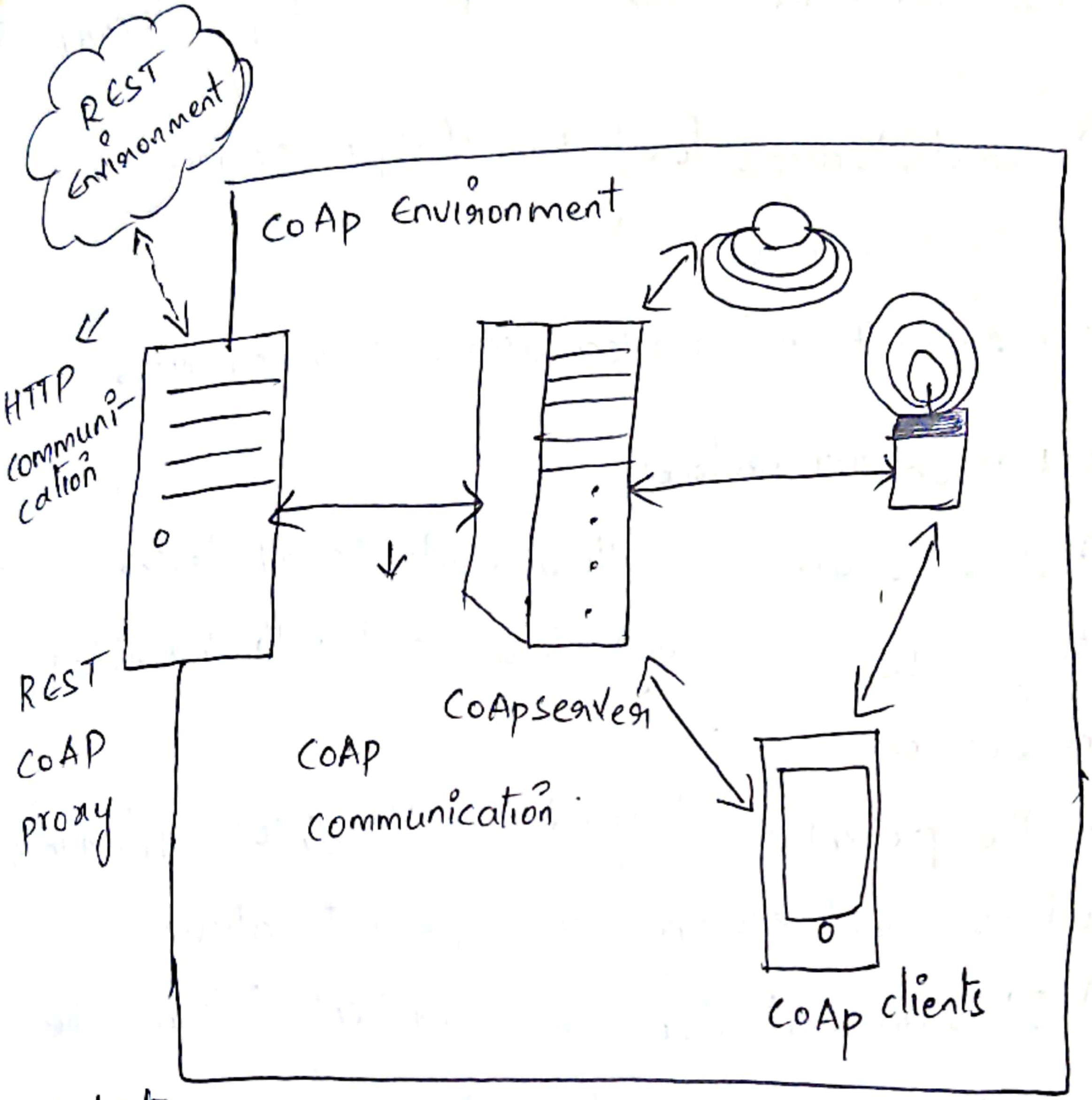
UNIT-III:- Syllabus

Design principles for the web Connectivity for connected Devices- web communication protocols for connected devices, Message communication protocols for connected devices, web connectivity for connected Devices.

