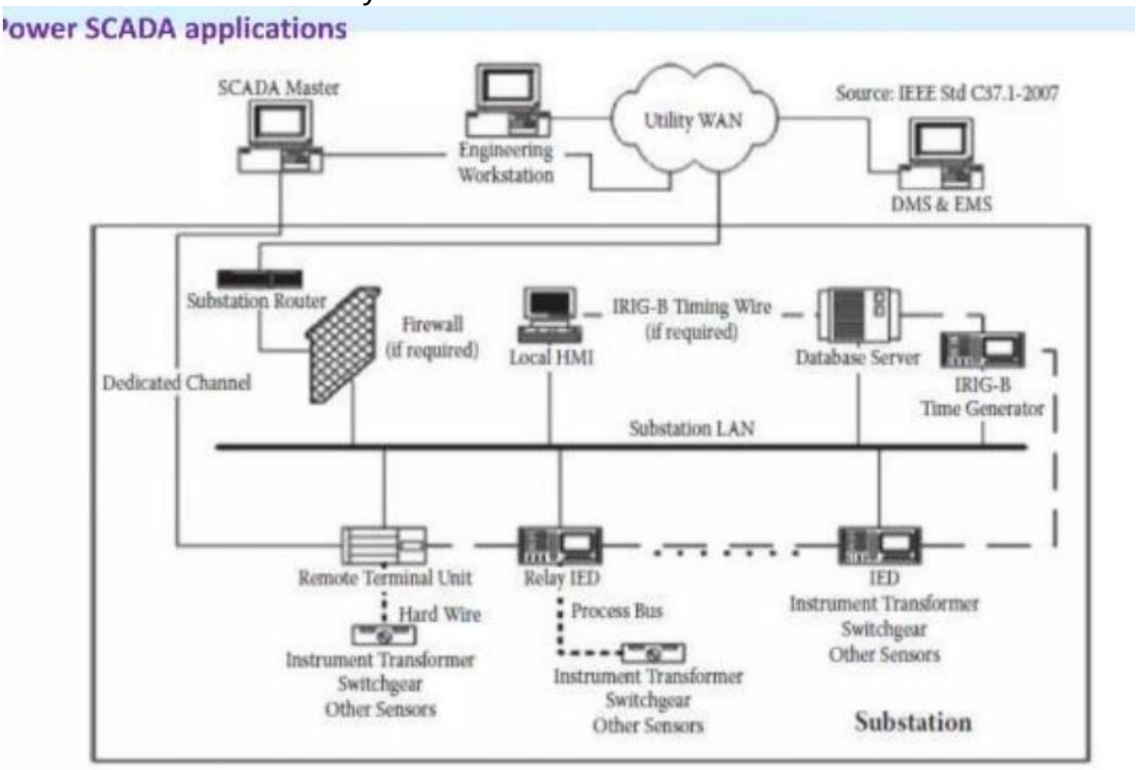
f. Deployment and operational view:

The Deployment and Operational View depends on the specific actual use case and requirements.

Unit 3

Q.1. Explain SCADA protocol

1. SCADA (Supervisory Control and Data Acquisition) is an essential component of the industrial automation arena. It represents a pillar of the IoT ecosystem, particularly in the context of network-based industrial automation and the use of intelligent electronic devices (IEDs) or IoT devices.
2. The IEEE Stud C37.1TM, established in 2007, is a standard specification for SCADA and automation systems.



3. It defines the SCADA architecture and provides guidelines for its implementation. This standard addresses various levels of SCADA systems, including the technologies used and their interactions within the architecture.

4. In recent years, with the advent of IoT and M2M communication, the processing capabilities in industrial automation have become distributed.
5. Functions that were traditionally performed at the control center can now be executed by IEDs, allowing for direct communication and interaction between devices.
6. However, despite the shift in functionality to IEDs, utilities still require a master station or an IoT platform to manage and operate the power system effectively.
7. This master station serves as a control center and receives data from substations or other entities within the electric industry structure, such as GENCO, TRANSCO, DISCO, ISO, and RTO.
8. While the IEEE Stud C37.1TM provides comprehensive guidelines for SCADA systems, it does not specify XML data formats or componentized architecture details.
9. Consequently, SCADA has been considered a traditional control system market, with professionals in the field often unaware of internet-based IT innovations and the concept of IoT.
10. To bridge this gap and leverage the benefits of IoT, it is important for individuals working in the SCADA domain to stay updated on IT advancements and understand how they relate to their work.
11. By embracing new concepts such as IoT, SCADA systems can evolve and integrate with modern technologies, leading to improved efficiency, scalability, and interoperability in industrial automation.

Q.2. Write Short notes on

1. MODBUS protocol

a. Modbus is a serial communications protocol that was originally published by Modicum, now Schneider Electric, in 1979. It has become a commonly used protocol for connecting industrial electronic devices. The reasons for its widespread use in industrial environments are:
   i. Developed for industrial applications: Modbus was specifically designed to meet the requirements of industrial automation and control systems.
   ii. Openly published and royalty-free: The Modbus protocol specifications are openly available, allowing anyone to implement it without any licensing fees.
   iii. Easy to deploy and maintain: Modbus is relatively simple to implement, making it easier for industrial devices to communicate with each other. Its simplicity also contributes to easier troubleshooting and maintenance.
   iv. Enables communication among many devices: Modbus allows multiple devices to be connected to the same network and communicate with each other, facilitating data exchange and coordination in industrial systems.

b. Modbus supports different versions and variations of the protocol, including:

i. **Modbus RTU:** This version uses binary encoding for communication over serial lines, typically using RS-232 or RS-485 interfaces.
ii. **Modbus ASCII:** Similar to Modbus RTU, but with ASCII encoding for communication, allowing for easier human readability but slower transmission speeds.
iii. **Modbus TCP/IP or Modbus TCP:** This version uses the TCP/IP protocol for communication over Ethernet networks, providing fast and reliable data transfer.
iv. **Modbus over TCP/IP or Modbus over TCP or Modbus RTU/IP:** This variant encapsulates Modbus RTU frames within TCP/IP packets, enabling Modbus communication over Ethernet networks.
v. **Modbus over UDP:** This variant uses the UDP protocol for communication, offering a lightweight and connectionless mode of transport.
vi. **Modbus Plus (Modbus+, MB+, or MBP):** A proprietary variant of Modbus designed for high-speed communication and distributed control systems.
vii. **Pemex Modbus:** A specific implementation of Modbus used by Pemex, Mexico's state-owned petroleum company.
viii. **Enron Modbus:** A variant of Modbus used by Enron, an energy company that ceased operations in 2001.

c. These different versions and variations of Modbus provide flexibility in choosing the appropriate communication method based on the specific requirements of industrial systems and their networking infrastructure.

2. BACNET

a. Banat (Building Automation and Control network) is a communications protocol specifically designed for building automation and control systems.
b. It provides a standardized mechanism for computerized building automation devices to exchange information and communicate with each other.
c. Banat is widely used in various applications within buildings, including HVAC (Heating, Ventilating, and Air Conditioning) control, lighting control, access control, and fire detection systems.
d. The Banat protocol defines a set of services that facilitate communication between building devices. These services include:
    i. Who-Is and I-Am used for device discovery, allowing devices to identify themselves on the network.
    ii. Who-Has and I-Have: Used for object discovery, enabling devices to query and announce the presence of specific objects or properties.
    iii. Read-Property and Write-Property: Services for sharing data between devices, where properties of objects can be read or modified.

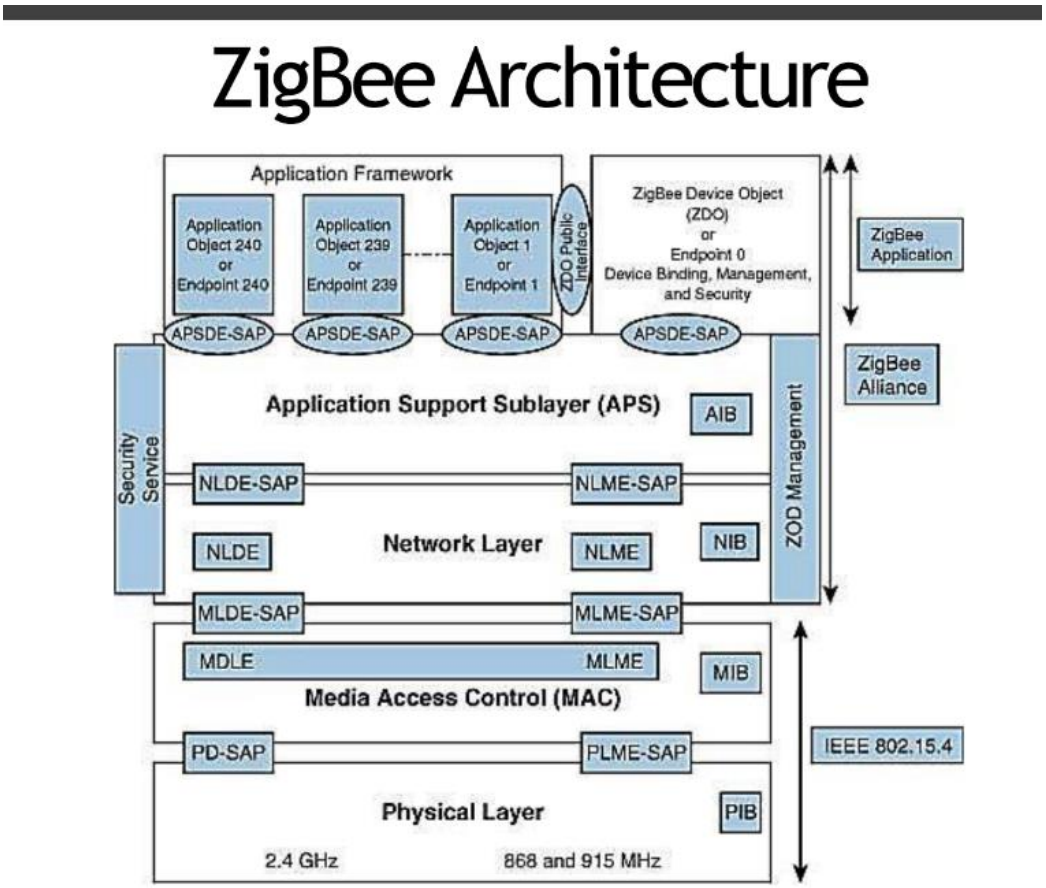
e. Banat also defines a comprehensive set of 60 object types that represent various aspects of a building automation and control system.
f. These object types include controllers, sensors, actuators, schedules, alarms, and more.
g. The protocol specifies how these objects can be acted upon by the defined services.
h. In terms of physical and data link layers, Banat supports multiple protocols, including:
    i. **ARCNET:** A local area network protocol that uses token passing for communication.
    ii. **Ethernet:** A widely used networking standard that allows devices to communicate over local and wide area networks.
    iii. **Banat/IP:** Banat over Internet Protocol, which allows Banat communication over IP networks.
    iv. **Banat/IPv6:** Similar to Banat/IP, but specifically designed for IPv6 networks.
    v. **Point-To-Point over RS-232:** Serial communication protocol that enables direct communication between two devices.
    vi. **Master-Slave/Token-Passing over RS-485:** A multi-drop serial communication protocol that supports communication between multiple devices on the same network.
    vii. **ZigBee:** A low-power wireless communication protocol used for short-range device-to-device communication.
    viii. **Lon Talk:** A protocol used in building automation systems, similar to Banat that allows devices to communicate over various media types, including twisted pair, power lines, and wireless.
i. These different physical and data link layers supported by Banat provide flexibility in terms of network infrastructure and enable devices to communicate seamlessly within building automation systems.

