IOT Notes - Internet of Things detailed and well explained answer easy to understand

Internet Of Things (University of Mumbai)

Internet of Things

What is IOT?

- IoT stands for the "Internet of Things," which refers to a network of interconnected devices that communicate with each other via the internet. These devices can include anything from smartphones and smart home appliances to sensors and industrial equipment.
- The Internet of Things is made up of a range of technologies that enable devices to communicate with each other and with central systems to collect and analyze data. These technologies include wireless networking, cloud computing, and data analytics.
- IoT is transforming many industries by enabling companies to collect vast amounts of data and analyze it in real-time, leading to more efficient and effective business operations.
- It is also driving innovation in areas such as healthcare, smart cities, and agriculture, among others.

What are the charateristics of IOT ?

1. Connectivity: IoT devices are connected to the internet or other networks, allowing them to communicate and share data with other devices and systems.
2. Sensors and Actuators: IoT devices are equipped with sensors to gather data from their environment and actuators to control devices based on that data.
3. Data Analytics: IoT devices generate a massive amount of data that can be analyzed to gain insights into the device's performance, environment, and user behavior.
4. Interoperability: IoT devices from different manufacturers and with different capabilities should be able to communicate and work together seamlessly.
5. Security: IoT devices must be secure, as they are collecting and sharing sensitive data. Strong encryption, authentication, and access controls are essential for ensuring the security of IoT devices and networks.
6. Scalability: IoT networks must be designed to handle a vast number of connected devices and data streams, as the number of connected devices is expected to grow exponentially in the coming years.
7. Real-time communication: IoT devices communicate in real-time, allowing for quick decision-making and response to events as they happen.
8. Automation: IoT devices can automate processes and tasks, reducing the need for human intervention and improving efficiency.
9. Remote control: IoT devices can be controlled remotely, allowing users to monitor and control them from anywhere with an internet connection.
10. Predictive maintenance: IoT devices can use data analytics to predict when maintenance is needed, reducing downtime and extending the lifespan of equipment.

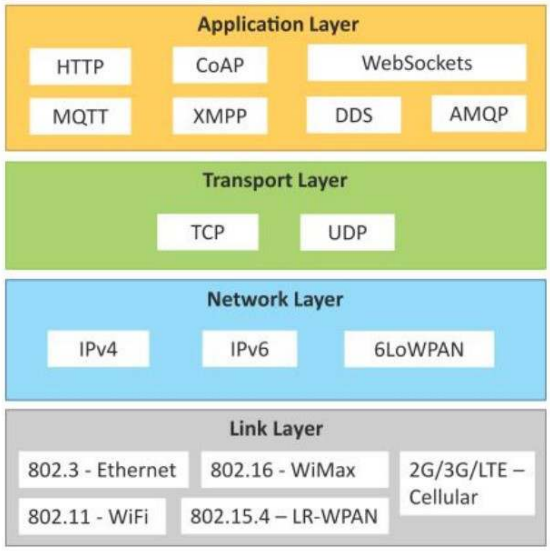
Explain Fog Computing and its characteristics?

- Fog computing is a type of distributed computing architecture that enables data processing to be moved closer to the edge of the network, closer to where the data is generated, rather than being sent to a central location (like the cloud) for processing.
- Fog computing can offer many benefits, such as reduced latency, improved security, and increased bandwidth efficiency, as data is processed locally and only relevant data is sent to the cloud for further analysis.
- It also enables organizations to make faster and more informed decisions in real-time, as data processing is done closer to the source.
- Examples of applications of fog computing include autonomous vehicles, smart factories, and healthcare monitoring systems, among others.

Characteristics of Fog Computing

1. Proximity to end-users: Fog computing enables data processing to be done at the edge of the network, closer to where the data is generated, which can reduce latency and improve performance.
2. Distributed architecture: Fog computing uses a distributed network of computing resources, such as routers, gateways, and edge devices, to process and analyze data.
3. Real-time processing: Fog computing enables data processing and analysis in real-time, which can support mission-critical applications that require immediate action.
4. Security: Fog computing can provide better security and privacy compared to cloud computing by processing sensitive data locally and reducing the risk of data breaches.
5. Scalability: Fog computing enables the addition of new computing resources to the network, allowing for increased scalability as the number of IoT devices and applications grow.