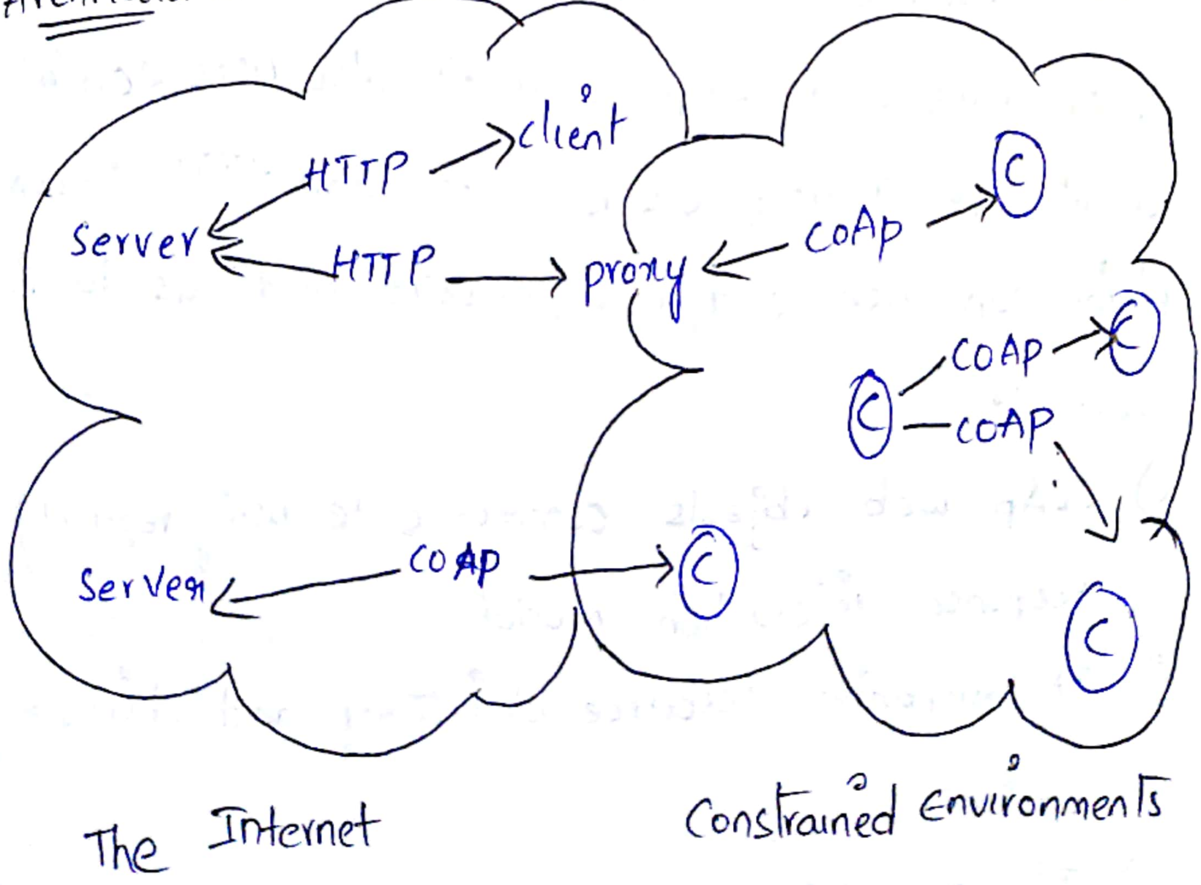
1) web communication protocols for connected devices:-

The protocols are used for Communication between machines or between a machine and server. Due to constraints in processing capabilities and the low power requirements of IoT devices with limited bandwidth capabilities, a need was felt for dedicated standards and protocols especially designed for IoT. The protocols are:

i) Constrained Application protocol (CoAp):-



Architecture:-



a) Direct and indirect access of CoAp client objects to CoAp Server.

b) CoAp clients access for lookup of object or resource using a resource directory.

c) CoAp client and Server access using proxies.

1) It is an IOT protocol

2) It is designed to allow simple & small device to join the IoT through low bandwidth restricted networks.

3) The protocol is designed for M2M & IoT applications such as smart energy & buildings automation.

4) It is an application layer protocol follows the request - response pattern/model.

5) CoAp runs over ODP portal. It also uses Restful architecture. It uses less resources than HTTP. In CoAp client can use GET, PUT, DELETE methods during request.

6) CoAp web objects communicate using request/ response interaction model.

7) It supports resource directory and discovery functions.

8) It provides asynchronous communication

9) It is developed by ICTF(Internet Engineering Task Force) ·Developed to enable smart device to connect to the internet.

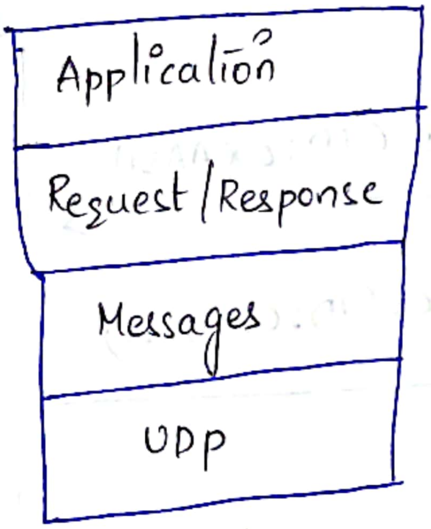
10) COAP is based on "client Server model!"

i) Asusual, clients shall send the request to the server.

ii) servers shall respond

iii) clients are free to do " Get, put --!"

*COAP layers:-



It is divided into 2 layers:

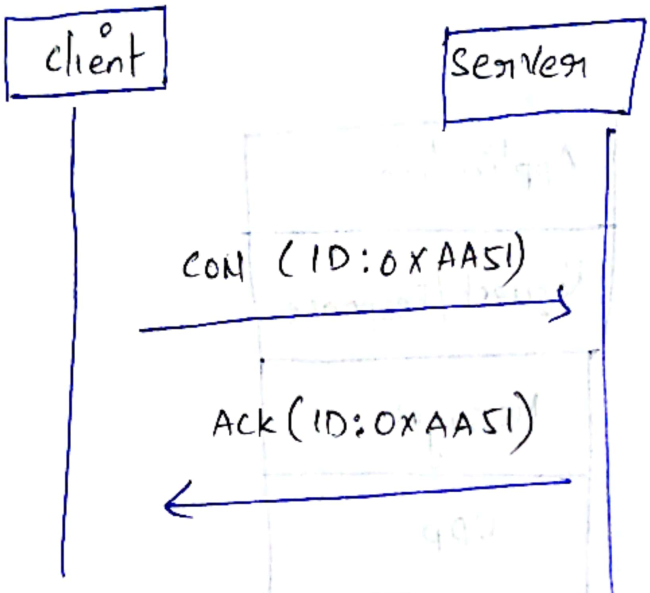
1) upper layer:- It Concerns Communication method & deal with request/response method.

2) lower layer:- It has been designed to deal with UDP & asychronous messages.

CoAp - Message layer - Message Types:-

It supports 4 types of messages.

i, Confirmable (CON):- Reliable Messaging is obtained using a confirmable message (CON). A CON message is sent again and again until the other party sends an acknowledge message (Ack). The Ack message contains the same ID of the confirmable message (CON).

