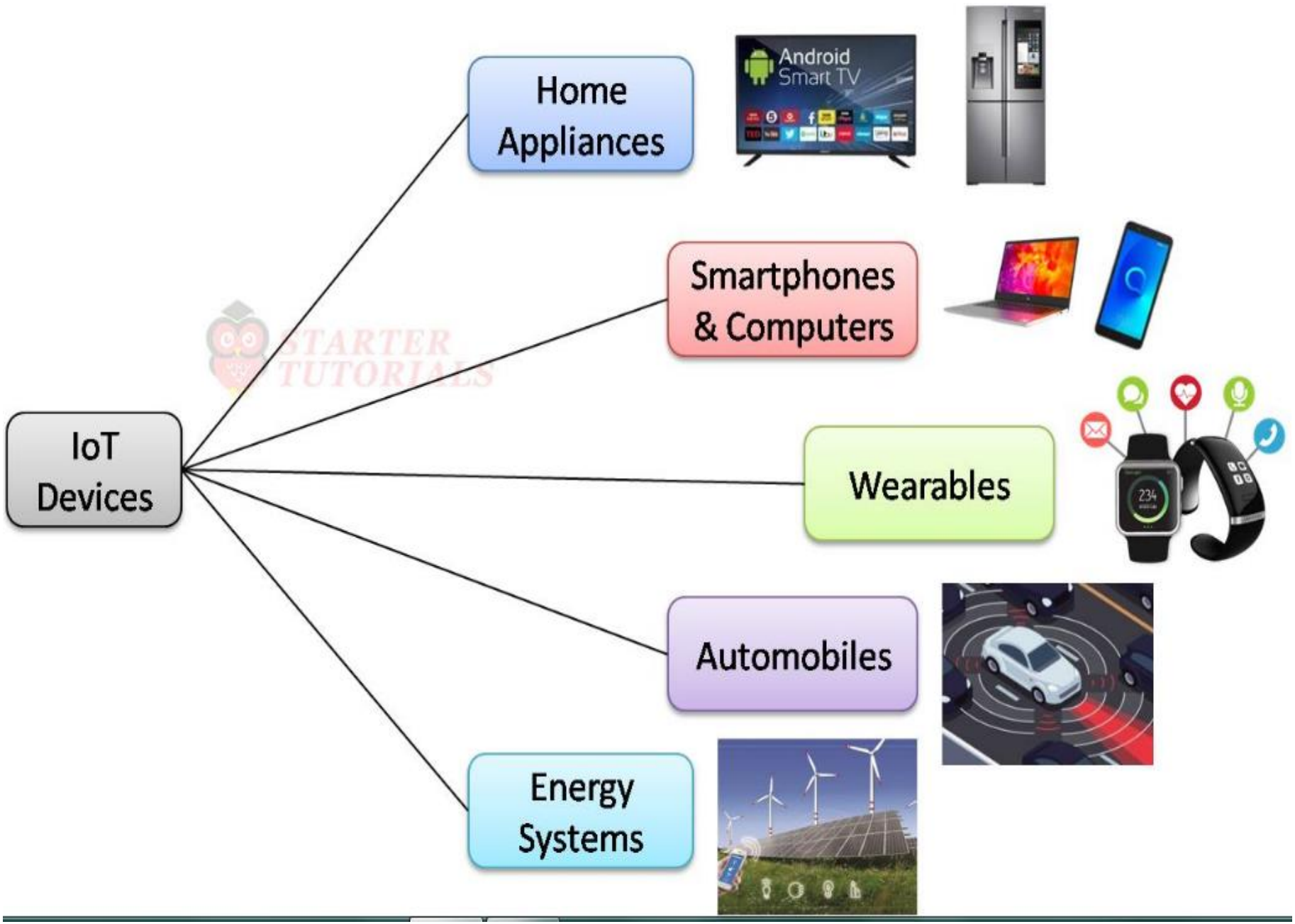
Physical Structure of IOT

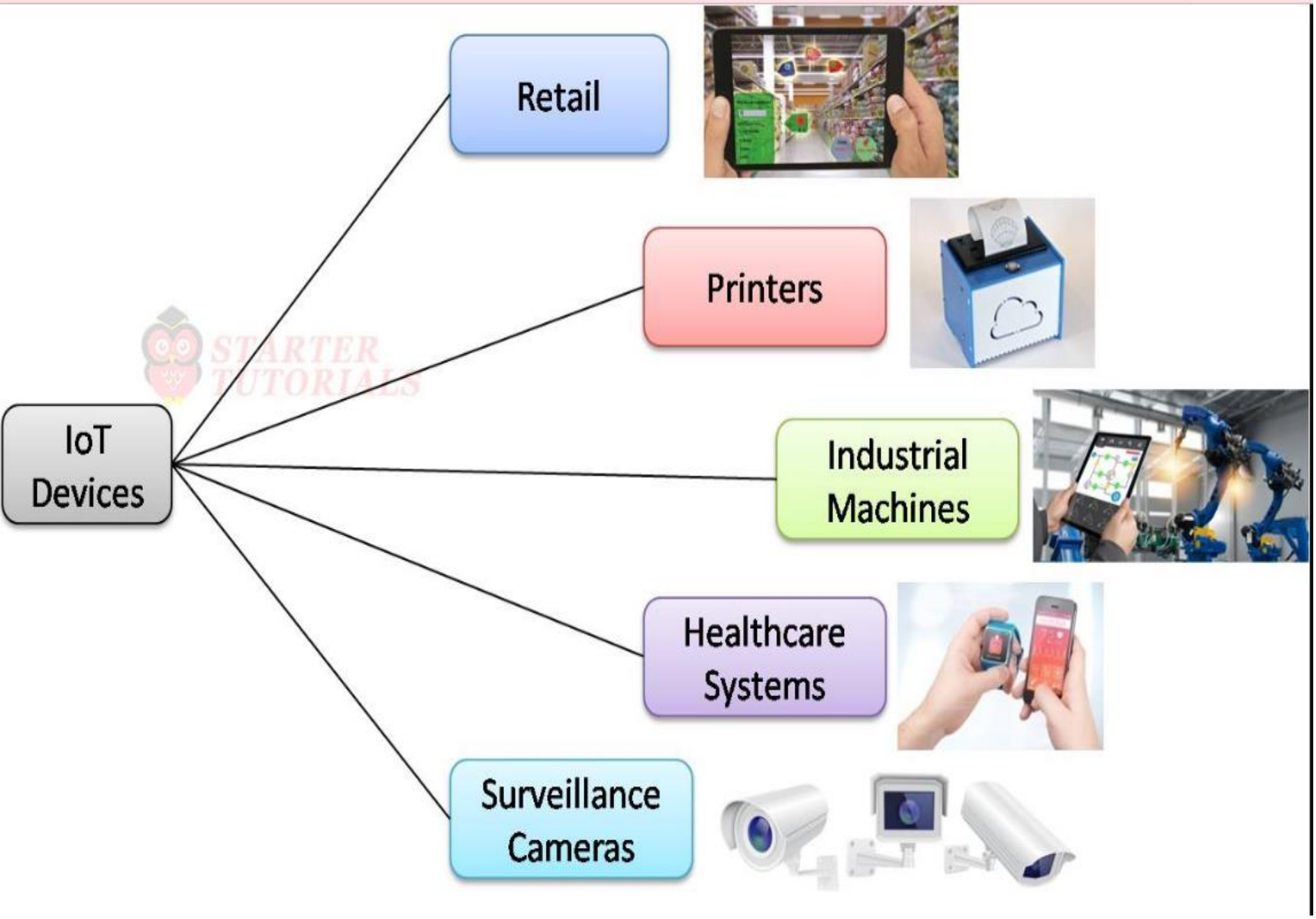
What are the structure of IoT?

- A basic IoT architecture consists of three layers: **Perception (the sensors, gadgets, and other devices) Network (the connectivity between devices) Application (the layer the user interacts with)**

Physical Design of IoT: Things

Things in IoT refers to IoT devices
Things have unique identities.
Things can perform sensing, actuation, and monitoring
Some of the examples of IoT devices are:
Home appliances: smart TV, smart refrigerator, smart AC, etc.
Smart phones and computers
Wearables: smart watch, smart sensors, etc.
Automobiles like self-driving cars
Energy systems
Retail : smart payment
Printers
Industrial machines
Healthcare: smart watch, smart healthcare, etc.
Surveillance: smart cameras, smart trackers, etc.


IoT Devices
Home Appliances
Android Smart TV
Smartphones & Computers
Wearables
Automobiles
Energy Systems
STARTER TUTORIALS


IoT Devices
Retail
Printers
Industrial Machines
Healthcare Systems
Surveillance Cameras

Things actions in IOT

Things can exchange data with other connected devices and applications, collect data from other devices, and process the data either locally or send it to centralized servers or cloud. IoT devices can have several interfaces like:

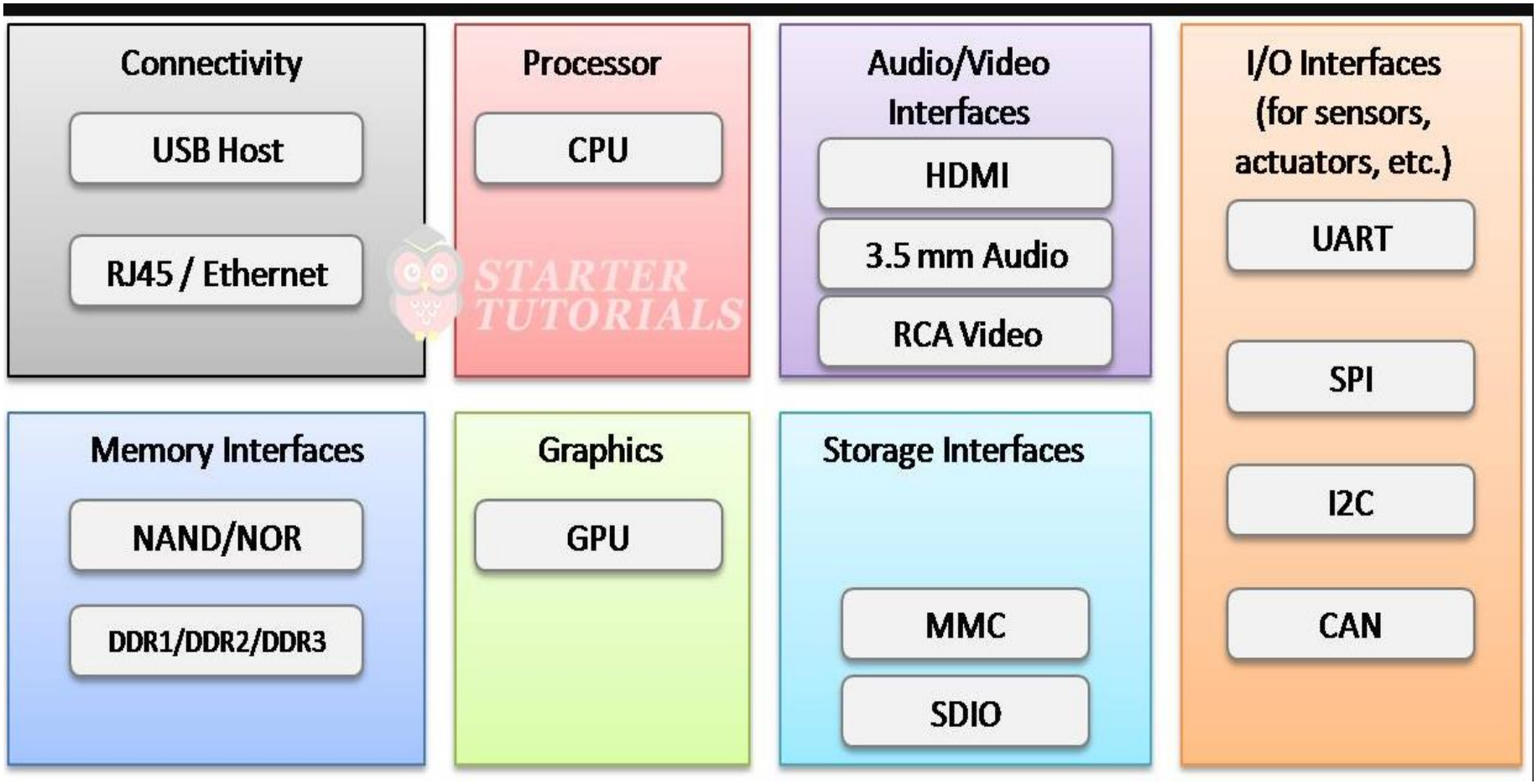
I/O interfaces for sensors

Interface for Internet connectivity

Memory and storage connectivity interfaces

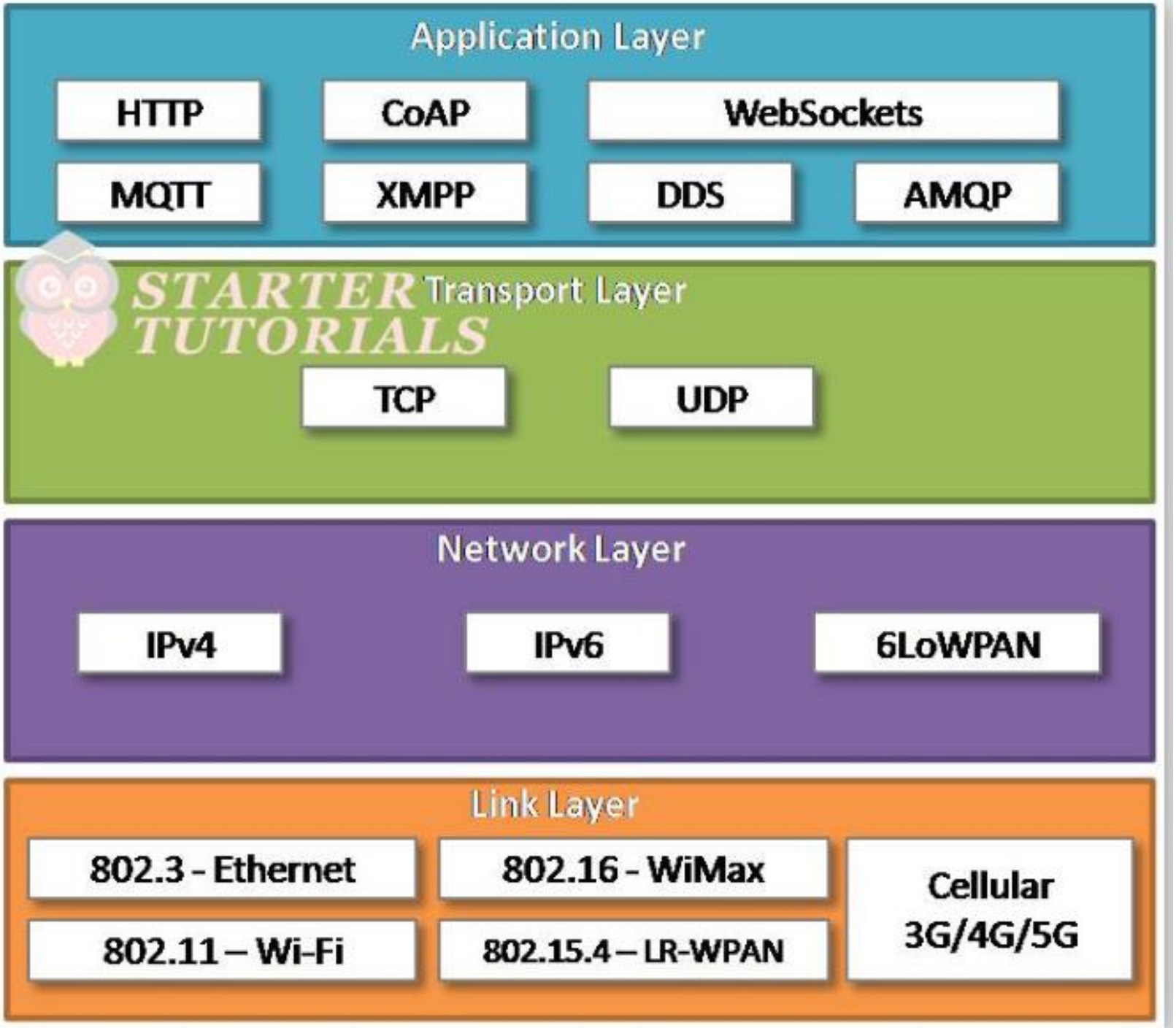
Audio/video interfaces

The generic block diagram of an IOT devices



Physical Design of IoT: IoT Protocols

A protocol is a set of rules that governs the communication between two or more devices. A protocol defines the rules, syntax, semantics and synchronization of communication and possible error recovery methods

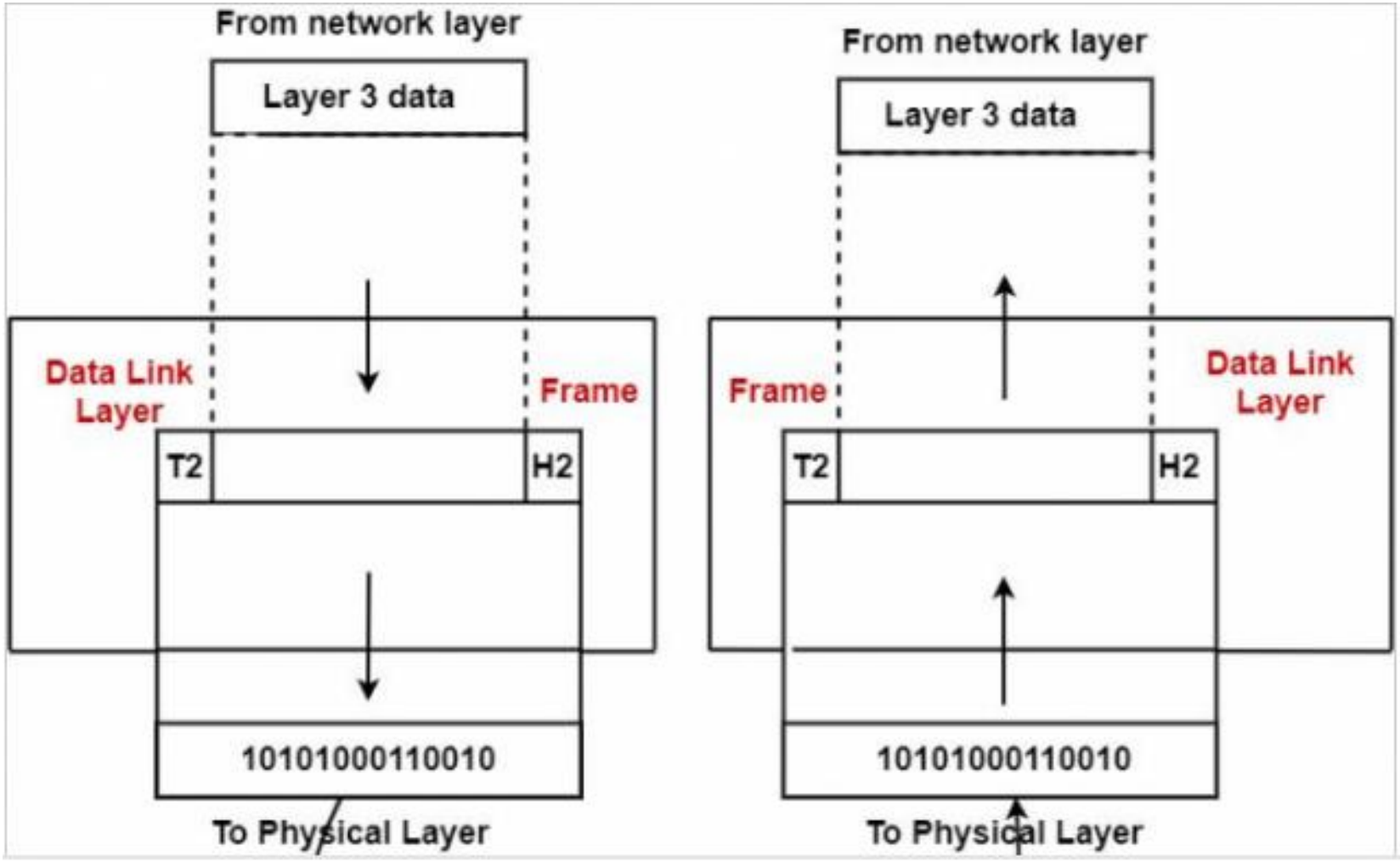

Application Layer
HTTP
CoAP
WebSockets
MQTT
XMPP
DDS
AMQP
STARTER TUTORIALS
Transport Layer
TCP
UDP
Network Layer
IPv4
IPv6
6LoWPAN
Link Layer
802.3 - Ethernet
802.16 - WiMax
802.11 – Wi-Fi
802.15.4 – LR-WPAN
Cellular 3G/4G/5G

Link Layer

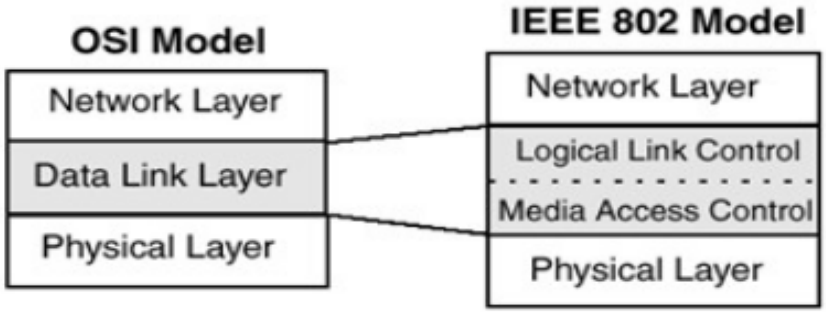
Link-layer protocols are the type of data transmission protocol used to help send data over the physical layer. They also determine how devices signal and code packets on the network.

The data link layer is **the protocol layer in a program that handles the moving of data into and out of a physical link in a network**.

- The primary service of the data link layer is to support error-free transmission.



Module 4.0: Data Link Layer



Ch 10- 11

- The Logical Link Control **(LLC)** sublayer
  - Framing
  - Flow Control
  - Error Control
- The Media Access Control **(MAC)** sublayer
  - Random Access (CSMA)
  - Token Passing

It controls the transmission of data packets via remotely shared channels.

LPWAN

- LPWAN or Low-power wide area networks is a media access control (MAC) layer protocol designed for public networks in large-scale with a single operator. It is used for secure mobile bi-directional communication in wireless battery operated devices. and is ideal where low power and long range is needed with millions and millions of devices connected.