3. ZigBee architecture

a. ZigBee is an IEEE 802.15.4-based specification that defines a suite of high-level communication protocols for creating personal area networks with small, low-power digital radios.
b. It is designed for applications that require low power, low-bandwidth communication, making it suitable for various domains such as home automation and medical device data collection.
c. The ZigBee architecture is divided into three main sections:
    i. **IEEE 802.15.4:** This section includes the Media Access Control (MAC) and Physical (PHY) layers of the ZigBee protocol stack. The IEEE 802.15.4 standard provides the foundation for low-rate wireless personal area networks and defines the specifications for the physical transmission of data and the medium access control mechanism.

ii. **ZigBee Layers:** The ZigBee layers build upon the IEEE 802.15.4 standard and consist of the network layer, the ZigBee Device Object (ZDO), the application sublayer, and security management.
    1. Network Layer: The network layer is responsible for establishing and maintaining communication between ZigBee devices. It handles tasks such as device discovery, network formation, routing, and addressing.
    2. ZigBee Device Object (ZDO): The ZDO provides management and control functions for the ZigBee network. It handles tasks such as device initialization, network management, and service discovery.
    3. Application Sublayer: The application sublayer enables the implementation of various ZigBee-based applications. It defines the application framework and interfaces for developing specific functionality on top of the ZigBee network layer.
    4. Security Management: ZigBee provides built-in security mechanisms to ensure secure communication between devices. The security management layer handles tasks such as key establishment, authentication, and encryption.

iii. **Manufacturer Application:** Manufacturers of ZigBee devices can utilize the ZigBee application profile provided by the ZigBee Alliance or develop their own application profile tailored to their specific needs. The manufacturer application layer defines the functionality and behavior specific to a particular application domain.



d. Overall, the ZigBee architecture enables the creation of low power, low-bandwidth personal area networks and provides a standardized framework for communication, device management, and security in ZigBee-based applications.

Q.3. what are issues related to IoT Protocol standardization

1. Standardization in the IoT field, like in any other domain, has its own challenges and considerations. Here are some issues commonly associated with IoT standardization:
    a. **Fragmentation:**
        i. The IoT ecosystem is highly diverse, consisting of numerous technologies, protocols, and applications.
        ii. This fragmentation can make it challenging to achieve widespread interoperability and seamless integration between different IoT devices and platforms.
    b. **Lack of unified standards:**
        i. The absence of universally accepted standards for IoT poses challenges in terms of compatibility, security, and scalability.
        ii. Different organizations and industry players may develop their own proprietary standards, leading to fragmentation and interoperability issues.
    c. **Pace of innovation:**

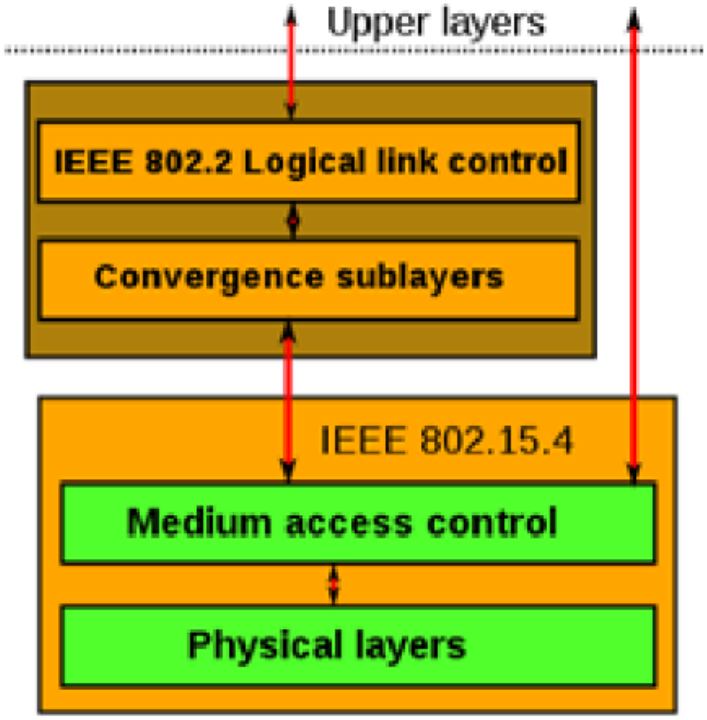
i. IoT is a rapidly evolving field, with new technologies, devices, and applications constantly emerging.
ii. Standardization processes may struggle to keep up with the pace of innovation, potentially hindering the adoption of new and transformative IoT solutions.

d. **Complex ecosystem:**
   i. IoT involves a wide range of stakeholders, including device manufacturers, software developers, service providers, and regulatory bodies.
   ii. Coordinating the efforts of these diverse stakeholders and aligning their interests to establish common standards can be complex and time-consuming.

e. **Balancing openness and proprietary solutions:**
   i. The balance between open standards and proprietary solutions is a constant challenge in IoT standardization.
   ii. Open standards foster interoperability and collaboration, while proprietary solutions can drive innovation and differentiation.
   iii. Striking the right balance is essential to ensure both interoperability and the incentive for innovation.

Q.4. Explain IEEE 802.15.4 standard

1. IEEE 802.15.4 is a set of protocols that define low-rate wireless personal area networks (LR-WPANs).
2. IEEE 802.15.4 focuses on LR-WPANs, which are networks characterized by low power, low-cost devices that communicate over short distances.
3. LR-WPANs are commonly used in IoT applications where multiple sensor nodes need to communicate with each other.

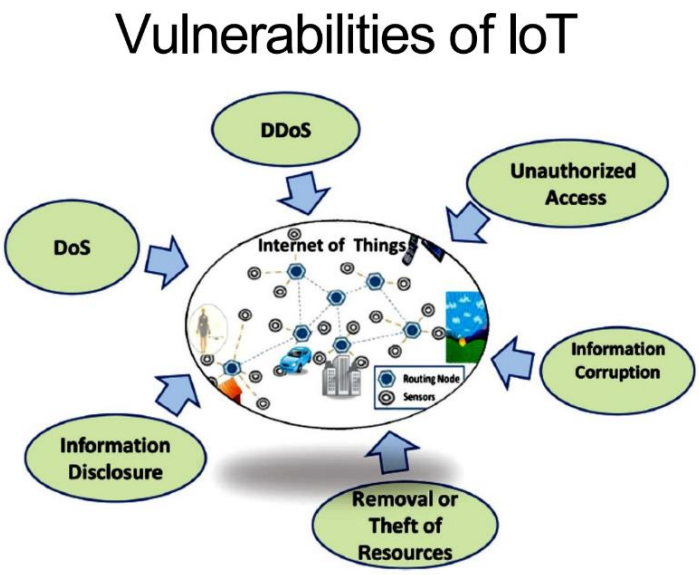


   a. **Physical Layer (PHY):** The PHY layer in IEEE 802.15.4 provides the data transmission service and interfaces with the physical layer management entity. It defines the modulation, data rates, and channel access mechanisms for LR-WPAN communication.
   b. **Media Access Control (MAC):** The MAC layer in IEEE 802.15.4 enables the transmission of MAC frames over the physical channel. It defines the medium access control mechanisms, including channel access methods, addressing, and frame structure.
4. The IEEE 802.15.4 standard was defined in 2003 and has undergone subsequent revisions and amendments. It provides a standardized framework for LR-WPAN communication, ensuring interoperability and compatibility between devices from different manufacturers.
5. <u>Uses of IEEE 802.15.4:</u>
   a. **IoT Applications:** IEEE 802.15.4 is well-suited for IoT applications where multiple sensor nodes need to communicate in a low-power and cost-effective manner. LR-WPANs based on IEEE 802.15.4 can support a large number of devices deployed together, making it scalable for IoT deployments.
   b. **Network Maintenance:** LR-WPANs based on IEEE 802.15.4 typically have low-cost and reliable network maintenance. The protocol's design and features help optimize power consumption, making it suitable for battery-powered devices that require long battery life and low maintenance.

6. Overall, IEEE 802.15.4 provides a standardized framework for LR-WPAN communication, offering an efficient and reliable solution for IoT applications where low-power, low-cost, and short-range wireless communication is required.

Q.5. What are the vulnerabilities in IoT security?

1. The vulnerabilities of IoT pose significant security risks. Here are some common vulnerabilities and examples of real-world attacks:



   a. **Unauthorized Access:** Unauthorized individuals gaining access to IoT resources can lead to tampering or misuse of devices. Identity-based verification and secure access control mechanisms are necessary to prevent unauthorized access.
   b. **Information Corruption:** IoT device credentials and data must be protected from tampering. Secure designs for access rights, credential management, and data exchange are crucial to prevent information corruption.
   c. **Theft of Resources:** Insecure communication channels can enable theft of shared resources, leading to man-in-the-middle attacks. Proper encryption and authentication mechanisms should be implemented to mitigate this vulnerability.
   d. **Information Disclosure:** Distributed data in IoT systems must be protected from disclosure. Context-aware access control mechanisms should be enforced to regulate access to sensitive information and prevent unauthorized disclosure.
   e. **Denial of Service (DoS) Attack:** DoS attacks attempt to prevent legitimate users from accessing services by overwhelming the network with excessive requests. Unauthorized users flood the network, causing disruption. Robust network infrastructure and defense mechanisms are essential to mitigate DoS attacks.

f. **Distributed Denial of Service (Dodos) Attack:** Dodos attacks involve multiple compromised systems targeting a single system, causing a denial of service. Compromised systems, controlled by the attacker, overwhelm the target system. These attacks can disrupt entire networks and cause widespread impact.