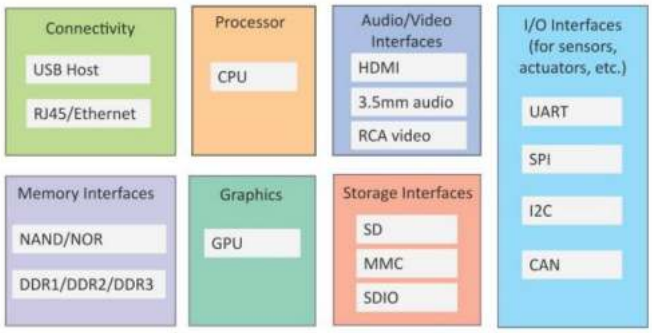
IOT Protocols



1. 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.
   a. HTTP
      Hypertext transfer protocol is a protocol that presents in an application layer for transmitting media documents. it is used to communicate between web browsers and servers. it makes a request to a server and then waits till it receives a response and in between the request server does not keep any data between the two requests.
   b. WebSocket
      This protocol enables two-way communication between a client and a host that can be run on an untrusted code in a controlled environment. This protocol is commonly used by web browsers.
2. 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.
   a. TCP
      The transmission control protocol is a protocol that defines how to establish and maintain a network that can exchange data in a proper manner using the internet protocol.
   b. UDP
      A user datagram protocol is a part of an internet protocol called the connectionless protocol. this protocol is not required to establish the connection to transfer data.
3. 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.
   a. IPv4
      This is a protocol address that is a unique and numerical label assigned to each device connected to the network. an IP address performs two main functions host and location addressing. IPv4 is an IP address that is 32-bit long.
   b. IPv6
      It is a successor of IPv4 that uses 128 bits for an IP address. it is developed by the IETF task force to deal with long-anticipated problems.
4. 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.
   a. Ethernet
      It is a set of technologies and protocols that are used primarily in LANs. it defines the physical layer and the medium access control for wired ethernet networks.
   b. WiFi
      It is a set of LAN protocols and specifies the set of media access control and physical layer protocols for implementing wireless local area networks.

Explain Physical Design of IOT?

1. The physical design of an IoT system includes the hardware components that make up the system, such as sensors, actuators, microcontrollers, communication modules, power supplies, and enclosures.
2. These components are designed to work together to gather data, process it, and communicate it to other devices and systems.
3. These devices generate data, and the data is used to perform analysis and do operations for improving the system.
4. For instance, a moisture sensor is used to obtain the moisture data from a location, and the system analyses it to give an output.
5. IOT devices are used to build a connection, process data, provide interfaces, provide storage, and provide graphics interfaces in an IoT system.



- Connectivity: Devices like USB hosts and ETHERNET are used for connectivity between the devices and the server.
- Processor: A processor like a CPU and other units are used to process the data. these data are further used to improve the decision quality of an IoT system.
- Audio/Video Interfaces: An interface like HDMI and RCA devices is used to record audio and videos in a system.
- Input/Output interface: To give input and output signals to sensors, and actuators we use things like UART, SPI, CAN, etc.
- Storage Interfaces: Things like SD, MMC, and SDIO are used to store the data generated from an IoT device.
- Other things like DDR and GPU are used to control the activity of an IoT system.

Define sensors in IOT? Give Classification of sensors and Explain types of sensors

In IoT, a sensor is a device that detects and measures physical phenomena such as temperature, humidity, light, motion, and sound, and converts them into digital data that can be processed and analyzed by an IoT system.

Sensors are a key component of IoT devices, as they enable devices to interact with the physical world and provide data that can be used to make informed decisions.