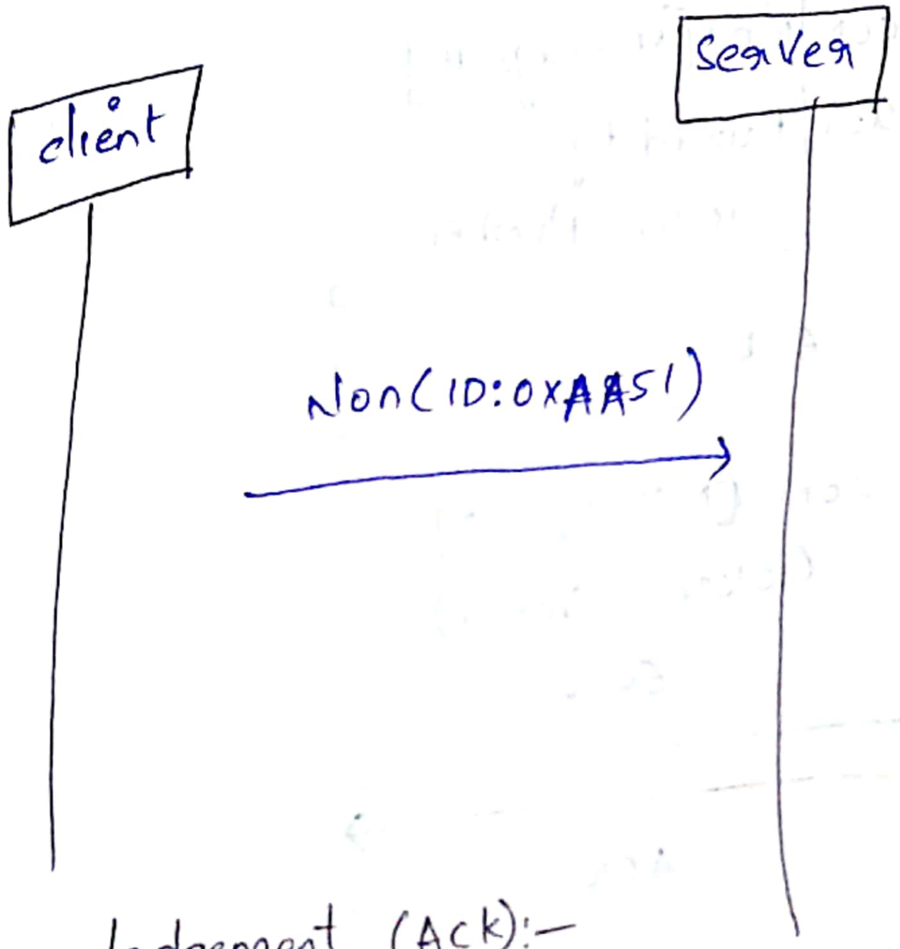


(ii) Non-confirmable (Non):-

These are messages don't require an acknowledge by the server.

They are unreliable messages or in other words messages that do not ~~occur~~ contain critical information that must be delivered t the server.

Even if these messages are unreliable, they have a unique ID.

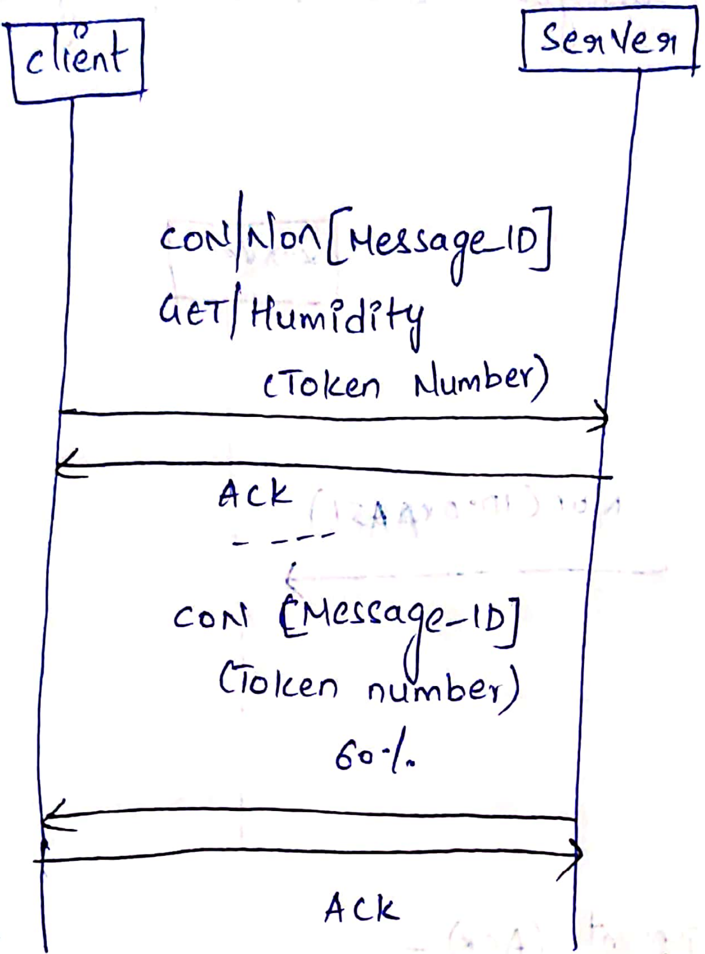


iii) Acknowledgement (Ack):-

COAP - Request / Response layer - Messages.

piggy backed: client shall send request with a CON/Non.

Ack shall be received immediately with corresponding token number and message. if not availble, the failure code shall be embedded as a part of the Ack.



iv) Rest

*COAP Message Format:-

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5

V	T	TKL	code	Message ID
Token (if any, TKl bytes)...				
options (if any)...				
1 1 1 1 1 1 1 1	payload (if any)...			