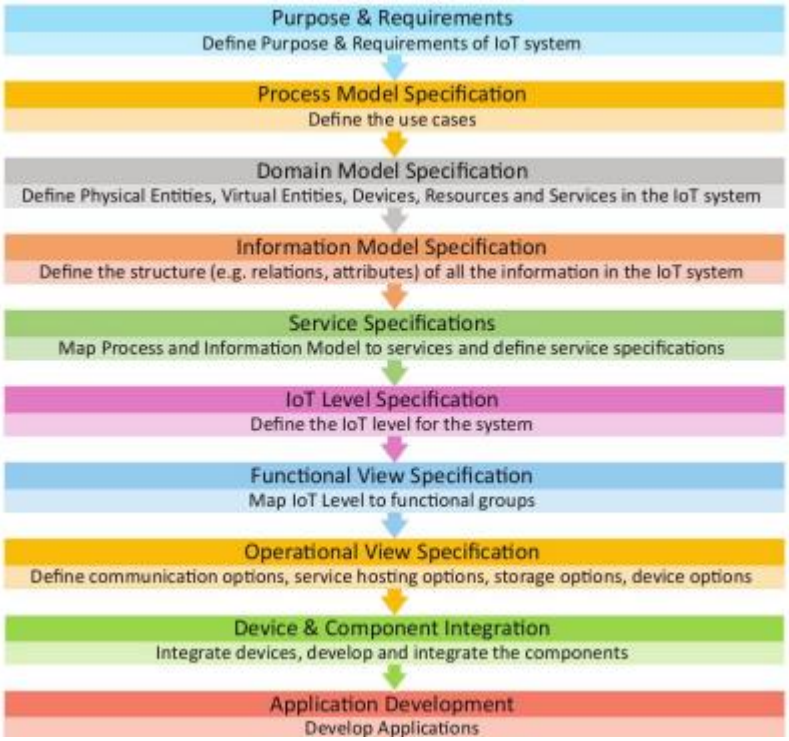
2. Real-world Examples:
    a. Carnal Botnet: Over 420,000 IoT devices, including routers and modems, were compromised to form a botnet.
    b. TREND net Camera Hack: Connected cameras from TREND net were hacked, and the live camera feeds were published online.
    c. Linux.Darlloz: This Iota worm infected around 100,000 systems, including TVs, routers, and even a fridge, displaying the vulnerability of connected devices.
3. These examples highlight the real risks posed by IoT vulnerabilities and the importance of implementing robust security measures to protect IoT systems and data.

Q.6. what are the key elements of IoT security

1. The key elements of security in an IoT environment include:
    a. **Authentication:** Establishing the identity of communicating devices or entities to ensure that the origin of electronic documents and messages is correctly identified.
    b. **Access Control:** Determining who should be allowed to access what resources. It prevents unauthorized use of resources by enforcing authentication and regulating access rights.
    c. **Data and Message Security:** Ensuring the authenticity, integrity, and confidentiality of data and messages. This includes mechanisms for source authenticity verification, detection of message modification, and protection of sensitive information.
    d. **Non-repudiation:** Preventing entities from denying their involvement in a communication or transaction. It ensures that senders cannot deny sending a message and recipients cannot deny receiving it.
    e. **Availability:** Ensuring that resources and services are accessible and operational when needed. This involves maintaining hardware, implementing backup systems, and protecting against malicious actions like Denial of Service (Do's) attacks.
    f. **Privacy:** Protecting the privacy of individuals and their personal data. This includes safeguarding sensitive information, implementing privacy-preserving techniques, and preventing unauthorized disclosure of identity or location information.
2. By addressing these key elements, IoT systems can enhance the security of their operations, protect user data and privacy, and ensure the reliable and trustworthy functioning of the ecosystem.

Unit 4

Q.1. Explain the steps involved used in IoT design methodology



- **Step 1: Purpose & Requirements Specification**

    1. The first step in IoT system design methodology is to define the purpose and requirements of the system.
    2. In this step, the system purpose, behavior and requirements (such as data collection requirements, data analysis requirements, system management requirements, data privacy and security requirements, user interface requirements,) are captured.

- **Step 2: Process Specification**

    1. The second step in the IoT design methodology is to define the process specification.
    2. In this step, the use cases of the IoT system are formally described based on and derived from the purpose and requirement specifications.

- **Step 3: Domain Model Specification**

    1. The third step in the IoT design methodology is to define the Domain Model.
    2. The domain model describes the main concepts, entities and objects in the domain of IoT systems to be designed.

3. Domain model defines the attributes of the objects and relationships between objects.
4. Domain model provides an abstract representation of the concepts, objects and entities in the IoT domain, independent of any specific technology or platform.
5. With the domain model, the IoT system designers can get an understanding of the IoT domain for which the system is to be designed.

- **Step 4: Information Model Specification**

1. The fourth step in the IoT design methodology is to define the Information Model.
2. Information Model defines the structure of all the information in the IoT system, for example, attributes of Virtual Entities, relations, etc.
3. Information model does not describe the specifics of how the information is represented or stored.
4. To define the information model, we first list the Virtual Entities defined in the Domain Model.
5. Information model adds more details to the Virtual Entities by defining their attributes and relations.

- **Step 5: Service Specifications**

1. The fifth step in the IoT design methodology is to define the service specifications.
2. Service specifications define the services in the IoT system, service types, service inputs/output, service endpoints, service schedules, service preconditions and service effects.

- **Step 6: IoT Level Specification**

1. The sixth step in the IoT design methodology is to define the IoT level for the system.

- **Step 7: Functional View Specification**

1. The seventh step in the IoT design methodology is to define the Functional View.
2. The Functional View (FV) defines the functions of the IoT systems grouped into various Functional Groups (FGs).
3. Each Functional Group either provides functionalities for interacting with instances of concepts defined in the Domain Model or provides information related to these concepts.

- **Step 8: Operational View Specification**

1. The eighth step in the IoT design methodology is to define the Operational View Specifications.
2. In this step, various options pertaining to the IoT system deployment and operation are defined, such as, service hosting options, storage options, device options, application hosting options, etc

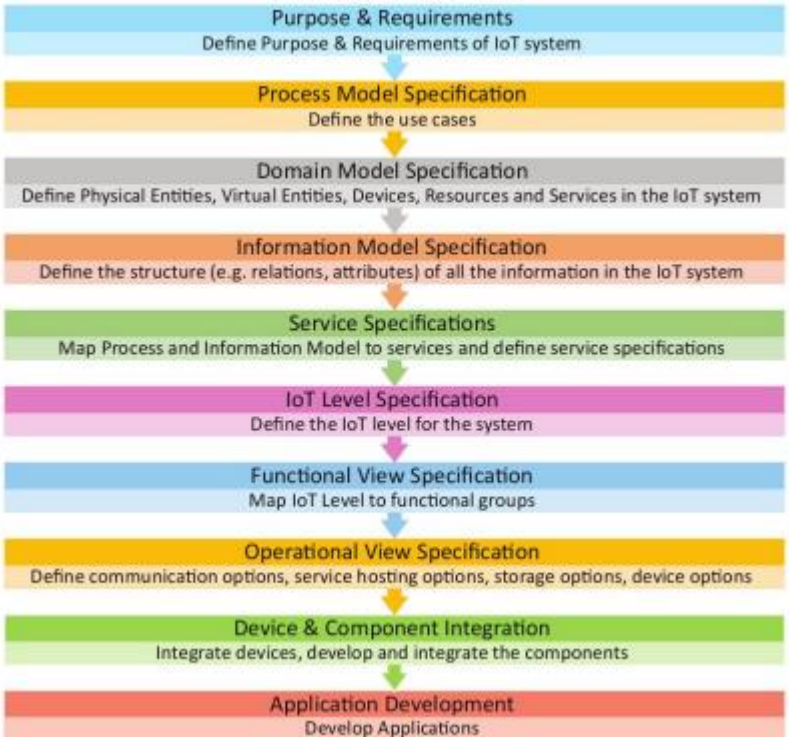
- **Step 9: Device & Component Integration**

1. The ninth step in the IoT design methodology is the integration of the devices and components.

- **Step 10: Application Development**

1. The final step in the IoT design methodology is to develop the IoT application.

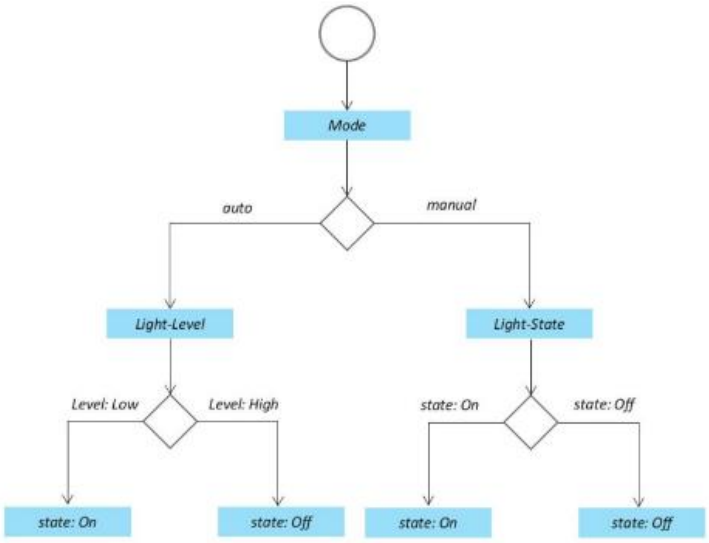
Q.2. Explain IoT design methodology using smart light application



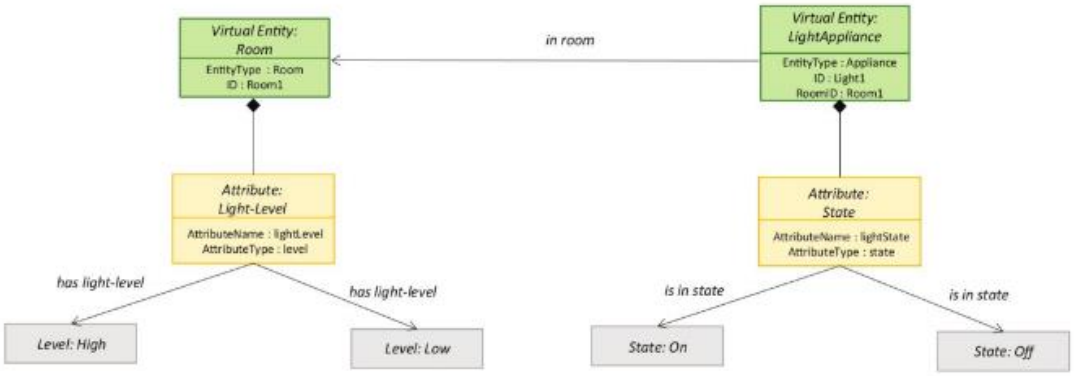
- **Step:1 - Purpose & Requirements**
  a. **Purpose:** A home automation system that allows controlling of the lights in a home remotely using a web application.
  b. **Behavior:** The home automation system should have auto and manual modes. In auto mode, the system measures the light level in the room and switches on the light when it gets dark. In manual mode, the system provides the option of manually and remotely switching on/off the light.