Classification of Sensors

Here are some common classifications of sensors:

1. Based on the physical phenomena they measure:
    a. Temperature sensors
    b. Pressure sensors
    c. Light sensors
    d. Sound sensors
2. Based on their applications:
    a. Environmental sensors (e.g., temperature, humidity, and air quality sensors)
    b. Industrial sensors (e.g., pressure, flow, and level sensors)
    c. Medical sensors (e.g., blood pressure sensors, ECG sensors)
    d. Automotive sensors (e.g., speed sensors, oxygen sensors)

There are many types of sensors that can be used in IoT devices, including:

1. Temperature sensors: Temperature sensors detect changes in temperature and are commonly used in heating and cooling systems, weather stations, and food storage systems.
2. Humidity sensors: Humidity sensors detect changes in humidity and are commonly used in **climate control systems**, **greenhouses**, and **food storage systems**.
3. Light sensors: Light sensors detect changes in light levels and are commonly used in lighting systems, security systems, and smart windows.
4. Motion sensors: Motion sensors detect movement and are commonly used in security systems, lighting systems, and energy management systems.
5. Sound sensors: Sound sensors detect changes in sound levels and are commonly used in acoustic monitoring systems, smart home assistants, and security systems.

What are Actuators

Actuators are devices used in IoT systems to control physical processes or systems by converting electrical or digital signals into mechanical or physical actions.

They are used to execute commands issued by IoT controllers or decision-making algorithms in response to sensor data or user inputs.

Actuators can be simple or complex devices that perform a wide range of actions, including:

1. Movement: Actuators can be used to move physical objects, such as opening or closing doors, windows, valves, or gates. They can also control motors to move wheels, drones, or robotic arms.
2. Heating and cooling: Actuators can control temperature by adjusting heating, ventilation, and air conditioning (HVAC) systems or turning on or off heaters or coolers.
3. Lighting: Actuators can control the brightness or color of lights or turn them on or off.
4. Sound: Actuators can produce sounds, such as alarms, beeps, or music.

Different types of Actuators

Servo motors: Servo motors are small motors that provide precise angular control and are often used in robotic applications.
They are controlled by sending a digital signal to the motor, which determines the position of the motor shaft.

Linear actuators: Linear actuators are devices that provide linear motion and are used to move objects in a straight line.
They are commonly used in home automation, robotics, and automotive applications.

Pumps: Pumps are used in many IoT applications, such as agriculture, manufacturing, and water treatment.
They are often controlled by digital signals and can be used to move liquids or gases.

Lights: Lights are one of the most common types of actuators in IoT applications.
They are often controlled by digital signals and can be used for various purposes, such as security lighting, mood lighting, and energy management.

Speakers: Speakers are used to convert digital signals into sound.
They are often used in IoT applications, such as home automation, voice assistants, and smart speakers.

Explain Logical Design of IOT?

In IoT, the logical design **refers to the high-level design of the system**, which includes the architecture, data flows, and interfaces between different components of the system.

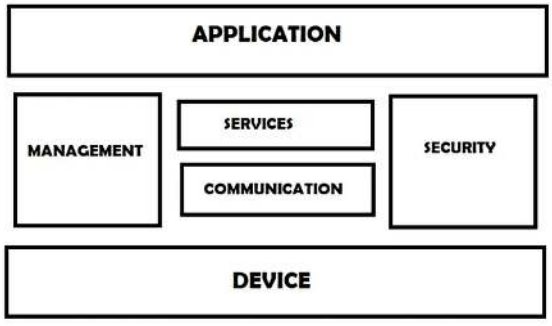
The logical design of an IoT system **focuses on defining the system's functionality** and how it will interact with other systems and users.

IoT logical design includes:

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

IOT Functional blocks

IoT systems include several functional blocks such as Devices, communication, security, services, and application.



This is an interface that the users can use to control and monitor various aspects of the IoT system.

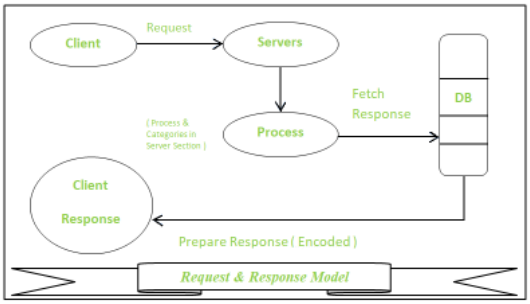
The application also allows users to view the **system status** and **view or analyze the processed data.**

Optional(Device: An IoT system comprises of devices that provide sensing, actuation, monitoring, and control functions.
Communication: Handles the communication for the IoT system.
Services: services for device monitoring, device control service, data publishing services, and services for device discovery.
Management: this block provides various functions to govern the IoT system.
Security: This block secures the IoT system and by providing functions such as authentication, authorization, message and content integrity, and data security.)