Wi-Fi (IEEE 802.11)

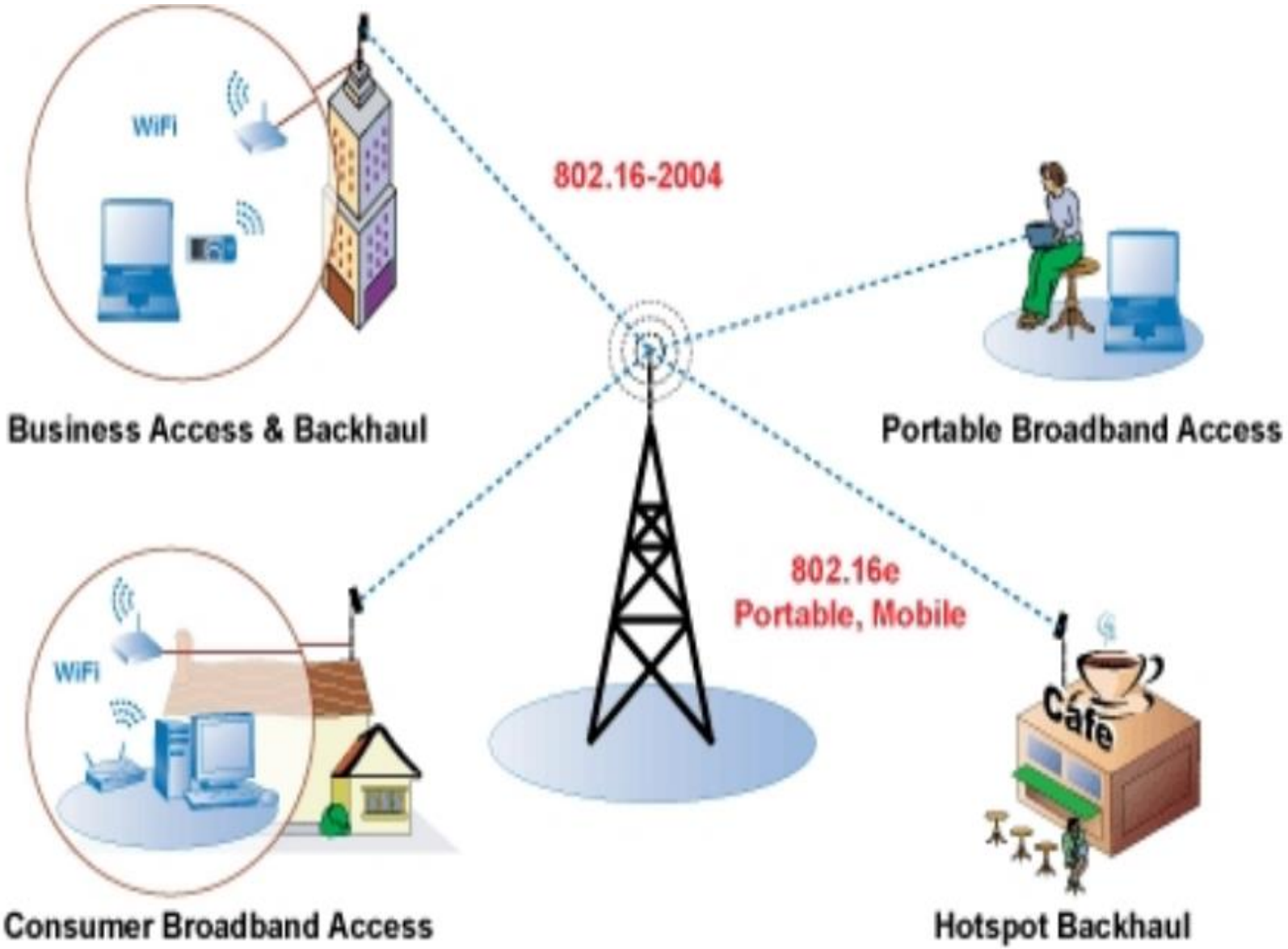
- It is a technology developed for electronic devices to connect to a wireless Local Area Network (WLAN). It has various encryption technologies WEP, WPA, WPA2, etc., and can be password protected. It has been used widely in the IoT ecosystem.

WiMAX

- Definition: WiMAX, the Worldwide Interoperability for Microwave Access, is **a telecommunications technology aimed at providing wireless data over long distances in a variety of ways, WiMAX basically supports point to multi point broadband wireless access.**

- WiMAX Technology has very high speed
- Fairly long distance communication is possible through WiMAX Technology. WiMAX changes the last mile problem for broadband in the same way as Wi-Fi has changed the last hundred feet of networking.
- Available bandwidth of WiMAX is shared between users.
- WiMAX's subscriber stations are small and can be mounted on rooftops.
- The WiMAX data is transmitted over air, thus reducing cabling cost.

Wimax

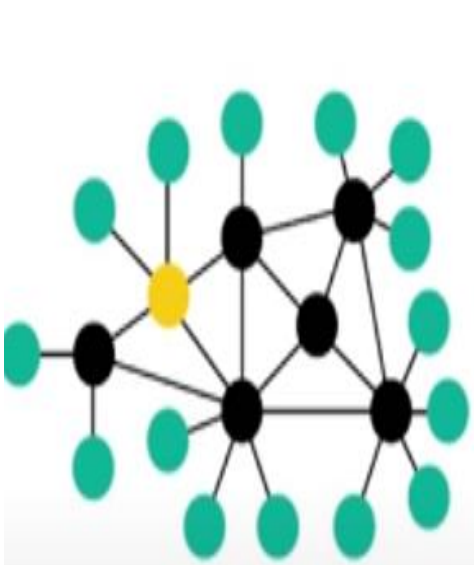

WiFi
802.16-2004
Business Access & Backhaul
Portable Broadband Access
802.16e
Portable, Mobile
WiFi
Café
Consumer Broadband Access
Hotspot Backhaul

ZigBee

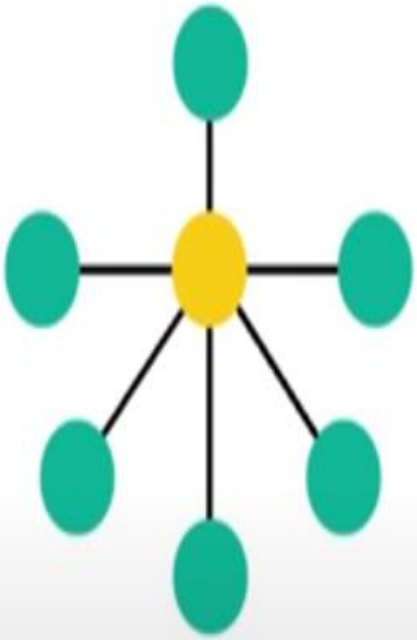
- It is an IEEE 802.15.4 standard-based protocol for personal area networks with short range, low power, and low data rate wireless data transfer. ZigBee is less expensive than other wireless personal area networks and also simple. It is called a mini version of Wi-Fi as they are both often used in similar applications in terms of household-based wireless communication.
- The ZigBee Network Protocol follows IEEE 802.15.4 standards for Physical and MAC layers, along with its own Network and Application layers.

Zigbee

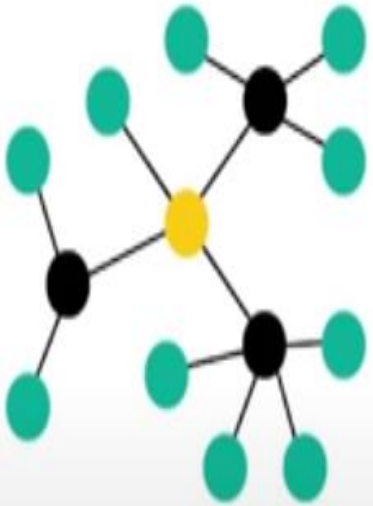
- Used to connect battery operated devices.



MESH TOPOLOGY



STAR TOPOLOGY



CLUSTER TOPOLOGY

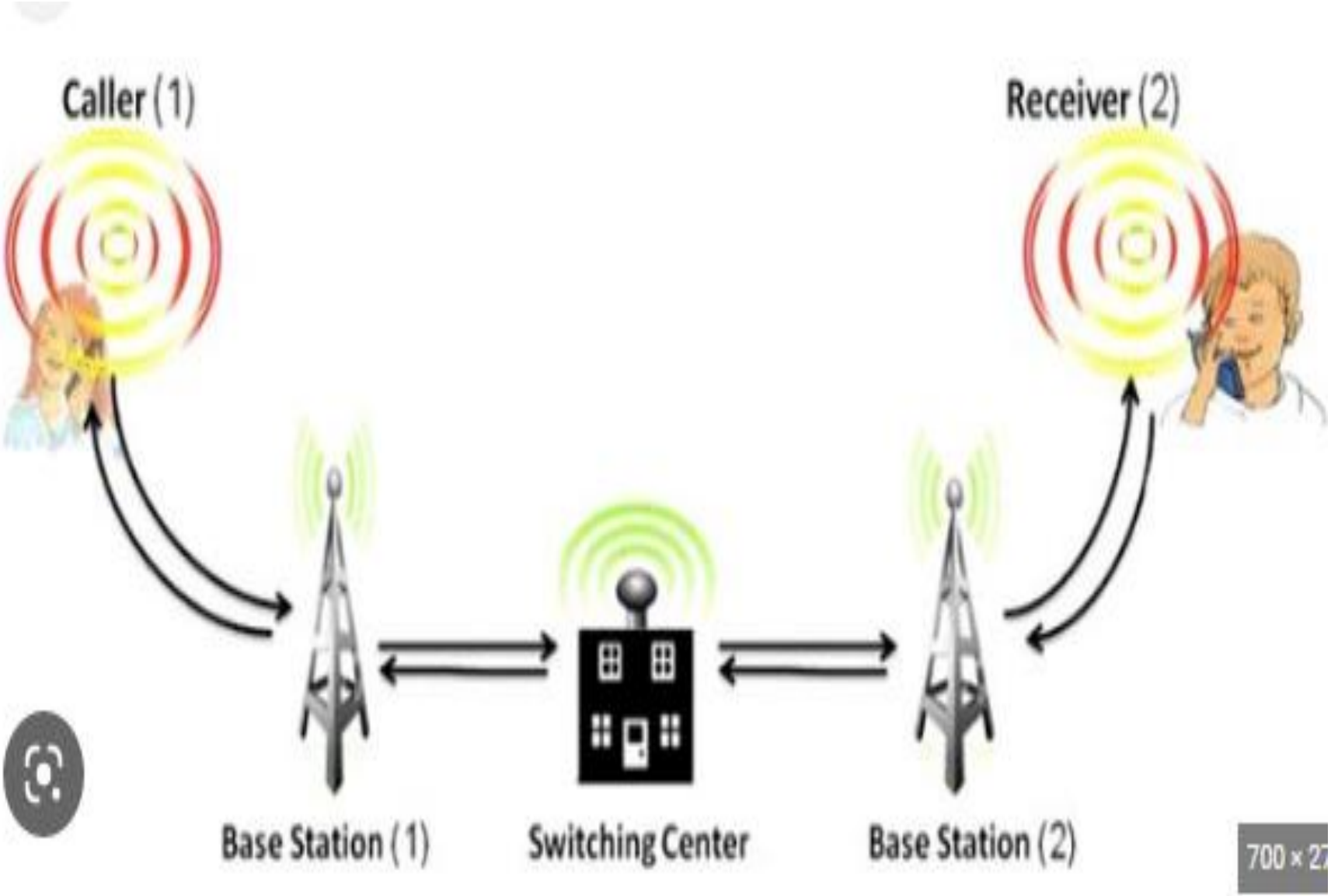
TECHNOLOGY	DATA RATE
	≥1Gbps
	1-3 Mbps
	250 Kbps