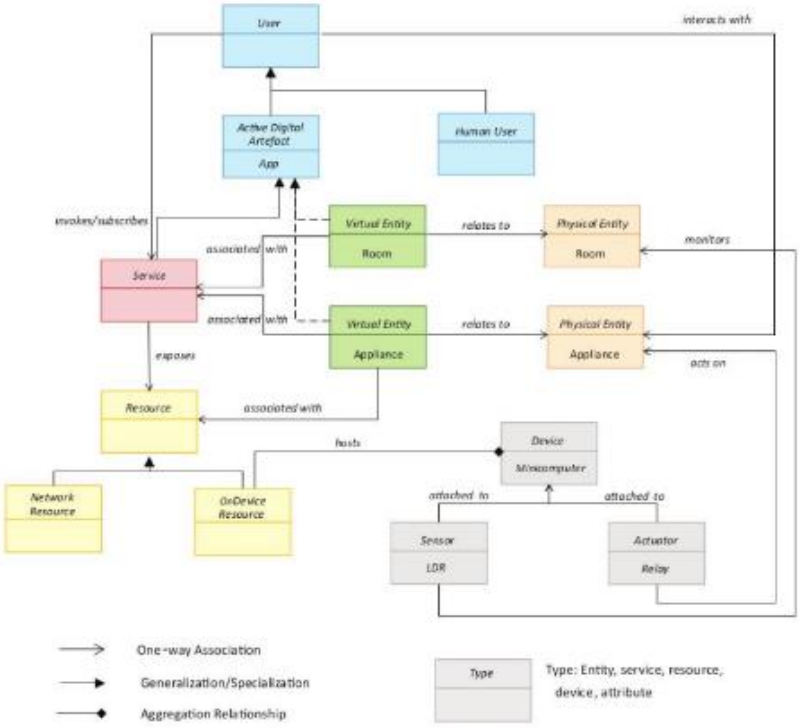
c. **System Management Requirement:** The system should provide remote monitoring and control functions.
d. **Data Analysis Requirement:** The system should perform local analysis of the data.
e. **Application Deployment Requirement:** The application should be deployed locally on the device, but should be accessible remotely.
f. **Security Requirement:** The system should have basic user authentication capability.

- **Step:2 - Process Specification**

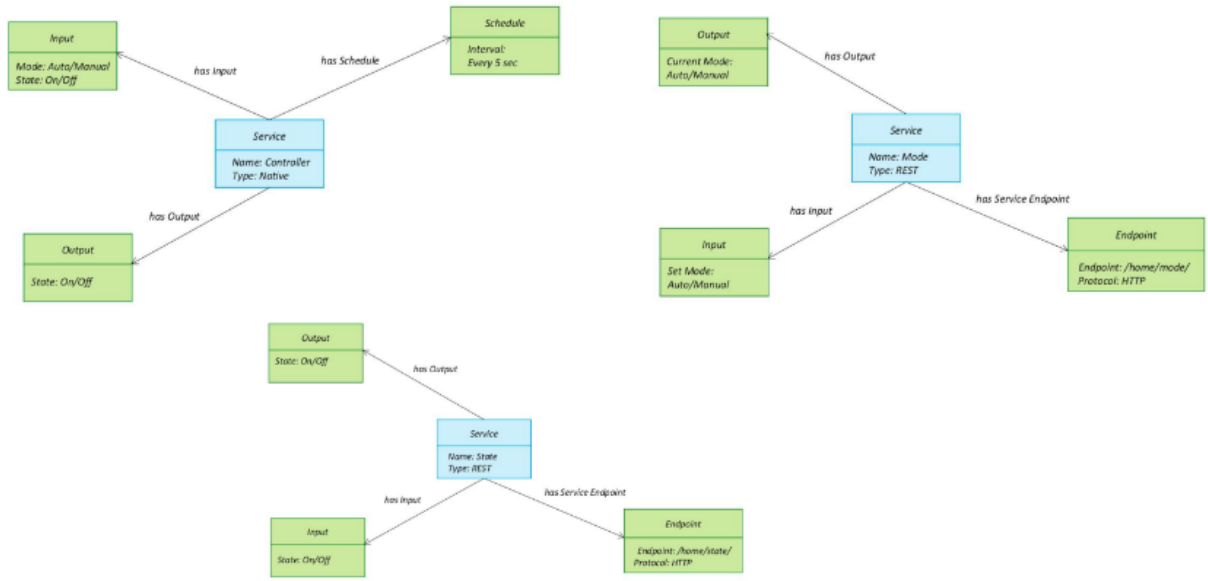


- **Step 3: Domain Model Specification**

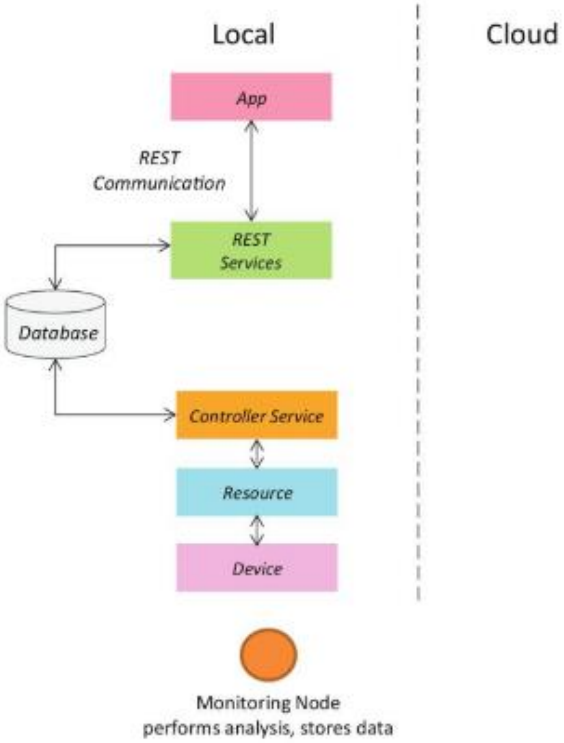




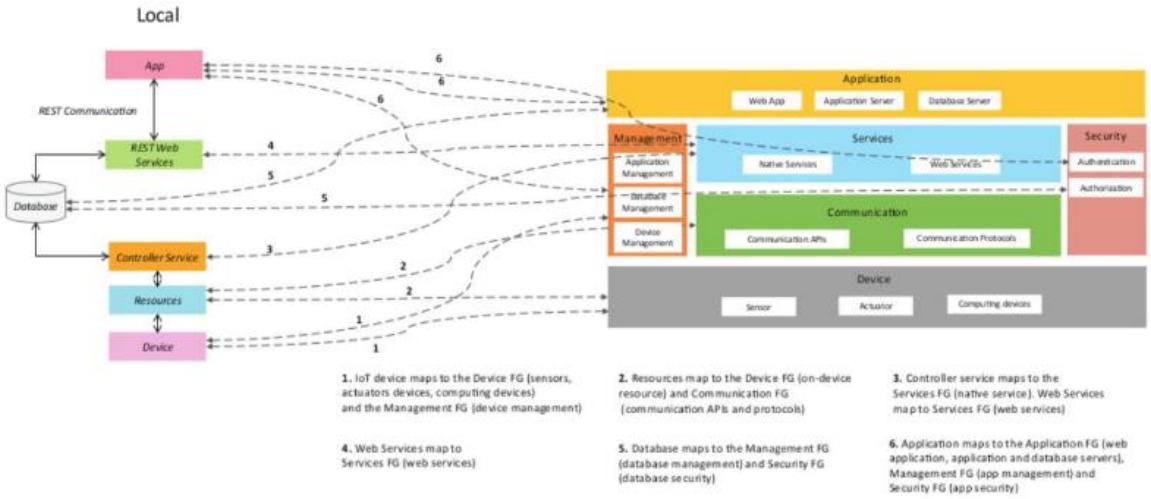
- **Step 4: Information Model Specification**
- **Step 5: Service Specifications**