IOT communication Model

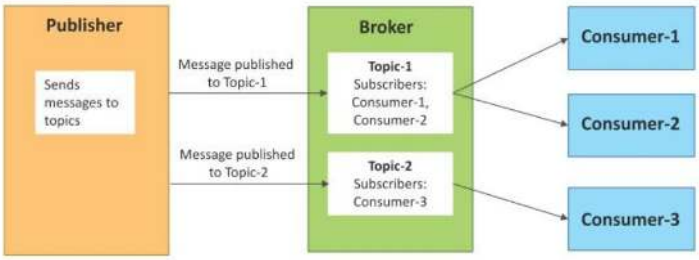
Request Response Communication Model

- In this model client sends a request message to the server, typically containing information about the operation to be performed.
- The server receives the request message fetches the data, retrieves resource representations, prepares the response, and then sends the response to the client.
- The client receives the response message and processes it, typically using the information provided in the response to update its state or display the result to the user.



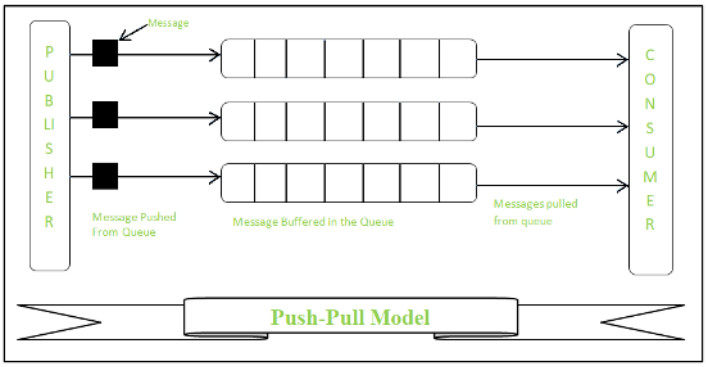
Publisher Subscribe Communication Model

- The Model comprises three entities publishers, Brokers and Consumers
- Publishers are the source of data. Publishers send the data to the topics which are managed by the broker.
- Publishers are not aware of the consumers.
- Consumers subscribe to the topic which are managed by the broker
- When the broker receives the data for a topic from the publisher, it sends the data to all subscribed consumers



Push Pull Communication Model

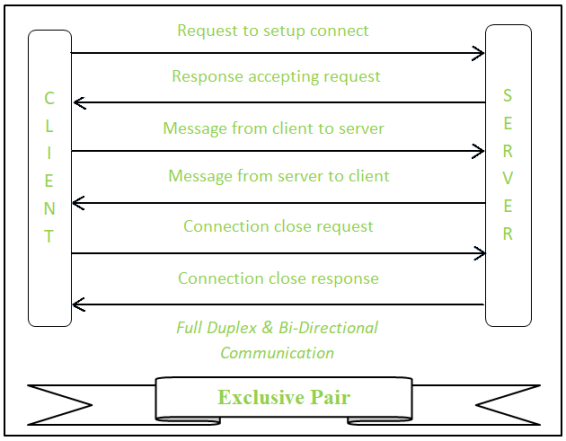
- The push-pull model constitutes data publishers, data consumers, and data queues.
- Publishers and Consumers are not aware of each other.
- Publishers publish the message/data and push it into the queue. The consumers, present on the other side, pull the data out of the queue. Thus, the queue acts as the buffer for the message when the difference occurs in the rate of push or pull of data on the side of a publisher and consumer.
- Queues help in decoupling the messaging between the producer and consumer.
- Queues also act as a buffer which helps in situations where there is a mismatch between the rate at which the producers push the data and consumers pull the data.