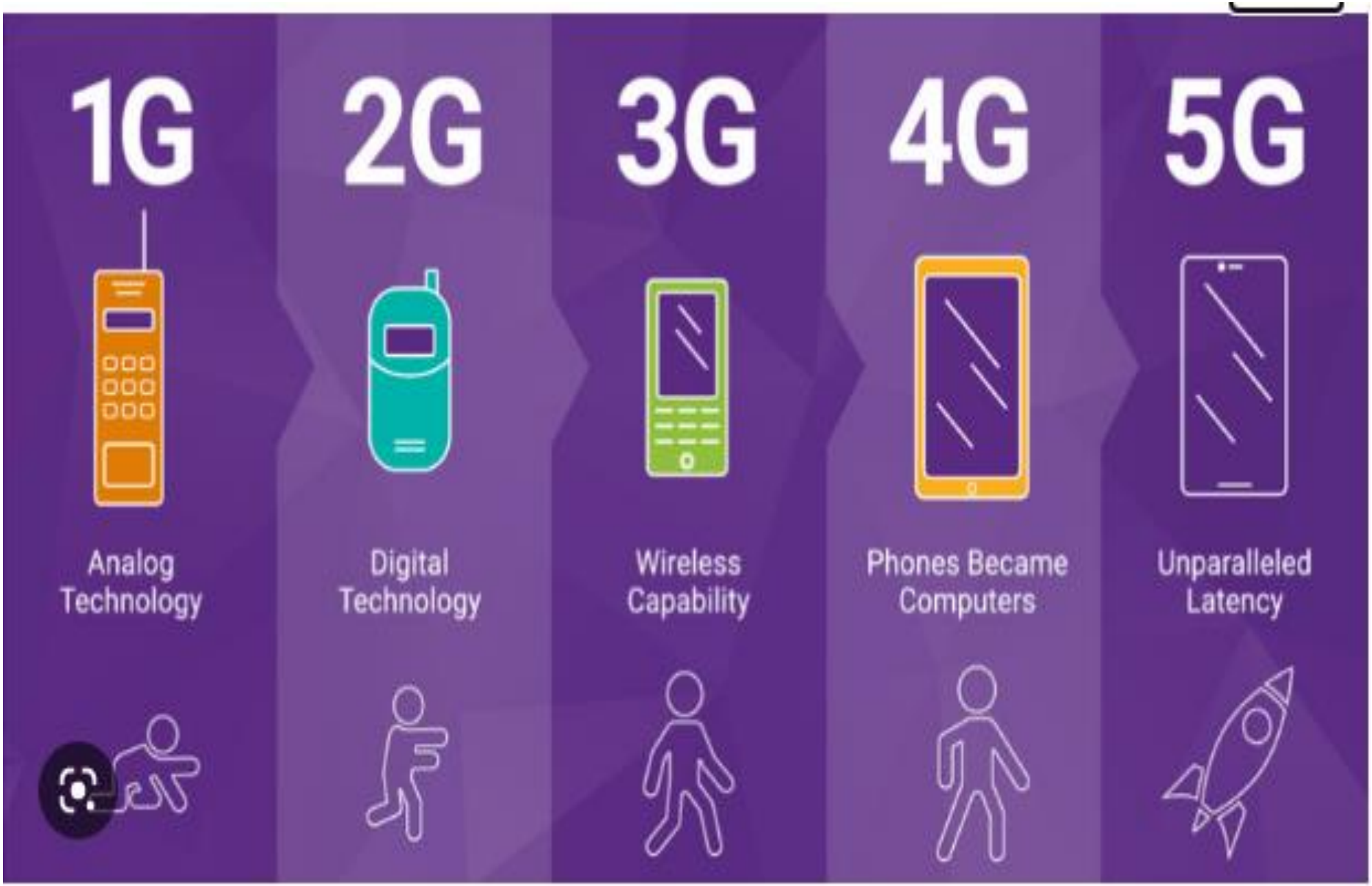
Bluetooth

- Bluetooth is **a wireless technology that uses a radio frequency to share data over a short distance, eliminating the need for wires**. You can use Bluetooth on your mobile device to share documents or to connect with other Bluetooth-enabled devices.
- Bluetooth is wireless technology that is used for sharing of data between two devices placed over short distances using a short wavelength such as ultra-high-frequency, radio waves in scientific, industrial and medical radio bands in the range of 2.400 to 2.485 GHz.

2G/3G/4G

- 3G allows additional features such as mobile Internet access, video calls and mobile TV. While the main function of 2G technology is the transmission of information through voice calls, 3G is all about transferring of information through data such as images and video.
- According to digital trends, 3G can reach network speeds of 7.2 mbps, 4G can reach network speeds of 150 mbps and 5G will eventually reach speeds in excess of 1gbps (with a theoretical maximum of 20gbps!)


Caller (1)
Receiver (2)
Base Station (1)
Switching Center
Base Station (2)


1G
Analog Technology
2G
Digital Technology
3G
Wireless Capability
4G
Phones Became Computers
5G
Unparalleled Latency

Standard	Name	Medium	Speed	Range
IEEE 802.3	Ethernet	Coaxial/ Twisted-pair/ Fiber optic	10 Mbps – 40 Gbps +	100 m
IEEE 802.11	Wi-Fi	Radio Waves	1 Mbps – 6.75 Gbps	30 m
IEEE 802.16	WiMax	Radio Waves	1.5 Mbps – 1 Gbps	50 Km
IEEE 802.15.4	LR-WPAN	Bluetooth	40 Kbps – 250 Kbps	10 m
2G/3G/4G	Cellular	Radio Waves	9.6 Kbps – 100 Mbps	16 Km

Network layer

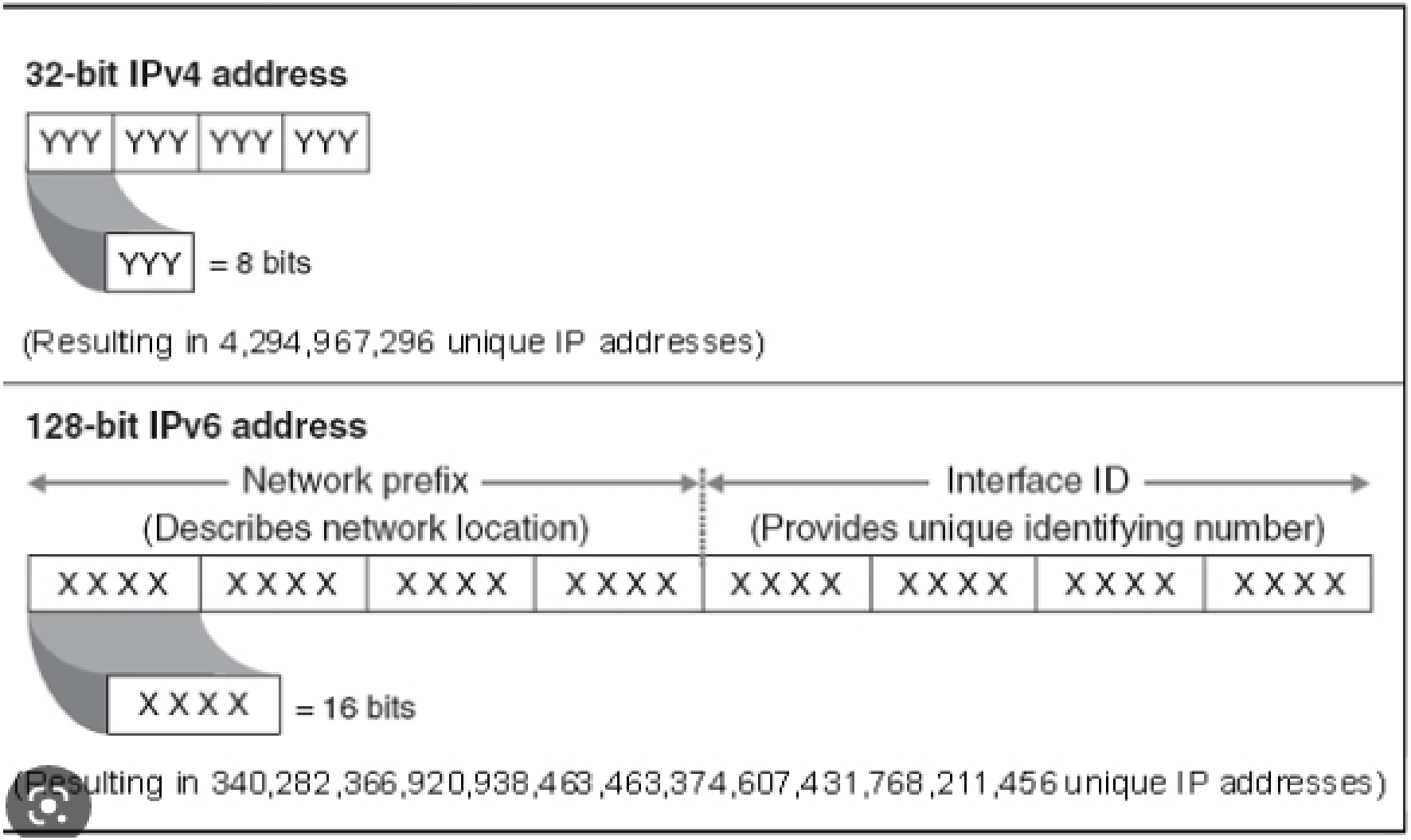
- This layer is used to send data from a source network to a destination network. For this, IPv4 and IPv6 protocols are used for host identification, which transfers data in packets.

IPV4

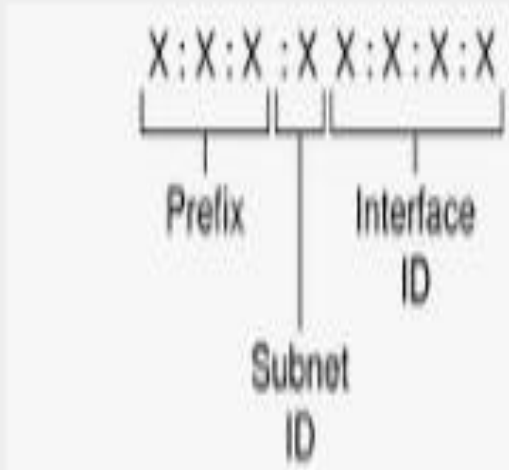
- **Internet Protocol version 4** (IPv4) is the fourth version of the standard that routes Internet traffic and other packet-switched networks.
- The more common format, known as dotted quad or dotted decimal, is x.x.x.x, where each x can be any value between 0 and 255. For example, 192.0.2.146 is a valid IPv4 address.