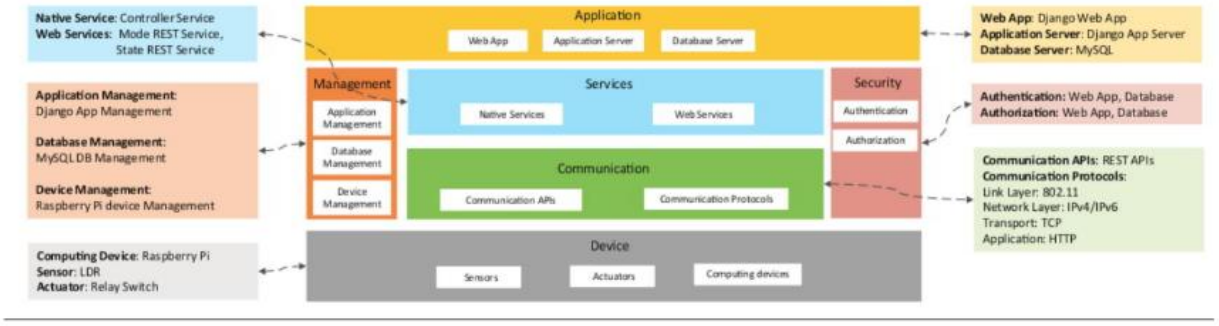
- **Step 6: IoT Level Specification**



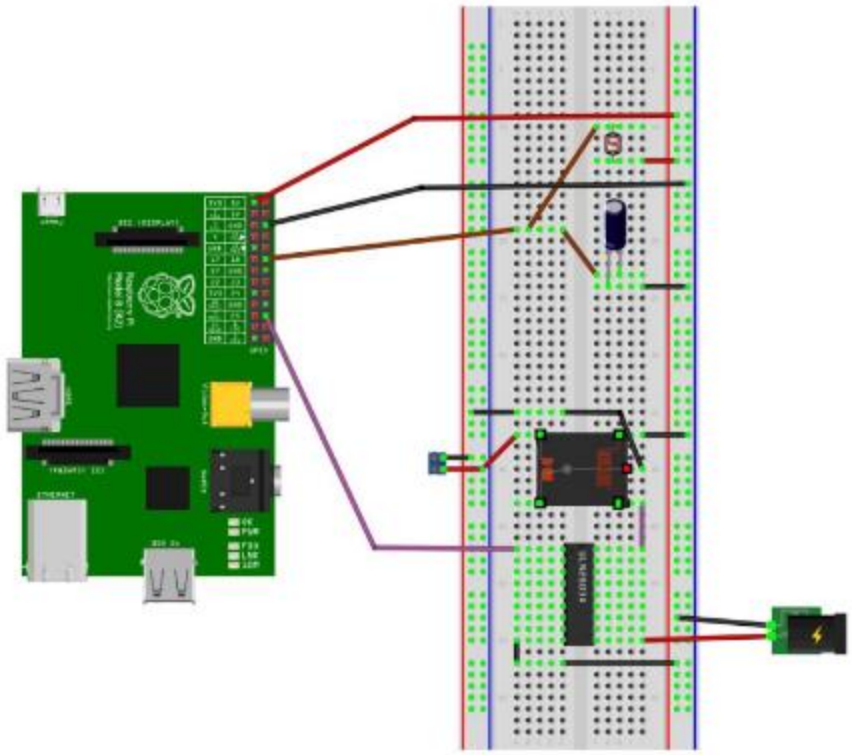
- **Step 7: Functional View Specification**



- **Step 8: Operational View Specification**



- **Step 9: Device & Component Integration**



- **Step 10: Application Development**
    1. **Auto-** Controls the light appliance automatically based on the lighting conditions in the room
    2. **Light-**
        a. When Auto mode is off, it is used for manually controlling the light appliance.
        b. When Auto mode is on, it reflects the current state of the light appliance.

Unit 5.

Q.1. Explain Iot domain application for Smart City, Smart Healthcare, Smart Factory, Smart Agriculture, Smart Environment

Smart City is a broad concept that encompasses the integration of IoT technologies to enhance the quality of life, sustainability, and efficiency within urban environments. Here are some IoT domain applications commonly used in Smart City initiatives:

1. **Smart Parking:**
    a. Smart parking makes the search for parking space easier and convenient for drivers.
    b. In smart parking, sensors are used for each parking slot, to detect whether the slot is occupied or not.
    c. This information is aggregated by local controllers and sent over the Internet to the database.
    d. Drivers can use an application to know about empty parking slots.
2. **Smart Lighting**
    a. Smart lighting systems for roads, parks, and buildings can help in saving energy.
    b. Smart lighting allows lighting to be dynamically controlled and adaptive to the ambient conditions.
    c. Smart lights connected to the Internet can be controlled remotely to configure lighting intensity and lighting schedule.
3. **Smart Roads:**
    a. Smart roads equipped with sensors can alert the users about poor driving conditions, traffic congestion, and accidents.
    b. Information sensed from the roads can be sent via the Internet to applications or social media.
    c. This helps in reducing traffic jams.
4. **Structural Health Monitoring**
    a. A network of sensors are used to monitor the vibration levels in the structures.
    b. Data from the sensors is analyzed to assess the health of the structures.
    c. By analyzing the data it is possible to detect cracks, locate damages to the structures and also calculate the remaining life of the structure.
5. **Surveillance**
    a. Surveillance of infrastructure, public transport and events in cities is required to ensure safety and security.
    b. City wide surveillance requires a large network of connected cameras.
    c. The video feeds from the cameras can be aggregated in cloud-based storage.
    d. Video analytics applications can be used to search for specific patterns in the collected feeds.
6. **Emergency Response**
    a. IoT systems can be used to monitor buildings, gas and water pipelines, public transport and power substations.

b. These systems provide alerts and help in mitigating disasters.
c. Along with cloud-based applications, IoT systems help to provide near real-time detection of adverse events.

Smart Healthcare

IoT technologies have significant potential to revolutionize the healthcare industry, enabling improved patient care, remote monitoring, and efficient healthcare management. Here are some IoT domain applications commonly used in Smart Healthcare:

1. **Health and Fitness Monitoring**
    a. With the advent of IoT, remote healthcare has become a viable option for attending to patients.
    b. There is no need for patients to visit hospitals for every minor health problem.
    c. The doctor can attend to such patients from a remote location.
    d. Different sensors can be fixed near the patient to monitor the health vitals of that patient.

2. **Wearable Electronics**
    a. Now-a-days there are different types of wearables available in the market to monitor health and lifestyles.
    b. Some examples of such wearables are smart watches, smart glasses, smart patches, smart garments, etc.

3. **Medication Management:**
    a. IoT can improve medication adherence and management.
    b. Smart pill dispensers and medication reminder systems send alerts to patients when it is time to take their medication.
    c. These devices can track and record medication consumption, helping patients and healthcare providers monitor adherence and adjust treatment plans accordingly.

4. **Health Data Analytics:**
    a. IoT-generated health data can be aggregated and analyzed to derive insights.
    b. Big data analytics and machine learning techniques can identify patterns, predict disease outbreaks, or generate personalized treatment recommendations.
    c. This can support evidence-based decision-making, improve patient outcomes, and optimize healthcare delivery.

Smart Factory

IoT technologies have been widely adopted in the manufacturing industry, leading to the concept of the "Smart Factory" or "Industrial IoT." These applications leverage connectivity, data analytics, and automation to optimize production processes, improve efficiency, and enable real-time monitoring. Here are some IoT domain applications commonly used in Smart Factories:

1. **Industrial Automation:**
    a. IoT enables the automation and control of various manufacturing processes.
    b. Connected sensors and actuators facilitate real-time monitoring and control of machines, robots, and production lines.
    c. This allows for efficient resource allocation, reduced downtime, and improved overall productivity.
2. **Supply Chain Optimization:**
    a. IoT-based supply chain management systems provide visibility and optimization across the entire supply chain.
    b. Sensors and connectivity enable real-time tracking of inventory, shipments, and logistics operations.
    c. This allows for efficient inventory management, reduced lead times, and improved coordination with suppliers and customers.
3. **Energy Management:**
    a. IoT enables energy monitoring and optimization within the factory.
    b. Connected sensors and energy meters track energy consumption and identify areas for improvement.
    c. Real-time analytics and control systems help optimize energy usage, reduce costs, and support sustainable manufacturing practices.
4. **Machine Diagnosis & Prognosis**
    a. The machines used in the industry can be fixed with sensors.
    b. The data from the sensors can be used to diagnose the machines.
    c. We can know if the machine is working up to the expected performance or not.
    d. The data analysis will also let the owner of the machine know when the life of the machine will be over.

Smart Agriculture

IoT technologies have significant potential in the field of agriculture, often referred to as "Smart Agriculture" or "Aggrotech." These applications leverage connectivity, sensors, and data analytics to optimize farming practices, increase crop yield, conserve resources, and enhance sustainability. Here are some IoT domain applications commonly used in Smart Agriculture:

1. **Smart Irrigation**
    a. Irrigation refers to the watering of plants.
    b. By using different sensors like temperature sensor, humidity sensor, soil moisture sensor, etc.

c. Data can be collected about the soil and the environment and let the farmer know when to turn on the water sprinklers to provide water to the plants.
d. This process is illustrated in the figure given below.

2. **GreenHouse Control**
    a. A greenhouse is an artificial field that can be grown inside buildings or on the roof tops.
    b. It is a controlled environment in which several types of sensors are fixed to gather data about the soil, environment and other parameters.
    c. The data from the green house is aggregated at a local gateway and sent to the server via the Internet.
    d. The data at the server is analyzed and appropriate alerts are sent to the owner of the green house.
3. **Livestock Monitoring:**
    a. IoT technologies can be used to monitor the health and well-being of livestock.
    b. Wearable devices or sensors attached to animals collect data on vital signs, behavior, and location.
    c. This helps farmers detect health issues, track animal movements, optimize feeding, and improve overall livestock management.

4. **Supply Chain Optimization:**
    a. IoT-based supply chain management systems improve the efficiency of agricultural supply chains.
    b. Sensors and connectivity enable real-time tracking of products, perishable goods, and storage conditions.
    c. This ensures timely delivery, reduces waste, and helps maintain product quality throughout the supply chain.
5. **Crop Monitoring and Management:**
    a. IoT sensors and imaging technologies enable real-time monitoring and management of crops.
    b. Drones equipped with cameras or multispectral sensors can capture images of fields, providing insights into crop health, growth patterns, and potential pest or disease outbreaks.
    c. This data assists in timely interventions and optimized crop management practices.

6. **Weather and Climate Monitoring:**
    a. IoT weather stations collect data on temperature, humidity, rainfall, wind speed, and solar radiation.
    b. This data helps farmers assess weather patterns, predict adverse conditions, and plan their farming activities accordingly.
    c. Accurate weather information supports decision-making related to planting, harvesting, and pest management.

Smart Environment

IoT technologies are instrumental in monitoring and managing the environment, enabling sustainable practices and conservation efforts. Here are some IoT domain applications commonly used in Smart Environment initiatives:

1. **Weather Monitoring**
    a. IoT-based weather monitoring systems use different sensors to gather data.
    b. That data is sent to the cloud-based storage.
    c. The collection can be analyzed and visualized with applications.
    d. Weather alerts can be subscribed by users from such applications.

2. **Air Pollution Monitoring**
    a. IoT-based air pollution monitoring systems can monitor harmful gas emissions by factories and vehicles using gaseous and meteorological sensors.
    b. The collected data can be analyzed to make decisions on pollution control approaches.