V:- It is 2 bit unsigned integer. It mentions CoAp version number. set to one.

T:- It is 2 bit unsigned integer. Indicates message type viz. confirmable (0), non confirmable (1), Ack (2), ResT (3).

TKL:- It is 4 bit unsigned integer, indicates length of token (0 to 8 bytes)

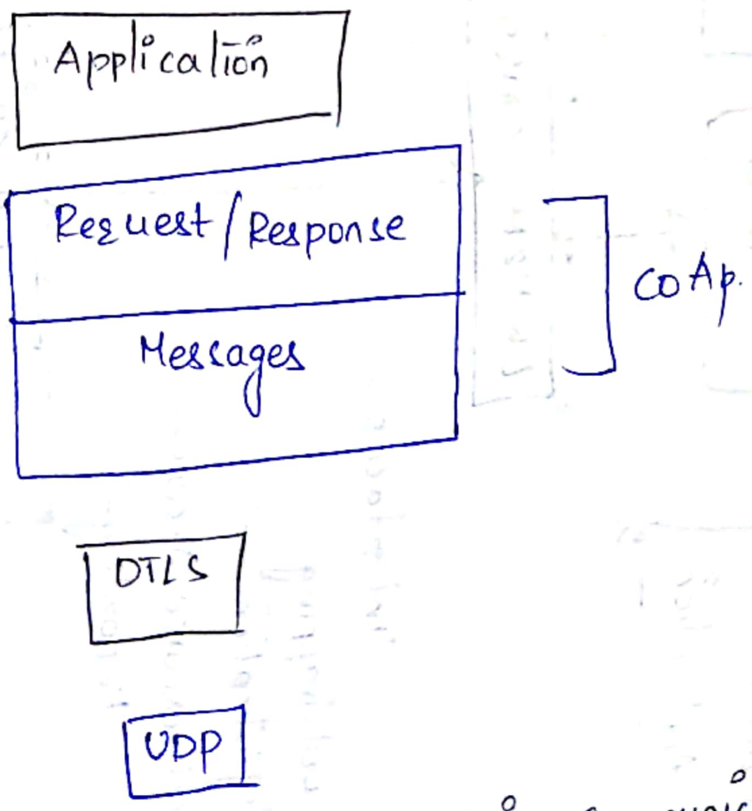
code:- Response code. (8 bit length)

Message ID:- identifier for each message sent. (16 bit).

Token:- optional response matching token.