Exclusive Pair Communication Model

- Exclusive Pair is the bi-directional model, including full-duplex communication among client and server.

- The connection is constant and remains open till the client sends a request to close the connection.
- The Server has the record of all the connections which has been opened.
- This is a state-full connection model and the server is aware of all open connections.
- WebSocket based communication API is fully based on this model.



IOT communication API

In IoT, there are 2 communication APIs –

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

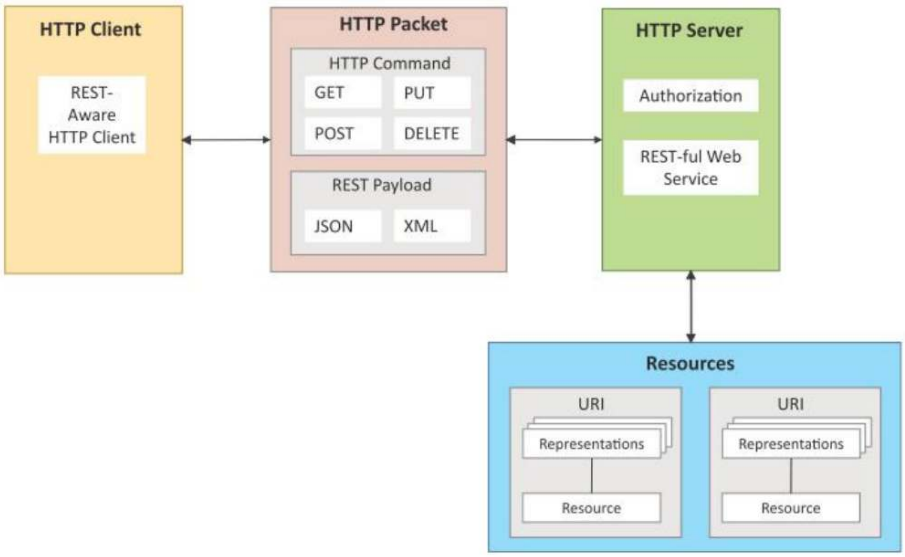
REST Based API

- REST (Representational State Transfer) is an architectural style that defines a set of constraints to be used when creating web services.
- REST-based APIs are a type of web service that follow these constraints and use HTTP requests to interact with data resources.
- REST-based APIs have a client-server architecture, where the client is responsible for sending HTTP requests to the server to retrieve or manipulate data resources.
- The server, in turn, responds to these requests with a representation of the requested resource, typically in JSON or XML format.

The key features of REST-based APIs include:

Resource identification through URIs: Each resource in a REST-based API is identified by a unique URI (Uniform Resource Identifier).

Use of standard HTTP methods: REST-based APIs use standard HTTP methods such as GET, POST, PUT, PATCH, and DELETE to interact with resources.



Web Socket based Communication APIs

Same as Exclusive pair Communication Model

IOT Levels and Deployment Template

An IOT system comprises of the following components:

Device:

- Allows identification, remote sensing, actuating and remote monitoring capabilities.
- IoT devices include wireless sensors, software, actuators, and computer devices.
- They are attached to a particular object that operates through the internet, enabling the transfer of data among objects or people automatically without human intervention.

Resource:

- These are Software components on the IoT device for accessing, processing, and storing sensor information, or controlling actuators connected to the device.
- Resources also include the software components that enable network access for the device.

Controller Service:

- A native service that runs on the device and work between node device and web services.
- Controller service sends data from the device to the web service and receives commands from the application (via web services) for controlling the device.

Database:

- A storage place for Collected or generated data.
- It can be local or cloud based.

Web Service:

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

Analysis Component:

- Responsible for analyzing the IoT data and generate results in a form which are easy for the user to understand.

Application:

- IoT applications provide an interface that the users can use to control and monitor various aspects of the IoT system.
- It allow users to view the system Monitor and processed data.

IOT levels

IOT Level 1

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

It is suitable for modeling low- cost and low-complexity solutions where the data involved is not big and the analysis requirements are not computationally intensive.