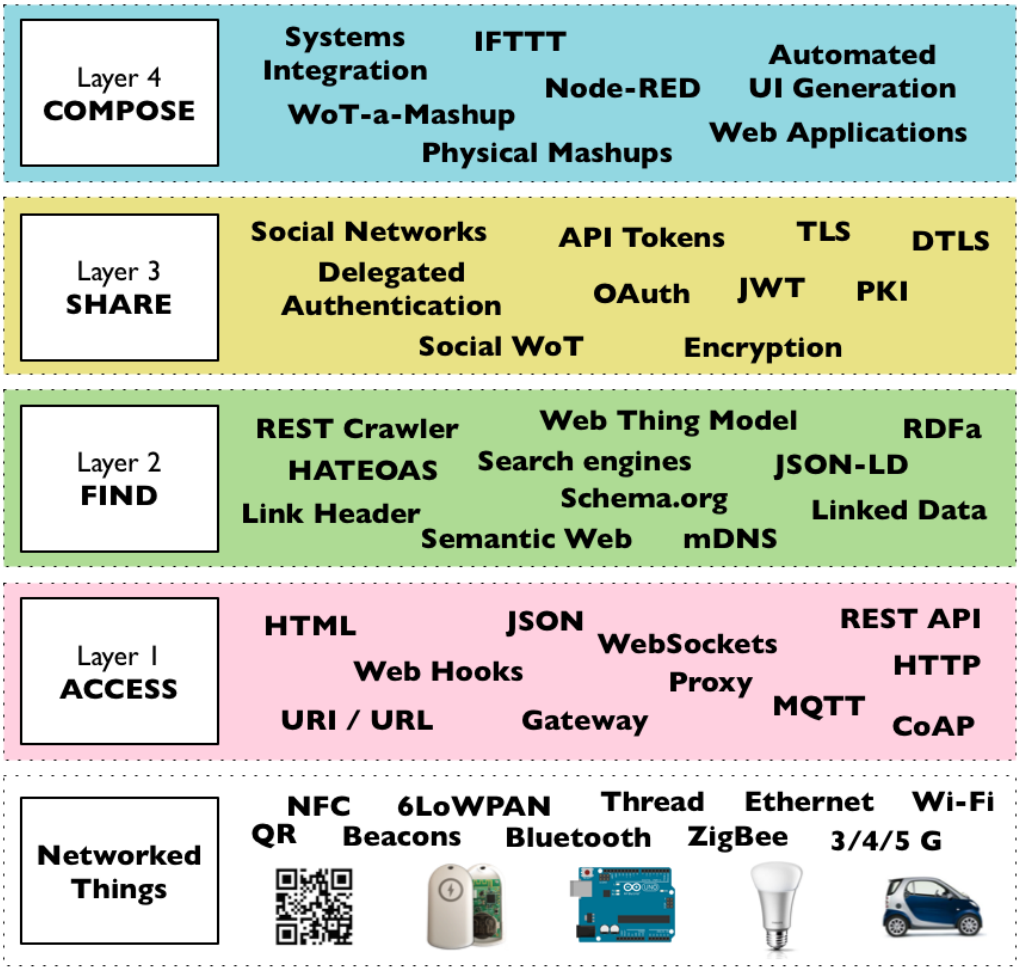
3. **Noise Pollution Monitoring**
    a. IoT-based noise pollution monitoring systems use a number of noise pollution monitoring systems that are deployed at different places in the city.
    b. The data on noise levels from the stations is collected on servers or in the cloud.
    c. The collected data can be analyzed to generate noise maps.

4. **Forest Fire Detection**
    a. IoT-based forest fire detection systems use a number of nodes deployed at various locations in the forest.
    b. Each monitoring node collects data about ambient conditions.
    c. This data will be collected and analyzed for the presence of fire and corresponding people will be alerted.
5. **River Floods Detection**
    a. IoT-based flood monitor systems use a number of sensor nodes to monitor the water level.
    b. Data from the sensors is aggregated on the server or in the cloud.
    c. Monitoring applications raise alerts in case of rapid increase in water level or when rapid flow rate is detected.

Unit 6

Q.1. Differentiate between Web of Things and cloud of things

Q.2. Explain Web of things architecture



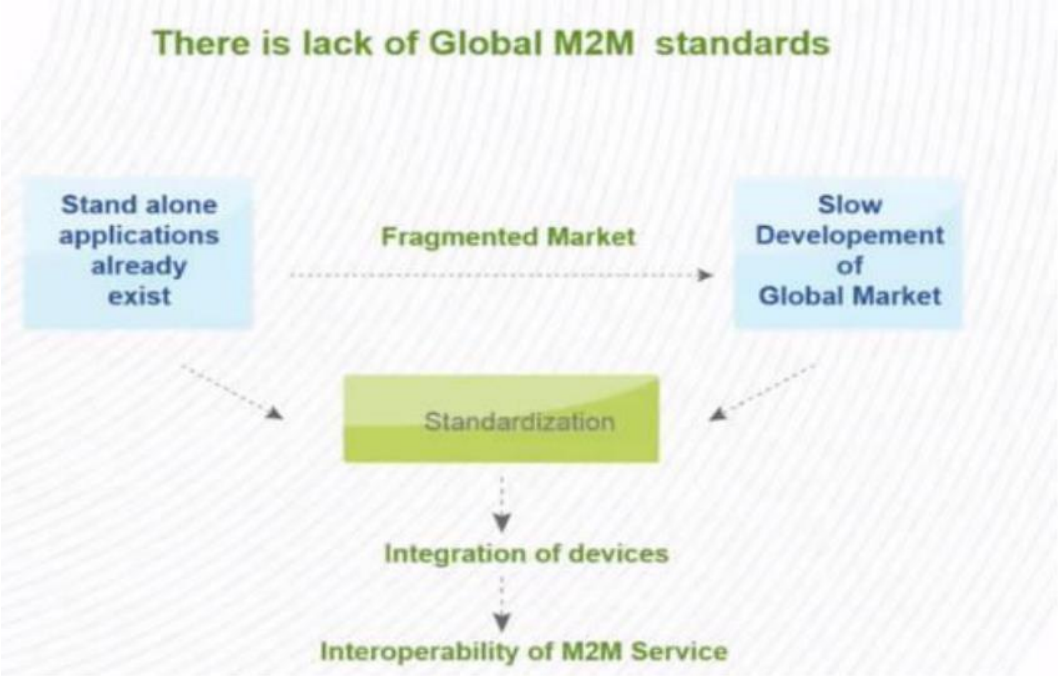
Source: Building the Web of Things: book.webofthings.io
Creative Commons Attribution 4.0

1. The Web of Things (WoT) architecture aims to enable the composition, sharing, finding, accessing, and networking of things in a standardized and interoperable manner.
2. Here's how these aspects are typically addressed within the WoT architecture:
    a. <u>Composition Layer:</u>
        i. The WoT allows the composition of multiple IoT devices and services to create higher-level functionalities.
        ii. This can be achieved through the use of Thing Descriptions, which provide a standardized way to describe the capabilities and interfaces of individual devices.
        iii. By understanding the capabilities of different devices, developers can create composite applications that leverage the functionalities of multiple things.

b. Sharing Layer:
    i. Sharing in the WoT refers to making IoT devices and their functionalities available to other users or applications.
    ii. Thing Descriptions play a crucial role here by providing a standardized format for describing the devices' capabilities, communication protocols, and interfaces.
    iii. By sharing Thing Descriptions, developers can enable others to discover and interact with their devices, fostering collaboration and reuse.
c. Finding Layer:
    i. Finding refers to the discovery of available IoT devices and services within the WoT ecosystem.
    ii. This is typically facilitated through a WoT Thing Directory, which acts as a central registry or directory of Web Things.
    iii. Developers or users can query the directory to discover and locate specific devices based on their metadata, capabilities, or other criteria.
    iv. The directory provides a mechanism for finding and accessing the desired things.
d. Access Layer:
    i. Accessing things in the WoT architecture involves interacting with IoT devices and their functionalities over the web.
    ii. The WoT Gateway plays a crucial role in providing access to the devices.
    iii. It acts as a bridge between the IoT protocols and web protocols, allowing communication and control over the web.
    iv. The gateway translates the protocols used by the devices into web-friendly protocols, enabling easy access and interaction with the devices through web-based applications or services.
e. Networking Things:
    i. Networking in the WoT architecture refers to the establishment of connections and communication between different IoT devices and services.
    ii. The WoT API provides standardized interfaces and protocols for accessing and controlling Web Things.
    iii. It enables devices to expose their properties, actions, and events, which can be consumed by other devices or services.
    iv. By leveraging the WoT API, devices can communicate and collaborate with each other within the WoT network.

Q 3 What is platform middleware of Web of things

1. Platform Middleware is also known as Application framework Or Three tiered Application Server
2. It provides natural fits for mapping the IoT objects to software objects.



3. European Telecommunications Standards Institute (ETSI) is working for M2M Standards
4. **These standards resolves**
    a. N/w Routing
    b. Synchronous Communication
    c. Subscribe Model
    d. Uniform data storage model
    e. Language Independent
5. **M2M Middleware Standards Key Elements**
    a. M2M Device
    b. M2M Area Network
    c. M2M Gateway
    d. M2M Communication Network
    e. M2M Application Server
6. **Platform Middleware of WoT for WSN**
    a. The Open Geospatial Consortium, Sensor Web Enablement (OGC SWE) is working for WSN Standardization.
    b. The goal of SWE is creation of web- based sensor networks to make all sensors and repositories of sensor data discoverable, accessible, and where applicable, controllable via the World Wide Web
    c. Enables
        i. Discovery of sensors, processes, and observations
        ii. Tasking of sensors or models
        iii. Access to observations and observation streams
        iv. Publish–subscribe capabilities for alerts
        v. Robust sensor system and process descriptions
    d. The following web service specifications have been produced by the OGC SWE Working Group

i. Sensor observation service—standard web interface for accessing observations
ii. Sensor planning service—standard web interface for tasking sensor systems and model and requesting acquisitions
iii. Sensor alert service—standard web interface for publishing and subscribing to sensor alerts
iv. Web notification service—standard web interface for asynchronous notification

e. The USN (Ubiquitous Sensor Networks) standardization is working for WSN Standardization.
    i. USN is a conceptual network or framework built over existing physical networks that makes use of sensed data and provide knowledge services
    ii. Main Components of USN:
        1. USN applications and services platform-technology framework to enable the effective use of a USN
        2. USN middleware-including functionalities for sensor network management and connectivity, event processing, sensor data mining.
        3. Network infrastructure- makes use of existing networks
        4. USN gateway- A node that interconnects sensor networks with other networks
        5. Sensor network- Network of interconnected sensor nodes