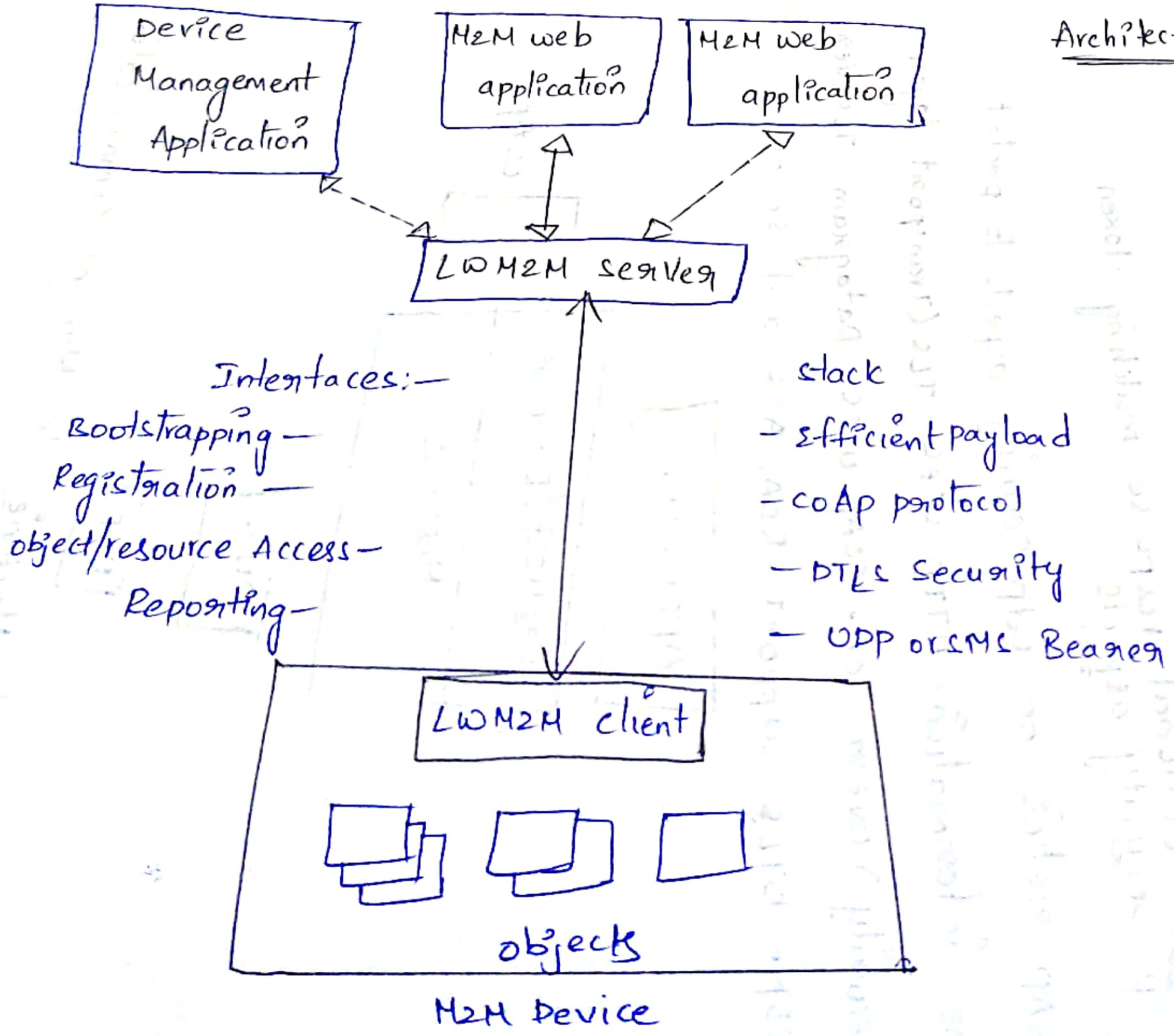
#CoAP security aspects:-

CoAP relies on udp security aspects to protect the information. HTTP uses TLS (Transport layer security) over Tcp. CoAp uses Datagram TLS over UDP. DTLS support RSA, AES, and so on.



8.(ii) Light Weight Machine-to-Machine Communication protocol (LWM2M):-

Architecture:—

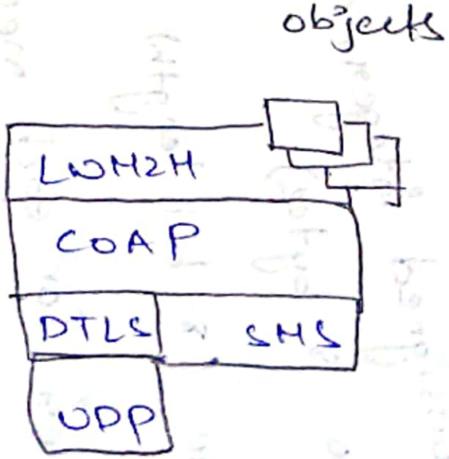


Interfaces:—

- Bootstrapping —
- Registration —
- object/resource Access —
- Reporting —

stack

- Efficient payload
- coAp protocol
- DTLS Security
- UDP or SMS Bearer



→An object or resource use CoAP, DTLS and UDP or SMS protocols for sending a request or response.

→use of plain text for a resource or use of JSON during a single data transfer.

→An object or its resource access using an URI

→Interface functions are for bootstrapping, registration, deregister (or) updating a client and its object.

→use of object model for resources and object can have single or multiple instances.

→OMA or other standard specifying organization defines the LWM2M objects for usages in M2M communication.

→organizations can register the other LWM2M objects & resources.

→M2M Management functions can be M2M service bootstrap function (MCBF) for credentials of devices and gateway.

2) Message Communication protocols for connected devices:—

① MQTT (Message Querying Telemeatry Transport)

② XMPP (Extensible Messaging and presence protocol)

③ Advanced Message Queuing protocol (AMQP)

① MQTT protocol:-

MQTT stands for message Queuying Telementary Transport.

→ It is a Machine to Machine (M2M) (or) IOT connectivity protocol.

→ It is a publish-Subscribe-based messaging protocol that transport messages between devices.

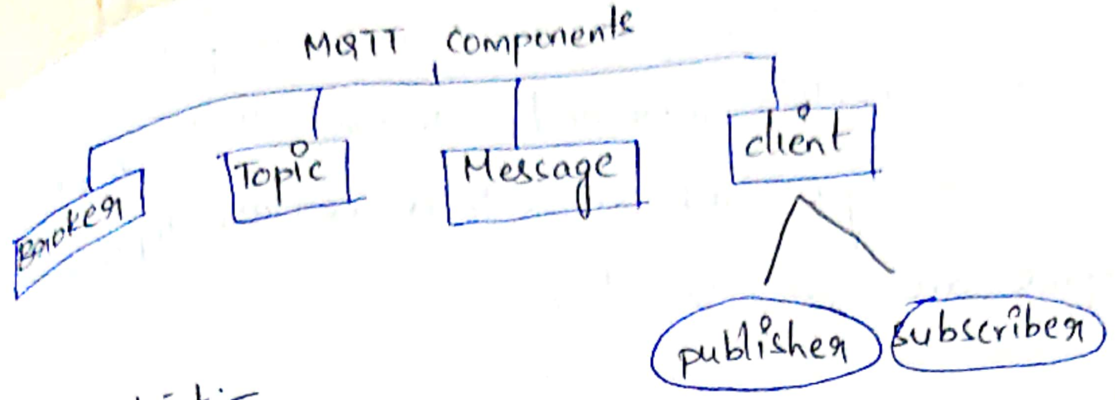
→ It usually runs over TC p/IP protocol.

→ It is very light weight and then suited for M2M WSN (Wireless Sensor Network) and IOT scenarios where sensor nodes Communicate with application through the MQTT message broker.

→ MQTT was intially developed by IBM & Eurotech.

→ It is designed for limited devices & networks with high intency, low bandwidth.

* MQTT components/~~Architecture~~:-



MQTT client:- A client can be either publisher or subscriber or both. That is, a client publish a message & receive another message at the same time. client subscribe to topics to publish & receive messages.

① publisher – It is a process, a device does to send its message to the broker.

② subscriber – where a device does to retrieve a message from the broker.

MQTT Broker:- It receives subscription from client on topics, receive messages from clients and forward these messages based on client subscriptions to interested clients. It is responsible for dispatching all messages between the clients.

Topic:- A topic is an identifier used by MQTT broker to identify rightful clients for delivering messages.