IPV6

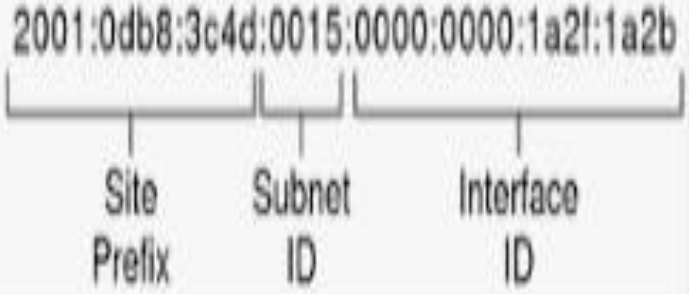
Figure 1: Comparison of IPv6 and IPv4 Address Scheme

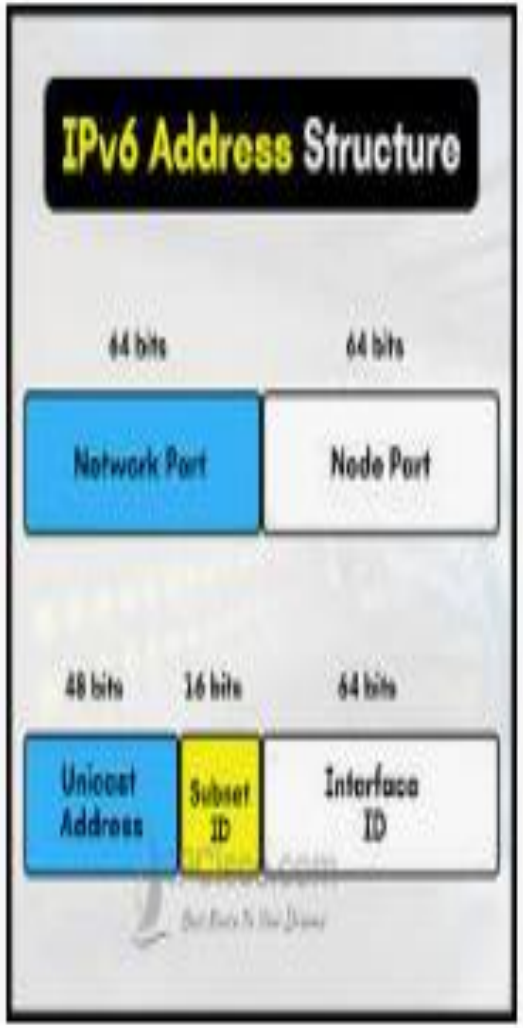


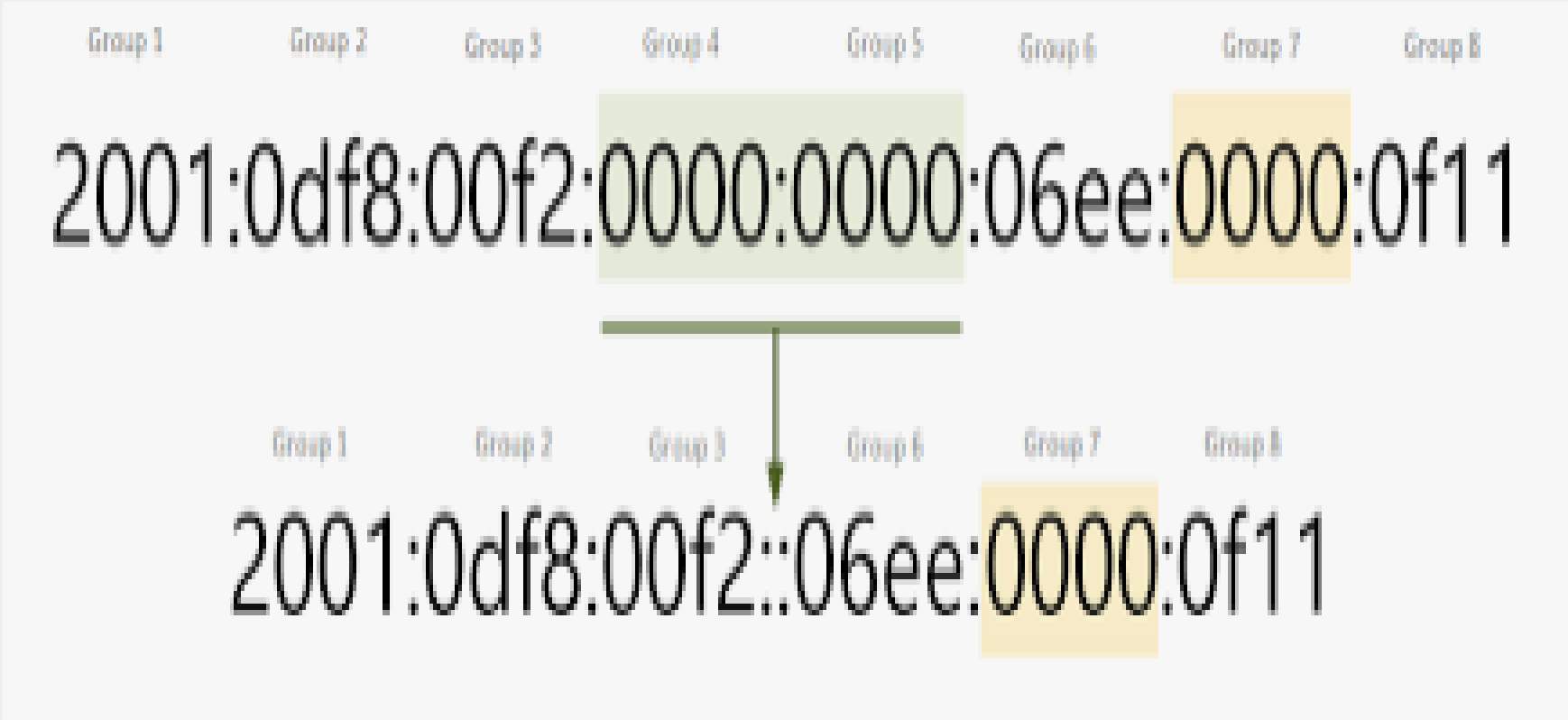
IPv6 Address Representation ...



Example:






Group 1
Group 2
Group 3
Group 4
Group 5
Group 6
Group 7
Group 8
2001:0df8:00f2:0000:0000:06ee:0000:0f11
Group 1
Group 2
Group 3
Group 6
Group 7
Group 8
2001:0df8:00f2::06ee:0000:0f11

Transport Layer Protocols

- This layer is responsible for data flow control and error handling, ensuring that there are rules in place to deal with errors. This layer also provides end-to-end message transfer capability, independent of the underlying network infrastructure. It provides essential connectivity between the two nodes on either end of the point-to-send-point-receive model used by key protocols such as TCP/IP.

Application Layer Protocols

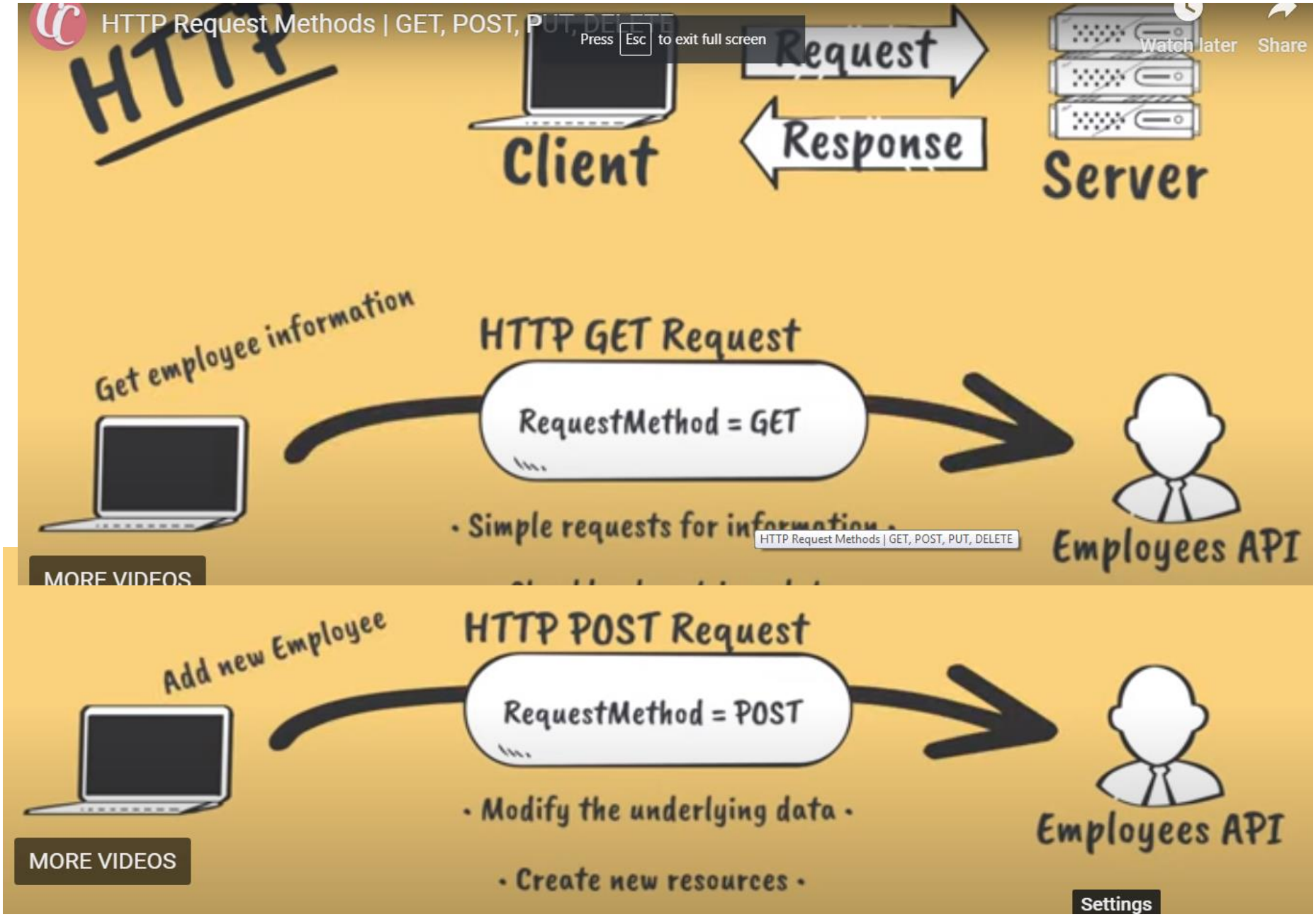
- On this layer, protocols use an application interface to define how the data can be sent over the network. These protocols include HTTP, XMPP, WebSocket, DDS, MQTT, and AMQP.

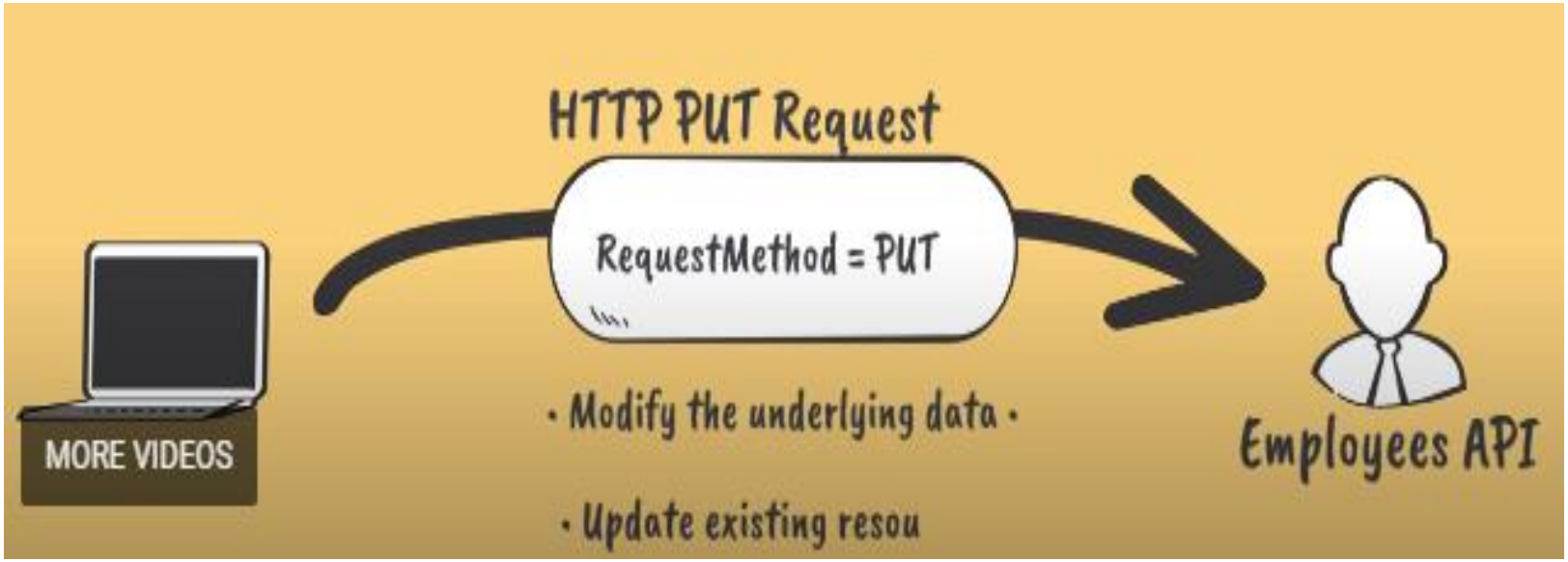
HTTP

- HTTP (hypertex transport protocol) is **a protocol for fetching resources such as HTML documents**. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser.