IOT Level 2

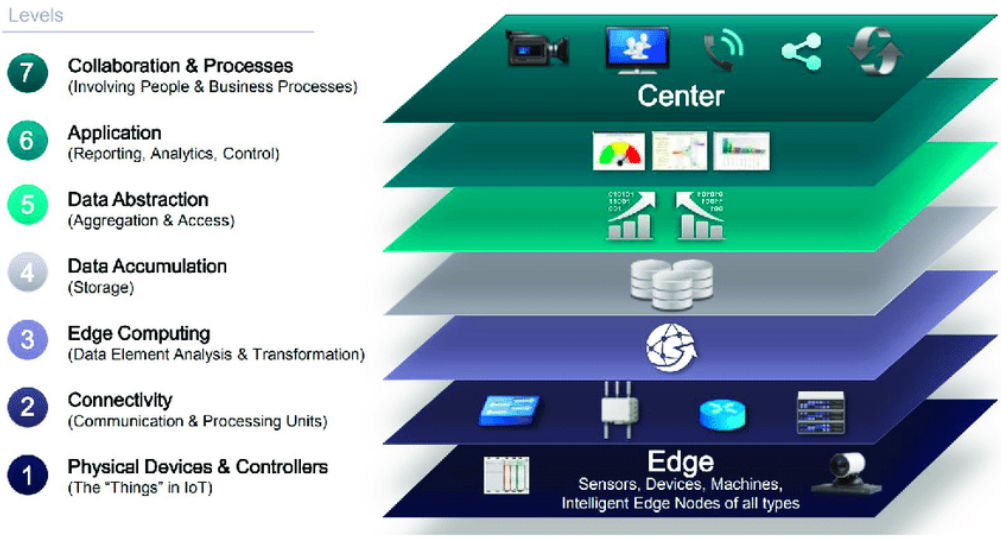
It has a single node that performs sensing and/or actuation and local analysis (IoT Device and collected data).

In this IoT Level Database and application establish in Cloud.

It is useful for solutions where the data involved is big, however, the primary analysis requirement is not computationally intensive and can be done locally itself.

Explain the IOT World Forum (IoTWF) Standardized Architecture.

Youtube link: **https://youtu.be/9a9VQnmdVzY**

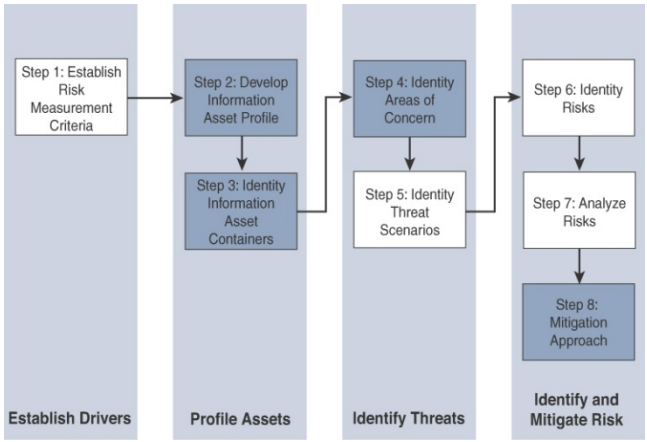


The Internet of Things World Forum (IoTWF) Standardized Architecture is a **reference model that outlines the key components** of an IoT system and how they interact with each other.

The architecture provides a common framework for designing and building IoT solutions, which can be applied across different industries and use cases.