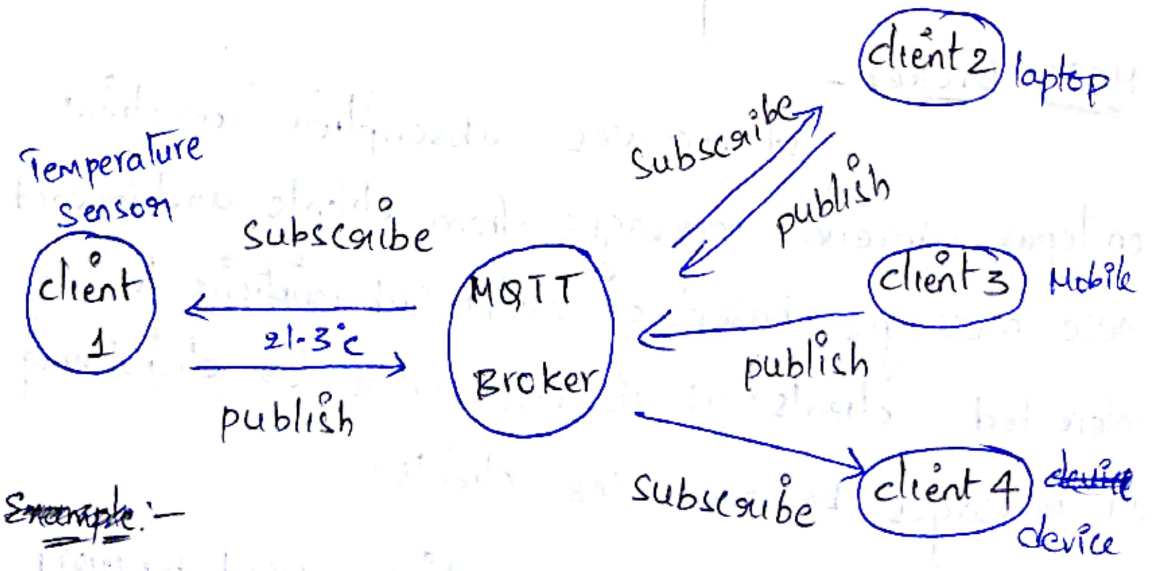
Each client that wants to send messages publishes them on a certain topic and each client that wants to receive messages subscribes to a certain topic.

Topics are case-sensitive.

Message:-

The message is the data that is carried out by the protocol across the network for the application. when the message is transmitted over the network, then the message contains the following parameters: payload data, Topic Name, collection of properties, Quality of Service (QoS).

* Architecture of MQTT and its working:-



~~Example~~:-

First of all, a subscriber subscribes to one or more topics.

In above diagram, the, laptop, mobile, device are subscribers. The temperature sensor is the publisher. Then one or more publishers publish messages to a server (MQTT Broker), (local or remote). The temperature sensor publishes its temperature value to the broker. Then the server publishes the message to the subscriber which have subscribed to the topic specified by the publishers

*Advantages of MQTT:-

Simplified communication:-

communication is a complex problem. MQTT reduces complexity, allowing a single connection to a message topic.

eliminate polling:-

MQTT allows instantaneous, push-based delivery, eliminating the need for message consumers to periodically check or "poll" for new information.

Dynamic targetting:-

MQTT makes discovery of services easier and less error prone.

Decouple and scale:-

MQTT also makes solutions more flexible and enables scale. It allows changes in communication patterns, changing or adding functionality without sending ripple effects across the system.

*characterstics:-

1) It is a M2M protocol, which means it allows device to communicate with one another.

2) It's a simple, lightweight messaging protocol that uses a publish/subscribe system to send and receive data between the client and server.

3) It is not necessary to establish a connection between the client and the server at the same time.

4) It allows for faster data transmission, allows for Quicker delivery. It's a protocol for sending message in real time.

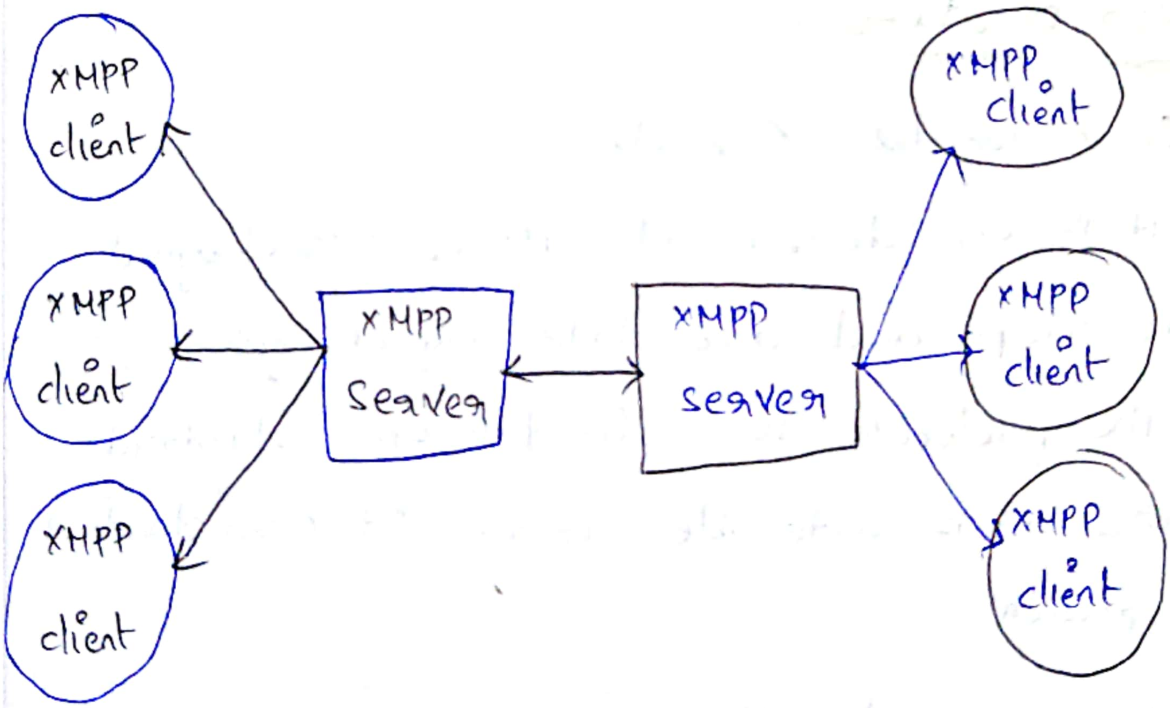
5) It allows customers to subscribe to a limited no. of topics in order to receive the information they require.