HTTP

- Port number 80
- Itself not reliable but use TCP to achieve reliability
- Inband protocol
- Stateless
- HTTP 1.0 non persistent
- HTTP 1.1 persistent
- Commands (GET,PUT,POST,DELETE,HEAD,OPTIONS)

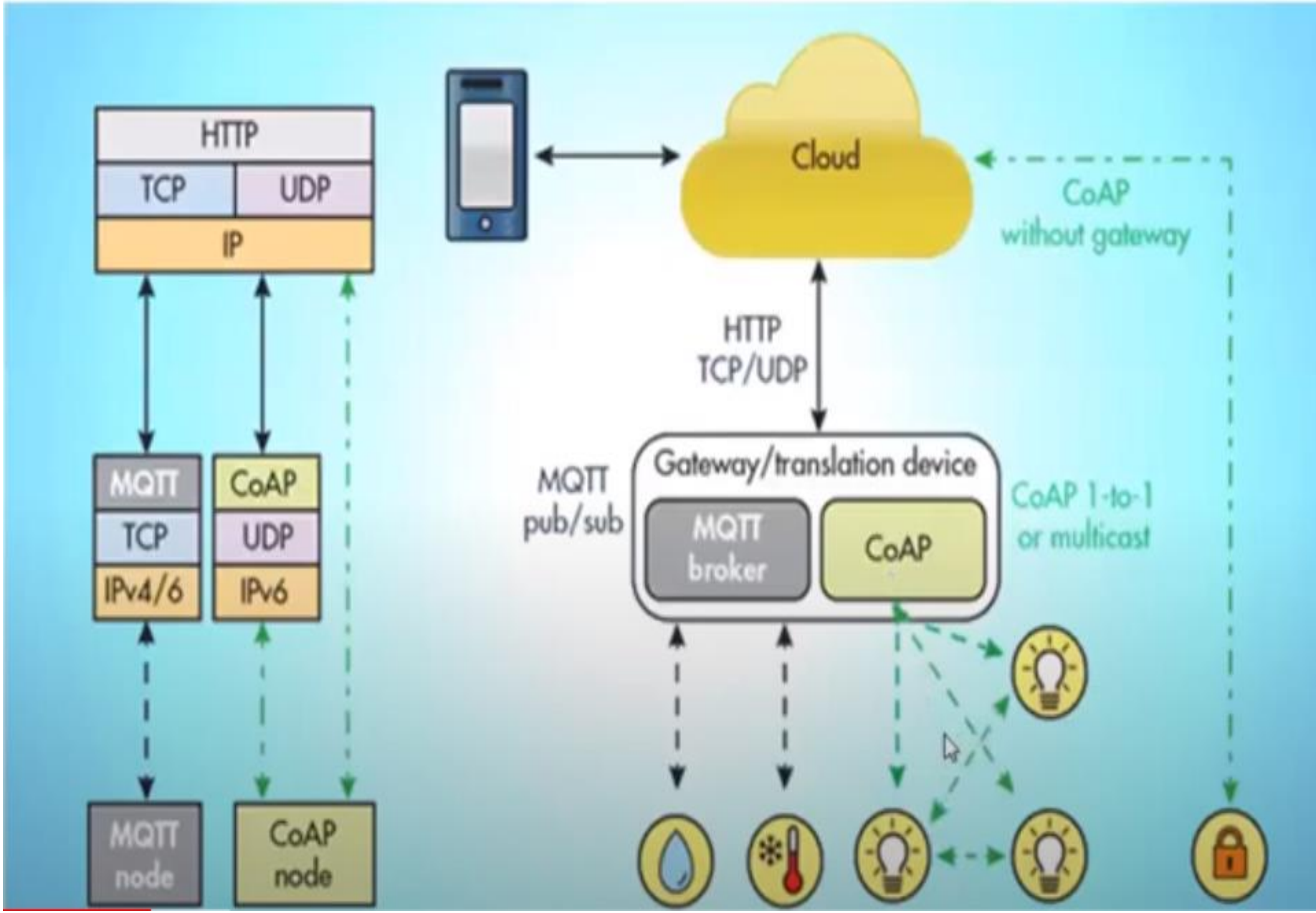

HTTP Request Methods | GET, POST, PUT, DELETE
Press Esc to exit full screen
Watch later
Share
HTTP
Request
Response
Client
Server
Get employee information
HTTP GET Request
RequestMethod = GET
• Simple requests for information •
Employees API
MORE VIDEOS
Add new Employee
HTTP POST Request
RequestMethod = POST
• Modify the underlying data •
• Create new resources •
Employees API
MORE VIDEOS
Settings


HTTP PUT Request
RequestMethod = PUT
• Modify the underlying data •
• Update existing resou
MORE VIDEOS
Employees API


HTTP DELETE Request
RequestMethod = DELETE
MORE VIDEOS
Employees API

COAP

- CoAP Protocol (Constrained Application Protocol) is a web-based protocol that has been specifically designed to connect small, semi-intelligent devices to the Internet of Things (IoT).
- Methods:
- GET, PUT,POST, and DELETE

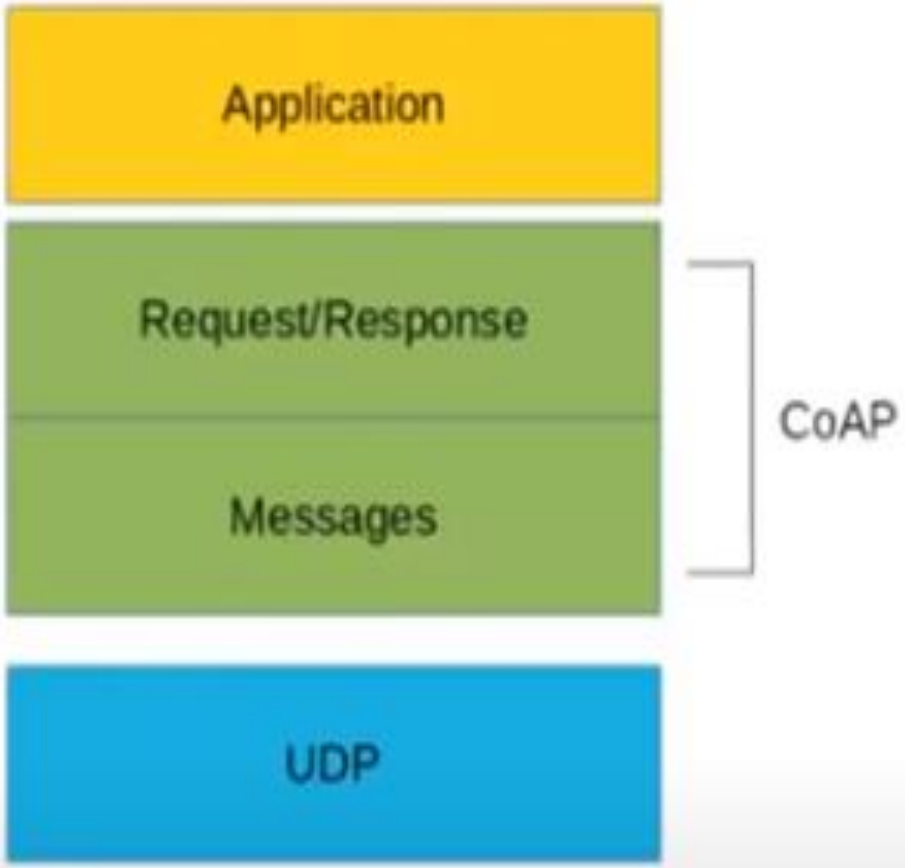

HTTP
TCP
UDP
IP
MQTT
TCP
IPv4/6
CoAP
UDP
IPv6
MQTT node
CoAP node
Cloud
CoAP without gateway
HTTP TCP/UDP
MQTT pub/sub
Gateway/translation device
MQTT broker
CoAP
CoAP 1-to-1 or multicast

CoAP layers

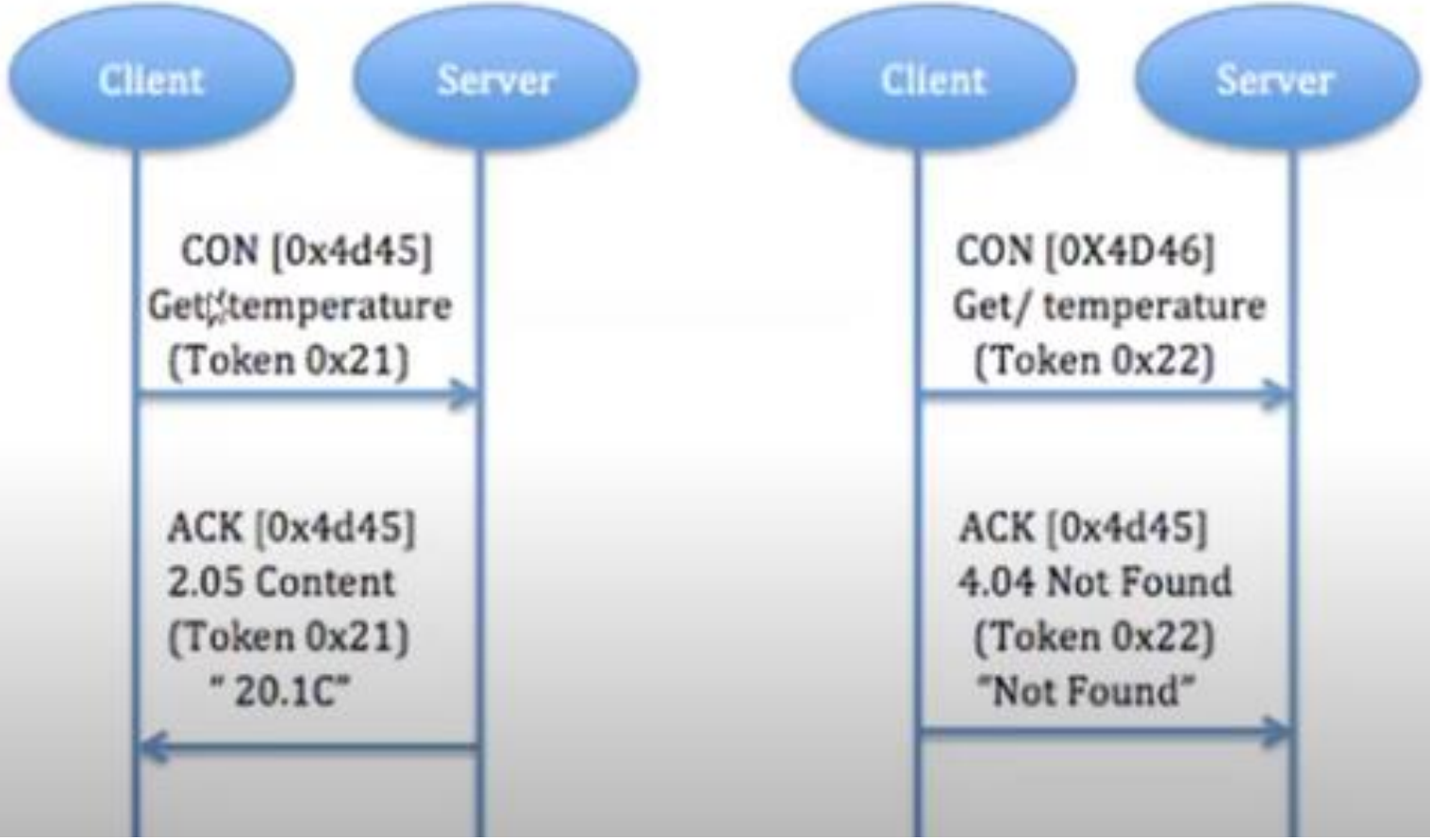
- There are two different layers that make CoAp

  - Messages
  - Request/Response.