1. Physical Devices & Controllers
   - **a.** This layer includes all the physical devices that are connected to the IoT system, such as sensors, actuators, and other smart devices.
   - **b.** These devices are responsible for collecting and transmitting data to the IoT platform.
2. Connectivity
   - **a.** This layer deals with connectivity that involves communication and processing.
   - **b.** It is responsible for reliable communication and transmission of data between devices and network.
3. Edge Computing
   - **a.** Edge computing is implemented in the connectivity layer, between the device layer and the data layer.
   - **b.** It **analyse the data**, **reduce network latency and bandwidth requirements, and improve overall system performance and reliability.**
   - **c.** Optional(By performing data processing and analysis locally, edge computing can reduce the amount of data that needs to be transmitted to the cloud for further analysis)
4. Data Accumulation
   - **a.** This layer is responsible for managing the data generated by the devices.
   - **b.** It includes **data storage, processing, and analysis tools** that enable real-time insights and decision-making.
5. Data Abstraction
   - **a.** A data abstraction layer abstracts the raw data generated by devices into a more meaningful and structured format.
   - **b.** Using this layer, **organizations can better interact with device data through data models, APIs, and other technologies.**
6. Application
   - **a.** This layer provides the user interface and applications that interact with the IoT system.
   - **b.** It includes dashboards, analytics tools, and other applications that enable users to visualize and interact with the data generated by the devices.
7. Collaboration & Processes
   - **a.** Collaboration and processes are part of the business layer, which defines the business processes and rules that govern the operation of the IoT system.
   - **b.** By leveraging data generated by IoT systems, organizations can optimize their operations and improve their bottom line.

Explain the OCTAVE risk assessment frameworks with neat diagram.



- OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) is a risk assessment framework.
- It is designed to help organizations identify and manage risks associated with their critical assets and information.

Step1:
The first step of the OCTAVE Allegro methodology is to establish a risk measurement criterion.
Establishing risk measurement criteria refers to the process of defining the metrics and parameters that will be used to measure and assess the level of risk associated with a particular asset or threat.

Step 2:
The second step is to develop an information asset profile.
Developing an information asset profile involves creating a detailed description of an organization's information assets, such as data, systems, and applications.

Step 3:
The third step is to identify information asset containers.
Identifying information asset containers involves grouping an organization's information assets into categories or containers based on their similarities in terms of purpose, functionality, and value.

Step 4:
The fourth step is to identify areas of concern.
Identifying areas of concern refers to the process of identifying potential risks or vulnerabilities within an organization's assets that may pose a threat to the confidentiality, integrity, or availability of its information.

Step 5:
The fifth step is to identify threat scenarios.
Identifying threat scenarios involves a structured process of analyzing the organization's information assets, identifying potential threats, and mapping out how those threats could exploit vulnerabilities to cause harm.

Step 6:
At the sixth step risks are identified.
This process typically involves gathering and analyzing data about an organization's assets to identify areas where security controls may be weak or non-existent.

Step 7:
The seventh step is risk analysis
Risk analysis is used to determine the likelihood and impact of each identified risk, and to prioritize risks according to their potential impact on the organization.

Step 8:
Finally, mitigation is applied at the eighth step.
Mitigation approach refers to the process of developing and implementing measures to reduce the identified risks to an acceptable level.

Compare CoAP and MQTT protocols.

Basis of	COAP	MQTT
Abbreviation	Constrained Application Protocol	Message Queuing Telemetry Transport
Communication Type	It uses Request-Response model.	It uses Publish-Subscribe model
Messaging Mode	This uses both Asynchronous and Synchronous.	This uses only Asynchronous
Transport layer protocol	This mainly uses User Datagram protocol(UDP)	This mainly uses Transmission Control protocol(TCP)
Header size	It has 4 bytes sized header	It has 2 bytes sized header
RESTful based	Yes it uses REST principles	No it does not uses REST principles
Persistence support	It does not has such support	It supports and best used for live data communication
Message Labelling	It provides by adding labels to the messages.	It has no such feature.
Usability/Security	It is used in Utility area networks and has secured mechanism.	It is used in IoT applications and is secure
Effectiveness	Effectiveness in LNN is excellent.	Effectiveness in LNN is low.
Communication Model	Communication model is one-one.	Communication model is many-many.

What are TCP and UDP Ports? Explain with Example.

- TCP is a connection-oriented protocol, which means it establishes a connection between two devices before transmitting data.
- It guarantees the delivery of data by using a three-way handshake process to establish and terminate connections.
- TCP is slower than UDP because of this connection overhead.
- TCP ports are used for services that require reliable transmission of data, such as web browsing, email, and file transfer.

For example, when you connect to a web server, your computer sends a request to the server's port 80 using TCP. The server then acknowledges the request and sends the requested web page back to your computer on a different port.