② XMPP:-

XMPP stands for extensible messaging and presence protocol.

It uses a client-server architecture. As the model is decentralized, no central server is required. open means to support M2M or P2P communications across a diverse set of networks. It is designed for messaging, chat, video, collaboration. The fundamental base of this protocol is XML and is used for messaging services such as whats App. The XMPP was originally named as Jabber and later known as XMPP.

XMPP architecture:-



The intial version of the XMPP was ~~TCP~~ with TCP using open ended XML.

After certain period of time, the XMPP was developed based on HTTP.

The XMPP can work with HTTP is through two different methods polling and Binding.

In polling method, the messages stored in the server are pulled or fetched. The fetching is done by XMPP client through HTTP GET and post vesuests. In Binding methods, Bidirectional streams over synchoronous HTTP enables the seaver to push the messages to the clients when they are sent. The binding approach is more effective than polling.

XMPP Denotation:-

X - X denotes extensible

- It is considered extensible as it is designed to accept and accomdate any changes.
- The protocols is defined in open standard and it is extencible because of open standard approach.

M - M denotes Messaging

- XMPP supports sending messages to the receiptents in real-time.

P - P denotes presence.

- It is helpful to identify the status

such as "offline" or "online" or "Busy".

- It also helps in understanding the receiplent is ready to receive the message or not.

P- p denotes protocol.

- The set of standards together acting as a protocol.

*XMPP characterstics:-

Security:- It has built in channel encryption feature, authenication and also resistant malware.

Decentralized:- It has decentralized client server architecture with an unlimited servers.

Extensible:- It can be used for variety of applications as it has which uses XML for delivering the services.

Community:- It has a good community of technologies, end users and also service providers.

*XMPP Advantages:-

open:- It is simple and free source

proven:- These are countless XMPP servers running in the internet today.

Flexible:- XMPP applications pastIM incorporate system administrations, content syndication, co-ordinated effort devices.

Secure:- provide end to end encryption by utilizing SAAL and TLS.

Extensible:- using the force of XML, anybody can construct custom usefulness on the top of the center conventions,

③ AMQP:-

→ The advanced message Querying protocol is an open standard for passing business messages between applications or organizations.

→ It connects system, feeds business processes with the information they need and reliably transmits ~~tt~~ onward the instructions that achieve their goals.

Amqp connects across:

1) organizations

- applications in different organizations

2) Technologies

- applications on different platform

3) Time.

- systems don't need to be available simultaneously.

4) space _ reliably operate at a distance, or over poor networks.

*AMQP Model:-

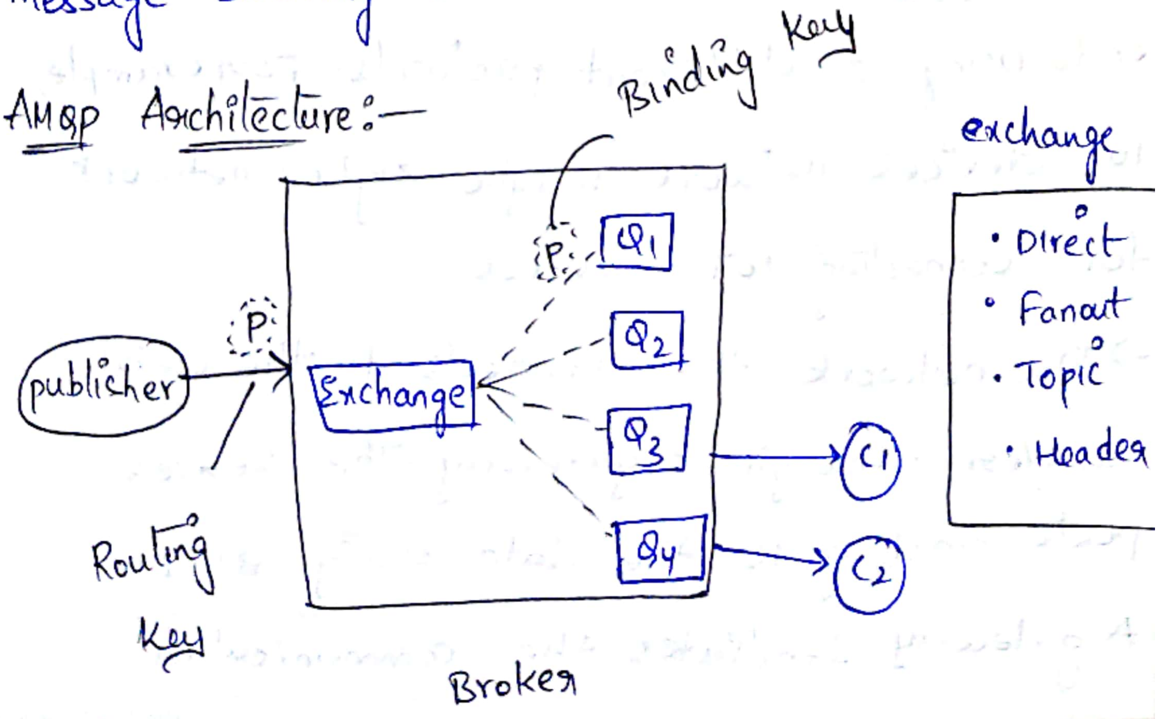
There are 3 principles sorts of parts:

① **Exchange:-** gets message from publisher applications and course these to messages queues depending the ability.