The Messages layer Deals with UDP and with asynchronous messages.



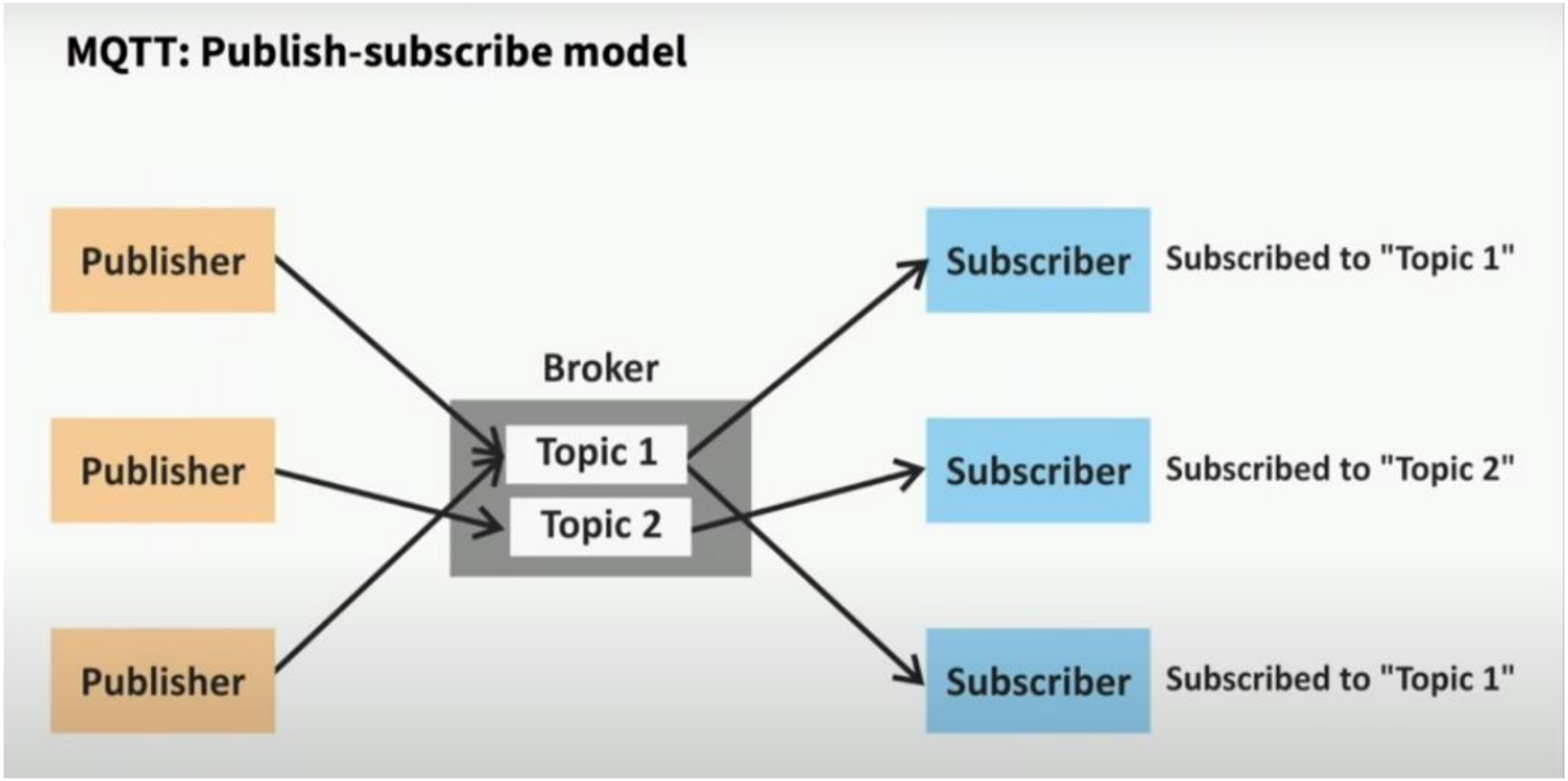
If the server has troubles managing the incoming request, it can send back a error message instead of the Acknowledge message (ACK):



MQTT

- It requires minimal resources since it is **lightweight and efficient**
- Support **bi-directional** messaging between device and cloud
- Can **scale to millions** of connected devices
- Support **reliable message** delivery
- Works well over **unreliable networks**
- **Security enabled**

MQTT


MQTT: Publish-subscribe model
Publisher
Publisher
Publisher
Broker
Topic 1
Topic 2
Subscriber
Subscribed to "Topic 1"
Subscriber
Subscribed to "Topic 2"
Subscriber
Subscribed to "Topic 1"

Methods

Connect

Disconnect

Publisher

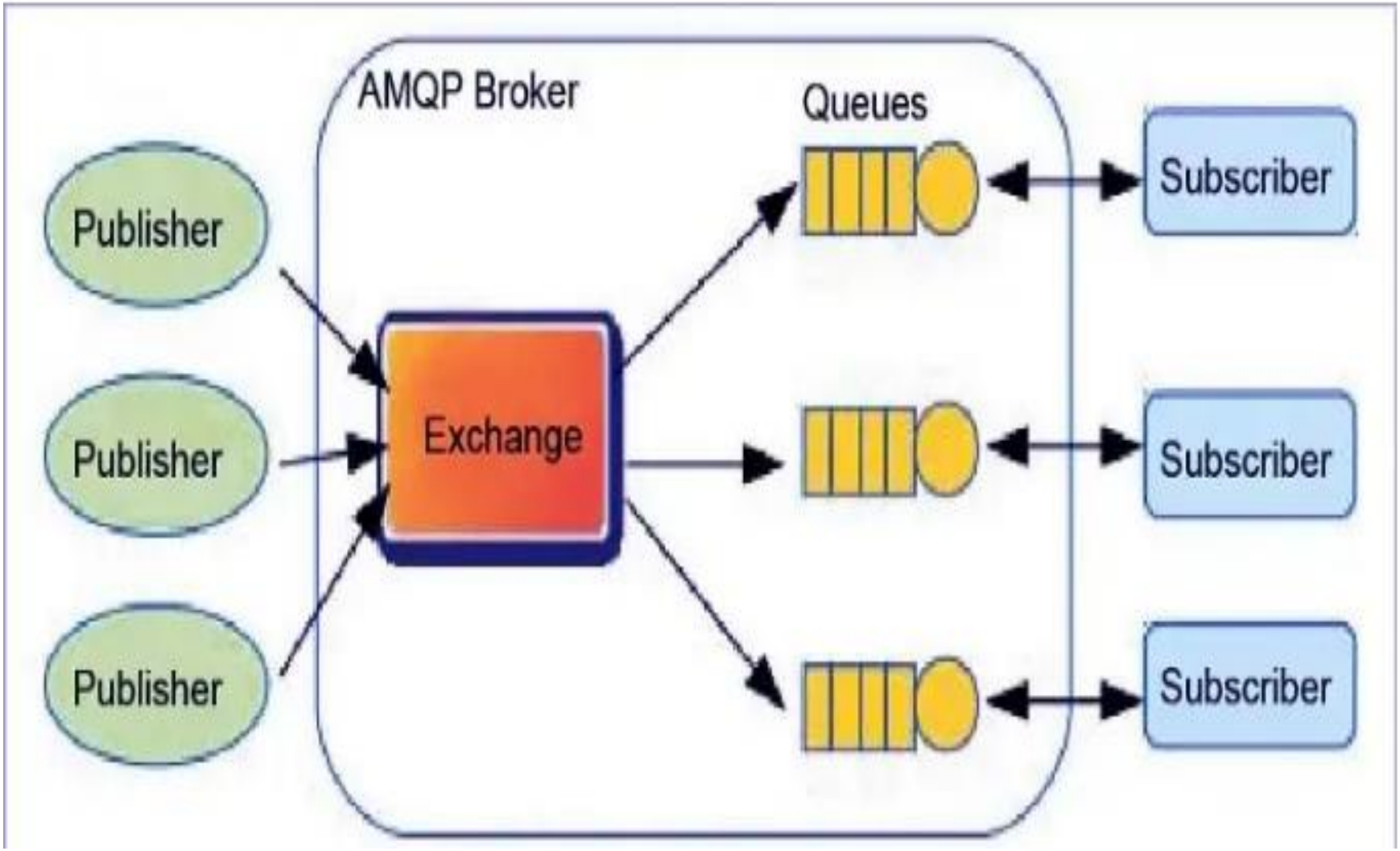
Subscriber

AMQP

- The Advanced Message Queuing Protocol (AMQP) is **an open standard for passing business messages between applications or organizations**. It connects systems, feeds business processes with the information they need and reliably transmits onward the instructions that achieve their goals.

Components of AMQP

- Exchange: Receives messages from publisher primarily based programs and routes them to ‘message queues’.
- Message Queue: Stores messages
- Binding: States the connection between the message queue and the change.


AMQP Broker
Queues
Publisher
Publisher
Publisher
Exchange
Subscriber
Subscriber
Subscriber

Types of frames

- Open
- Close
- Begin
- End
- Attach (link)
- Detach
- Transfer
- Flow

Types of Exchanges

- Direct (route key= bound key)
- Fanout(msg route to all bound links)
- Topic (route key= routing pattern specified in binding)
- Header (route key= header attributes)