- UDP, on the other hand, is a connectionless protocol that does not establish a connection before transmitting data.
- It is faster than TCP because it does not have the connection overhead, but it does not guarantee the delivery of data.
- UDP ports are used for services that require fast transmission of data but can tolerate some loss of data, such as online gaming, video streaming, and DNS.

For example, when you play an online game, your computer sends small packets of data to the game server's port using UDP. The game server sends back data to your computer's UDP port, and if any packets are lost during transmission, they are not retransmitted. This results in a faster gameplay experience at the cost of some data loss.

What is DNS how does it works?

I. DNS (Domain Name System) is a protocol used on the internet to translate human-readable domain names, such as www.example.com, into IP (Internet Protocol) addresses, such as 192.0.2.1, which are used by machines to identify and communicate with each other.
II. DNS works by using a hierarchical system of servers and databases to map domain names to IP addresses.
III. When you enter a domain name into your web browser, your computer sends a request to a DNS resolver, which is a server that is configured to query other DNS servers to find the IP address associated with the domain name.
IV. The resolver first checks its local cache to see if it has previously resolved the domain name.
V. If it has, it returns the IP address to your computer.
VI. If it has not, the resolver queries one or more DNS servers higher up in the hierarchy until it finds a server that can provide the IP address for the domain name.
VII. The DNS servers in the hierarchy can be thought of as a distributed database that contains mappings between domain names and IP addresses.
VIII. Once the resolver obtains the IP address, it returns it to your computer, which can then use that IP address to establish a connection to the web server associated with the domain name.
IX. This process happens quickly and automatically behind the scenes every time you visit a website or use an internet-connected application that relies on DNS.
X. DNS operates in a hierarchical structure, with top-level domains (TLDs) such as .com, .org, and .edu at the highest level, and subdomains below them.

Compare IEEE 802.15.4 ,IEEE 1901.2a, LoRaWAN, and NB-IoT with respect to different characteristics as data rate, frequency band, topology,range, wired or wireless etc.

(Help)

Explain Physical layer and MAC layer of IEEE 802.15.4

Explain working of NFC standard. Is NFC communication this secure?

Explain message format of CoAP and MQTT Protocols.

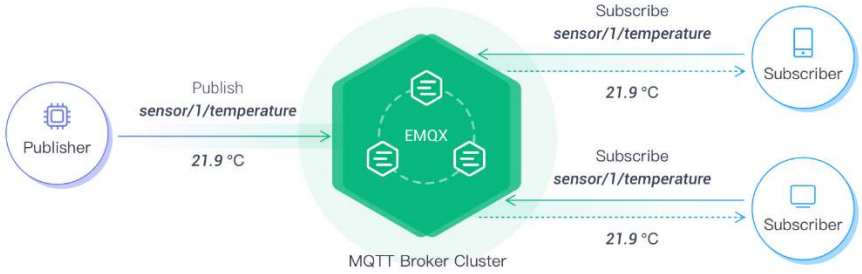
Write a short note on 6TiSCH add-on

What are the common challenges faced by OT Security?

Explain four schedule management mechanisms in 6TiSCH architecture.

(Paper)

With neat diagram explain MQTT publish subscribe framework, w.r.t. one example



The MQTT Publish-Subscribe pattern has four main components: Publisher, Subscriber, Broker, and Topic.
Publisher

The publisher is responsible for publishing messages to a topic.
It can only send data to one topic at a time and does not need to be concerned about whether the subscribers are online when publishing a message.

Subscriber

The subscriber receives messages by subscribing to a topic and can subscribe to multiple topics at once.
MQTT also supports load-balancing subscriptions among multiple subscribers through shared subscriptions.

Broker

The broker is responsible for receiving messages from publishers and forwarding them to the appropriate subscribers.
In addition, the broker also handles requests from clients for connecting, disconnecting, subscribing, and unsubscribing.

Topic

MQTT routes messages based on topics. A topic is typically leveled and separated with a slash / between the levels, this is similar to URL paths. For example, a topic could be sensor/1/temperature.
For example, a smart home system might use MQTT publish/subscribe to allow a motion sensor to publish a message when motion is detected.