② **Message Queue:-** stores message until they can be securely prepared by an customer application.

③ **Binding:-** characterize the relationship between the a message Queue and exchange gives the message steering criteria.

AMQP Architecture:-



3) web connectivity for connected devices:-

① communication gateway:-

→ communication gateway connects 2 application layers, one at sender and other at receiver.

→ The gateway also enables use of 2 different protocols, one at sender and other at receiver ends.

→ The gateway faciliates the communication between web server using the tcp/ip protocol conversion gateway and IoT devices.

→ It also facilitates communication between the devices using ~~HTTP,~~ CoAp. client and server using HTTP

→ connects the sender and receiver ~~tools~~ ends using 2 different protocols. For example, IoT devices network maybe zigBee network. for connecting IoT devices.

→ The network then connects to the web server through a gateway. The server posts and gets the data using HTTP.

A gateway faciliates the communication between IoT devices and web server.

② SOAP and REST:—

SOAP is acronym for simple object Access protocol. It is an XML based messaging protocol for exchanging information among computers. It is an application of the XML specification. It is platform independent. The best way to communicate between application is over HTTP. It has 3 major characterstics

① extensibility (Security and web services addressing are among the extensions under development)

② neutrality (SOAP can operate over any protocol such as HTTP, SMTP, TCP, UDP)

③ independence (SOAP allows for any programming model)//

Rest is an architectural style for developing web services. REST stands for Representational state transfer It works by pulting in place very strict constraints for the development of web services. Rest architectural styles have 5 principles:

P1: Everything is resource

P2: Every resource is identified by URI

P3: Use simple and uniform interfaces

P4: communication is done by representation

P5: Be stateless.

③ HTTP Restful:—

REST uses various representations to represent a resource like Text, JSON, XML. JSON is Now the most popular format being used in web services.

HTTP methods:—

1) GET – provides a read only access to a resource

2) PUT – used to create new resource

3) DELETE – used to remove new resource

4) POST – used to update an existing resource or create new resource.

5) OPTIONS – used to get the supported operations on a resource.

4) Restful web services:— It is a collection of open protocols and standards used for exchanging data b/w applications. web services based on REST architecture are known as Restful web services. s/w applications written in various programming languages and running on various platforms can use web services to exchange data over Computer networks like the internet in a manner similar to inter-process communication on a single computer//

Other:—

Difference between CoAp and MQTT protocols:—

CoAp	MQTT
CoAp stands for Constrained Application protocol	MQTT stands for Message Query telemetry transport.

2) It uses a request-response model.	2) It uses publish-subscriber prototype
3) It uses asynchronous and synchronous messaging	3) It uses only asynchronous mode for messaging.
4) It uses UDP	4) It uses TCP
5) The size of CoAP is 4 bytes	5) The size of MQTT is 2 bytes.
6) It is RESTful based	6) It is not RESTful based.
7) It doesnot have persistance support	7) It is mainly used for live communication and has persistance support.
8) It will give labels to the messages.	8) It doesnot have any such function.
9) It has low latency and NAT issues	9) It has ~~low~~ high latency and NAT issues.
10) It has a secured system, and its usability is in utility area networks.	10) It is very secure and its usability is in IOT applications.

Difference between MQTT and HTTP:-

MQTT	HTTP
1) Message Queuing Telementary Transport	1) Hyper text transfer protocol.

2) It works on publich/ Subscribe Model	2) It works on request/ response Model.
3) It has less complexity	2) It has more complex.
4) It runs over transmissions Control protocol (MQT-SN can use UDP)	4) It runs over TCP
5) This protocols design is Data centric	5) This protocol design is Document centric
6) It is of 2 bytes	6) It is of 8 bytes
7) It works on 1883 portal	7) It works on 80/8080 port
8) It provides data security with SSL/TLS	8) It doesn't provide security but https is built for that
9) The message size generated is less as it as uses binary format.	9) The message size generated is more as it uses ASCII format
10) 3 quality of service settings	10) All messages get the same level of service.

Difference between AMQP, ~~CoAP~~ and ~~XH~~ HTTP

AMQP	HTTP
1) Advanced Mesage Queuing protocol	1) Hypertext transfer protocol.
2) It was developed by JP Morgan	2) It was developed by TIM Berners lee.
3) It is an asynchronous data Communication.	3) It is a synchronous communication.
4) we can easily setup and manage the AMQP protocol	4) HTTP can be used in every aspect.
5) The message delivery is guarantee in AMQP	5) There is no guarantee for message delivery.
6) AMQP provides a subscribe interface	6) It provides a point-point interface.
7) It can manage server issues	7) It cant react to server issues
8) It is a cost effective protocol	8) It is a multipurpose protocol.
9) It has processed the message into slots	9) It can process the message as segments.

Difference between CoAP & HTTP.

COAP	HTTP
1) It uses UDP	1) It uses TCP
2) It uses IpV6 along with 6LOWPAN	2) It uses Ip layer
3) ~~It is~~ CoAp uses both client-server & publish subscribe models	3) It uses client and server architecture.
4) less overhead and it is simple	4) More overhead and it is complex
5) It doesn't need asynchronous Communication	5) It needs asynchronous communication
6) ~~Thee~~ The header size of CoAp is 4 byte.	6) It is undefined
7) Message size is small and undefined	7) Message size is ~~Small~~ large and undefined.
8) port number is 5683 (UDP port)	8) 80/443 (TLS/SSL) port number
9) Design for resource Constrained networking devices such as WSN/IOT/M2M	9) Design for internet devices where there is no issue of any resources.