7. Platform Middleware of WoT for SCADA:

a. ANSI/ ISA-95 is working for SCADA Standardization.
b. It is used to specify a framework for the interoperability of a set of software products used in the manufacturing domain and to facilitate its integration into a manufacturing application.
c. The objectives of ISA-95 are to provide consistent terminology that is a foundation for supplier and manufacturer communication, to provide consistent information models, and to provide a consistent operations model as a foundation for clarifying application functionality and how information is to be used.
d. Main Components of SCADA:
    i. Middleware
    ii. Interaction
    iii. Communication
    iv. Context
    v. Secure Distributed Storage
    vi. Proactive knowledge base
    vii. Tools
e. Main Components of ISA-95 for Middleware platform of SCADA:
    i. Interaction- These manage interaction between Users and Smart Products
    ii. Communication- These provide support for the information exchange between smart products

iii. Context- These provide components for sensing, processing, and distributing context information
iv. Proactive knowledge base- components for handling the knowledge of a smart product
v. Secure distributed storage- components for storing knowledge of a smart product in a secure and distributed way
vi. Tools- tools for developing smart products, such as for automatically extracting relevant information from manuals, editors

8. **Platform Middleware of WoT for RFID:**
    a. EPCglobal is working for RFID Standardization.
    b. BRIDGE Building Radio- frequency Identification solutions for the Global Environment under EPCGlobal is also working for RFID standardization which manages the exchange of RFID and aggregated information between nodes.
    c. The Cross UBiQuitous Platform (CUBIQ) ..Japan aims to develop a common platform that facilitates the development of context- aware applications.
    d. Main Components of RFID
        i. Manage Devices
        ii. Collect and Integrate Data
        iii. Structure and Filter Data
        iv. Tag ID Association
    e. The CUBIQ architecture consists of three layers
        i. Mobile terminals with RFID tag readers collect RFID tag info and record location.
        ii. The mobile terminals are connected via the core CUBIQ infrastructure and share RFID tag information.
        iii. Observers can search RFID tag information to estimate the location of the target person.

Q.4. Explain Unified Multitier architecture of WoT

1. SOA/EAI (Service-Oriented Architecture/Enterprise Application Integration) versus SODA/MAI (Service-Oriented Device Architecture/Device Middleware Architecture):
    a. SOA and EAI are established principles and methodologies for designing and developing software as interoperable services, often over the Internet. They involve the use of metadata and protocols like SOAP (Simple Object Access Protocol) to enable communication and integration between different software systems.
    b. In the WOT context, SODA and MAI are proposed as extensions to SOA to enable devices to connect to the architecture. SODA focuses on device integration into an SOA, and MAI (Device Middleware Architecture) is a broader term that encompasses middleware technologies for connecting and managing devices in the IoT (Internet of Things) ecosystem.
2. Inheriting and enhancing existing data formats and protocols:

    a. WOT/IoT applications should build upon and improve existing data formats and protocols to ensure interoperability and compatibility with the existing ecosystem.
    b. For example, the use of XML-based message formats like SOAP is common in SOA, and these formats can be extended or enhanced to accommodate the specific requirements of WOT/IoT applications.
3. Metadata in WOT/IoT:
    a. Just like in SOA, metadata plays an essential role in WOT/IoT applications.
    b. Metadata provides descriptions and information about the services, devices, and their capabilities in the architecture.
    c. In the WOT context, metadata helps in discovering and understanding the available devices, their functionalities, and how they can be accessed and utilized.
4. DDL (Device Description Language) in SODA:
    a. SODA introduces a core standard called DDL, which is a device description language based on XML encodings.
    b. DDL categorizes devices into three main categories: sensors, actuators, and complex devices.
    c. By using DDL, devices can be described in a standardized manner, allowing for easier integration and interaction within the WOT architecture.
5. OSGi (Open Services Gateway initiative):
    a. OSGi is often referred to as the "Universal Middleware" because it provides a module system and service platform for the Java programming language.
    b. OSGi's modular architecture aligns well with the distributed and heterogeneous nature of WOT/IoT systems.
    c. It enables the development of scalable and adaptable applications by allowing new functionality to be added or updated without affecting the entire system.
    d. OSGi can be integrated into the Unified Multitier WOT Architecture as a middleware layer that manages the dynamic deployment and interaction of WOT/IoT components.
    e. It can provide capabilities such as service discovery, event-driven communication, security, and lifecycle management for bundles and services.

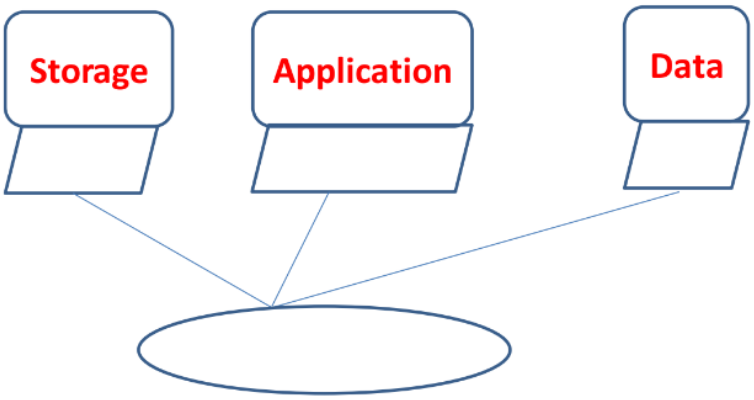
Q.5. Write short notes on Web portals and business intelligence

1. A web portal or links page is a website that functions as a point of access to information in the World Wide Web.
2. A portal presents information from diverse sources in a unified way.
3. Examples of public web portals include Yahoo, AOL, Excite, MSN
4. Apart from standard search engine feature, web portals offer other services such as e-mail, news, stock prices, information, databases and entertainment
5. Portals are categorized into-
    a. Horizontal Portals - cover many areas

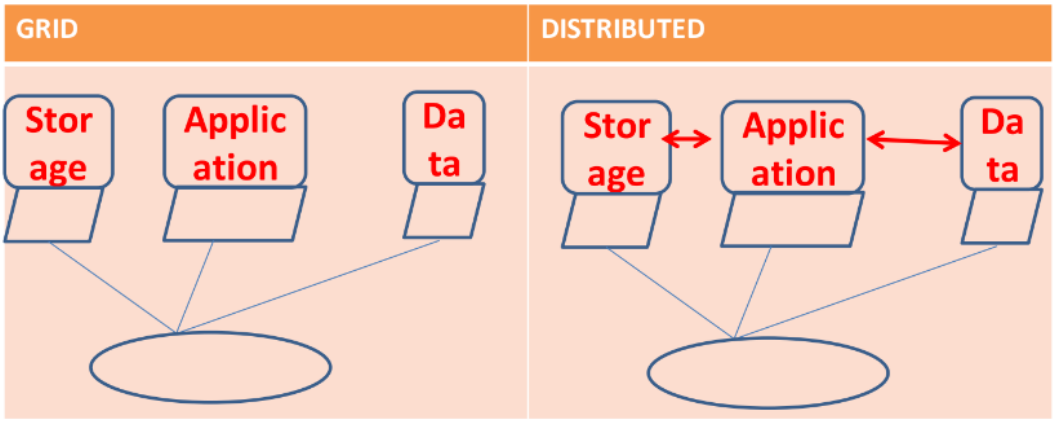
   b. Vertical Portals - focused on one functional area
6. WOT portals are vertical portals
7. When huge amount of data are collected in a IOT system, data mining can be conducted to acquire business intelligence (BI)
8. Data mining deals with finding patterns in data that are by user definition, interesting and valid.
9. Interdisciplinary area -databases, machine learning, pattern recognition, statistics, visualization, etc.
10. BI technologies provide historical, current, and predictive views of business operations.
11. Common functions of BI technologies are
    a. extract, transform, and load
    b. reporting, online analytical processing, analytics
    c. data mining, process mining, complex event processing
    d. business performance management, benchmarking, text mining, predictive analytics, and so on

Grid Computing

1. It is a collection of computing resources (like storage, processor, data , applications) from multiple locations to achieve a common goal.

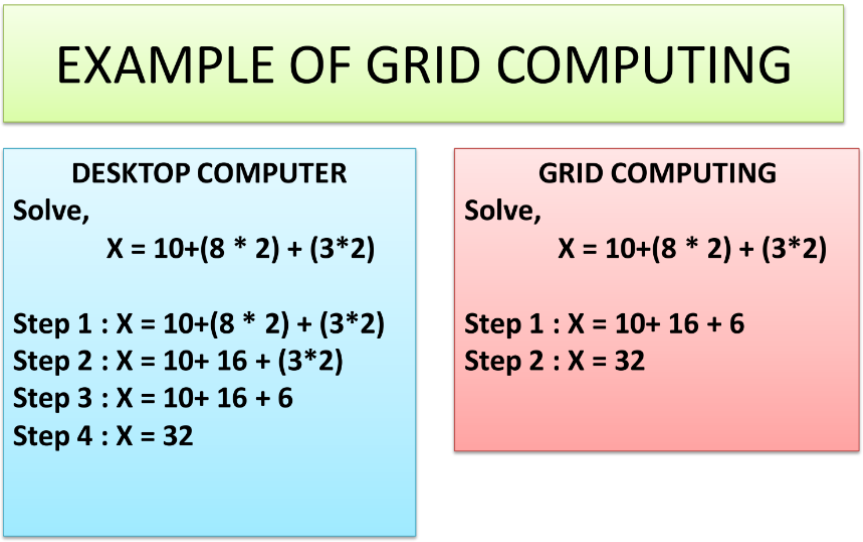


2. A grid is a distributed system with a non-interactive workload that involves a large number of files.



3. A computational grid is a h/w and s/w infrastructure that provides dependable, consistent & inexpensive access to high end computational capabilities.

4. The grid links together the computing resources and provides a mechanism to access them.
5. A grid computing system requires:
    a. At least one computer, usually a server which handles all the administrative duties for the system.
    b. A network of computers running special grid computing network software.
    c. A collection of computer softwares called middleware.



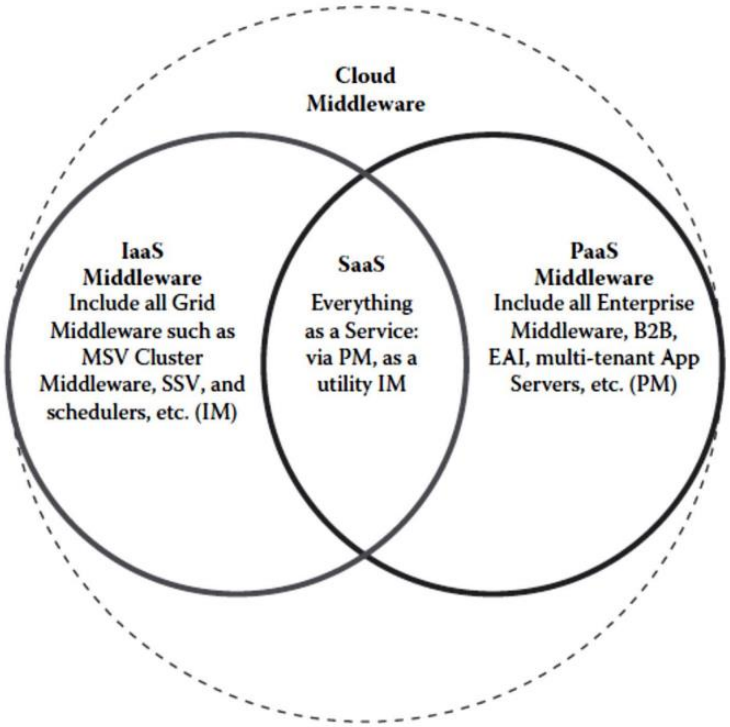
6. So Grid Computing saves time and increases processing speed.
7. <u>Disadvantages of Grid computing:</u>
    a. The users or project sponsors would have to bear the enormous cost of setting up and maintaining and monitoring the grid.
    b. Does not follow standard data communication protocol
8. <u>Applications of Grid Computing:</u>
    a. The European Organization for Nuclear Research (CERN) is one of the leading organizations running major grid computing initiatives including analyzing chemical compounds in the search for potential drugs for diseases such as avian flu.
    b. SETI (Search for Extraterrestrial Intelligence) @Home project is one of the earliest grid initiatives that downloads and analyzes data from radio telescopes. Participants simply need to download and run a program to join the grid network.

Q.6. What is cloud computing? What is cloud middleware?

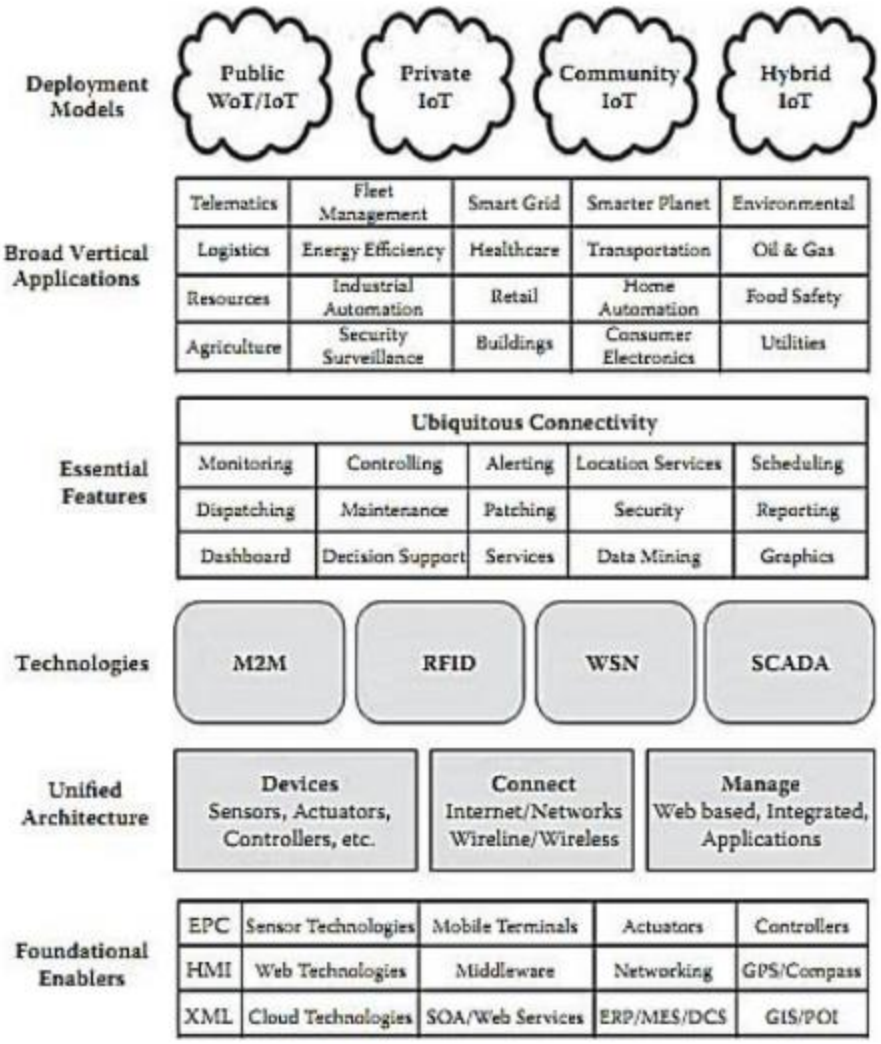
1. The cloud is a set of services and technologies that enable the delivery of computing services over the internet in real-time, allowing end-users instant access to data and applications from almost any device with internet access.
2. A cloud service has some distinct characteristics that differentiate it from traditional hosting.
    a. It is sold-on-demand , typically by the minute or the hours.
    b. It is elastic- a user can have as much or as little of a service as they want at any given time;

c. and the service is fully managed by the provider (the consumer needs nothing but a personal computer and internet access).

3. Benefits of Cloud Computing:
    i. Reduced Cost
    ii. Rapid Scalability
    iii. Highly Automated
    iv. Disaster Relief
    v. Flexibility
    vi. More Mobility
4. The cloud middleware consists of two kinds of middleware – IaaS and PaaS middleware
5. IaaS Middleware:
    a. **Network Management Middleware:** This middleware provides tools and services for managing and configuring network resources within the infrastructure. It includes functionalities like load balancing, virtual network setup, routing, firewall management, and network monitoring.
    b. **Grid Management Middleware:** Grid middleware helps manage distributed computing resources, such as clusters and grids, within an IaaS environment. It enables tasks like resource allocation, scheduling, and workload balancing across multiple nodes in the infrastructure.
    c. **Scheduling Middleware:** Scheduling middleware focuses on resource allocation and job scheduling in an IaaS environment. It ensures optimal utilization of resources by efficiently assigning tasks to available virtual machines or containers. Scheduling middleware may consider factors such as resource availability, workload characteristics, priorities, and policies.
6. PaaS Middleware:
    a. **Identity and Access Management (IAM) Middleware:** IAM middleware provides authentication, authorization, and user management services within a PaaS environment. It allows developers to secure their applications and manage user access, permissions, and roles. IAM middleware ensures that only authorized users can access and interact with the deployed applications.
    b. **Data Management Middleware:** Data management middleware in PaaS offers services and tools for storing, retrieving, and managing data within the platform. It includes functionalities like database management systems, data caching, replication, backup, and recovery. Data management middleware helps developers handle data-related tasks efficiently within their applications.
    c. **Business-level Solutions:** While not strictly categorized as middleware, PaaS platforms may provide additional business-level solutions as part of their offerings. These can include pre-built components, APIs, frameworks, and development tools for specific business domains or use cases. Examples could be solutions for e-commerce, customer relationship management (CRM), content management, or analytics.



Q.7. Explain Cloud of things architecture.



1. Deployment Models:
    a. **Public WOT/IoT:** In the public WOT/IoT deployment model, IoT devices and services are deployed on the public cloud infrastructure. It allows for easy accessibility, scalability, and cost-effectiveness but may raise concerns about data privacy and security.
    b. **Private IoT:** In the private IoT deployment model, the IoT infrastructure is hosted on a private cloud or on-premises within an organization's network. This model provides more control over data and security but may require higher upfront investment and maintenance costs.
    c. **Community IoT:** The community IoT deployment model involves a shared infrastructure among a group of organizations or individuals with common interests or objectives. It allows for collaboration, resource sharing, and cost-sharing while maintaining a certain level of control over data and security.

    d. **Hybrid IoT:** The hybrid IoT deployment model combines multiple deployment models, such as a mix of public, private, and community clouds. It provides flexibility by allowing organizations to choose the most suitable deployment option for different IoT applications or services.
2. Broad Vertical Applications:
    a. Broad vertical applications refer to the diverse range of domains where the CoT architecture can be applied.
    b. These include smart cities, industrial automation, healthcare, agriculture, transportation, energy management, and more.
    c. The CoT architecture can cater to various industry-specific requirements and use cases within these verticals.
3. Essential Features:
    a. Ubiquitous Connectivity: Ubiquitous connectivity is a key feature of the CoT architecture, enabling seamless communication and interaction between IoT devices, cloud services, and end-users.
    b. It ensures that devices can connect and transmit data reliably regardless of their location or network infrastructure.
4. Technologies:
    a. The CoT architecture incorporates various technologies to enable IoT connectivity, data management, and interoperability. Some common technologies used in CoT include:
        i. M2M (Machine-to-Machine): M2M technology enables direct communication between IoT devices without human intervention, allowing devices to exchange data and perform actions autonomously.
        ii. RFID (Radio Frequency Identification): RFID technology uses radio waves to identify and track objects equipped with RFID tags. It is widely used in supply chain management, asset tracking, and inventory management applications.
        iii. WSN (Wireless Sensor Networks): WSN technology involves a network of wireless sensors that collect and transmit data from the physical environment. WSNs are commonly used in environmental monitoring, smart agriculture, and structural health monitoring applications.
        iv. SCADA (Supervisory Control and Data Acquisition): SCADA systems are used to monitor and control industrial processes and infrastructure. They enable real-time data acquisition, control, and visualization of remote devices and systems.
5. Unified Architecture:
    a. The unified architecture of the CoT encompasses the integration of devices, connectivity, and management.
    b. It provides a cohesive framework for connecting IoT devices, managing their interactions, and integrating them into cloud-based services and applications.
    c. The unified architecture ensures interoperability, scalability, and ease of development and deployment of IoT solutions.

6. Foundational Enablers:
    a. EPC (Electronic Product Code): EPC is a standardized identification system for uniquely identifying physical objects in the IoT. It enables efficient inventory management, product tracking, and supply chain optimization.
    b. HMI (Human-Machine Interface): HMI refers to the interface between humans and machines. It allows users to interact with IoT devices and services, providing intuitive controls and visualizations.
    c. XML (Extensible Markup Language): XML is a markup language used for structuring and encoding data in a human-readable format. It is often used for data exchange and interoperability between different IoT systems and platforms.
7. These features, technologies, and enablers collectively contribute to the design and implementation.