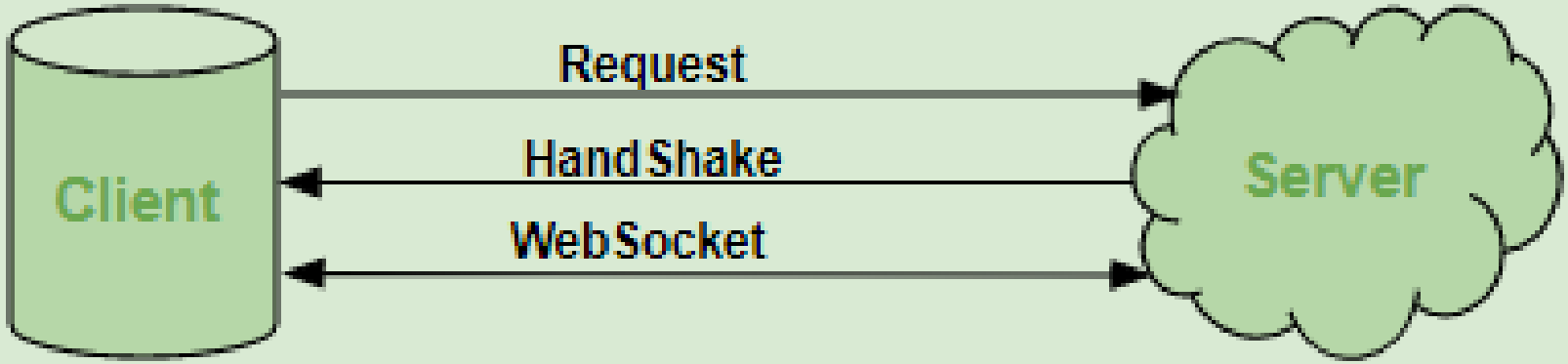
Websocket

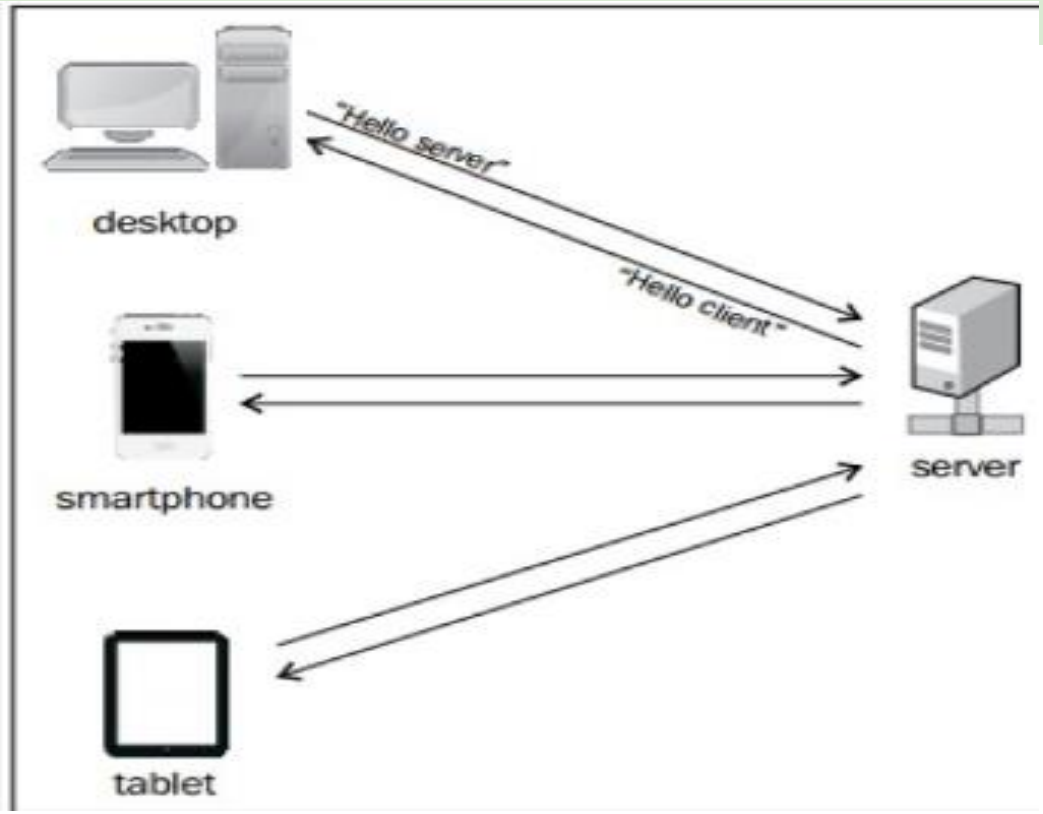
- WebSocket is **a bidirectional communication protocol that can send the data from the client to the server or from the server to the client by reusing the established connection channel**. The connection is kept alive until terminated by either the client or the server.
- WebSocket can be used if we want any real-time updated or continuous streams of data that are being transmitted over the network

methods

- Statefull
- Full duplex
- Single TCP connection
- No head overhead
- Suitable for real time application

WebSocket Connection





- **Real-time web application:** Real-time web application uses a web socket to show the data at the client end, which is continuously being sent by the backend server. In WebSocket, data is continuously pushed/transmitted into the same connection which is already open, that is why WebSocket is faster and improves the application performance.

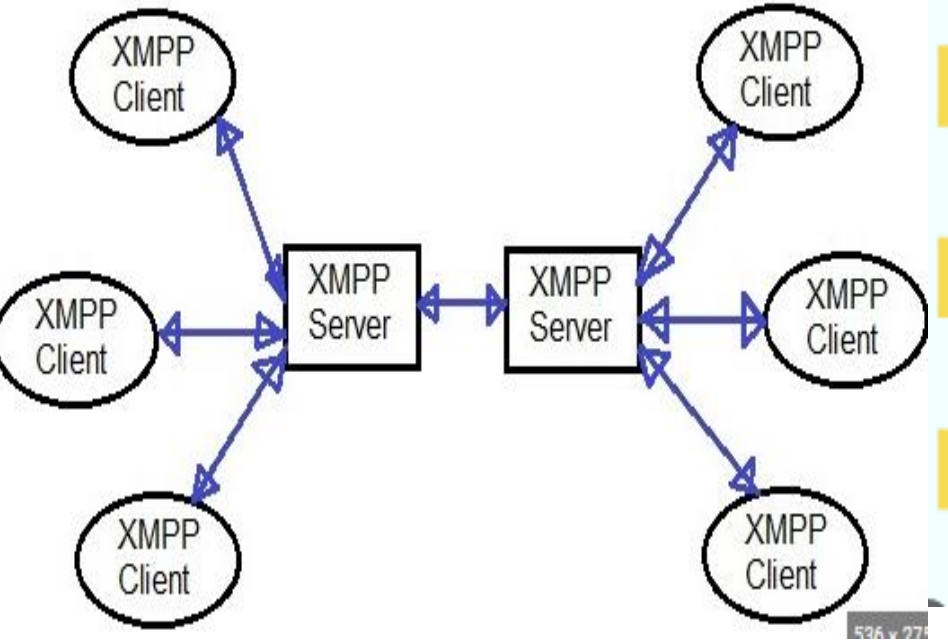
For e.g. in a trading website or bitcoin trading, for displaying the price fluctuation and movement data is continuously pushed by the backend server to the client end by using a WebSocket channel.

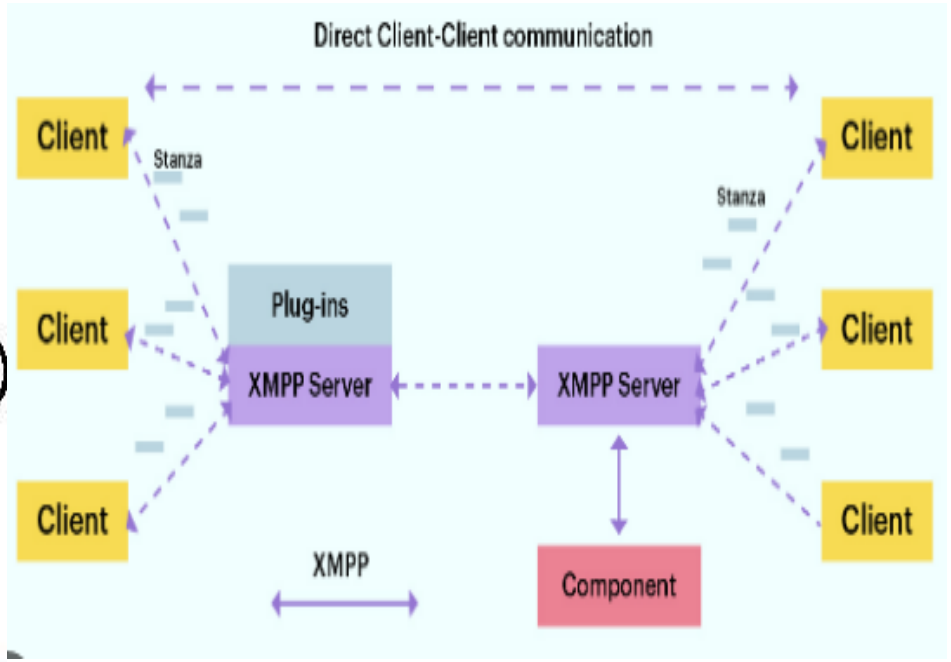
XMPP

- XMPP is the Extensible Messaging and Presence Protocol, a set of open technologies for instant messaging, presence, multi-party chat, voice and video calls, collaboration, lightweight middleware, content syndication, and generalized routing of XML data.
- Extensible Markup Language is a markup language and file format for storing, transmitting, and reconstructing arbitrary data

XMPP supports both client –server and server to server communication path

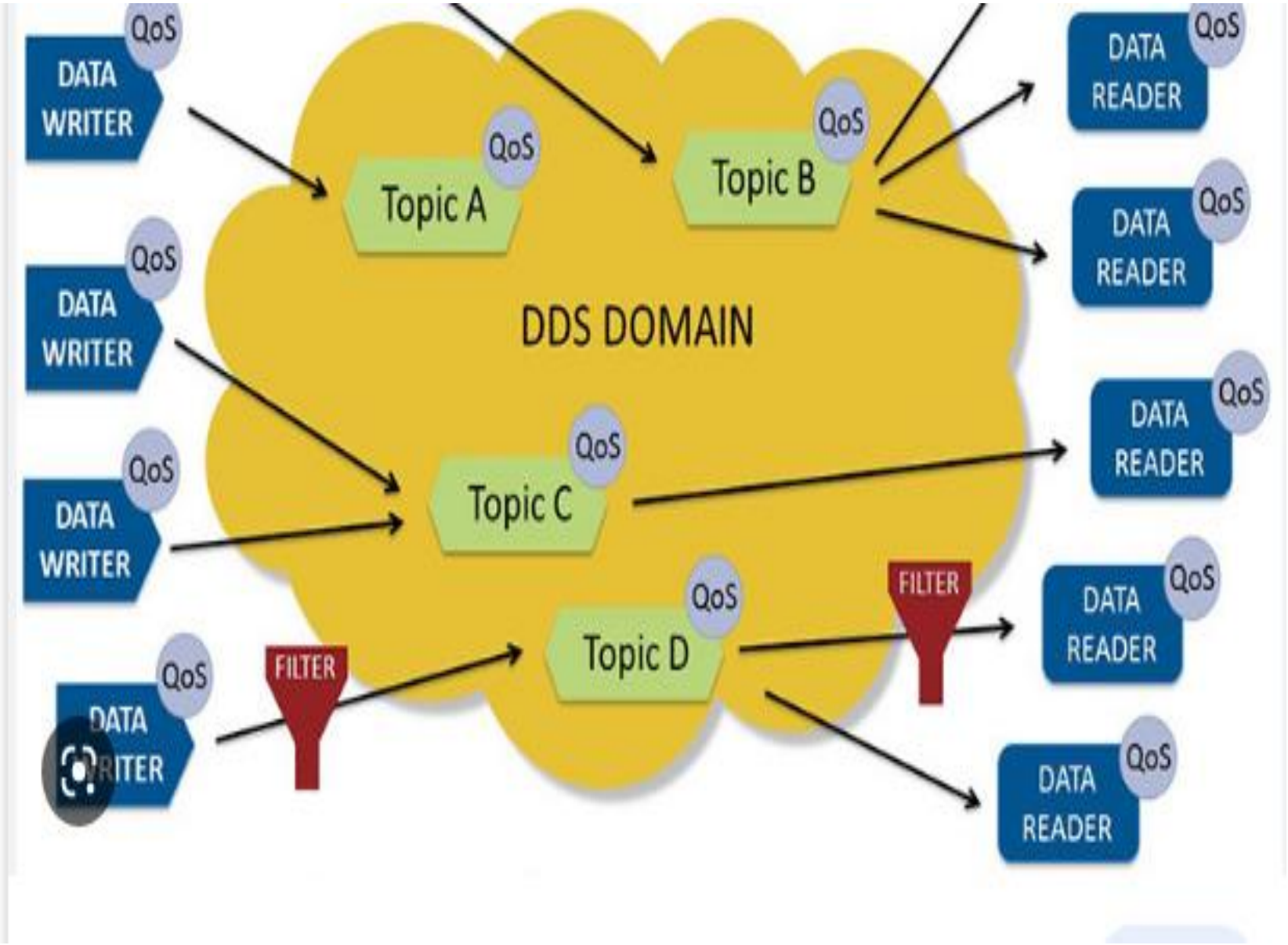
In IOT XMPP allows real time communication between IOT devices.





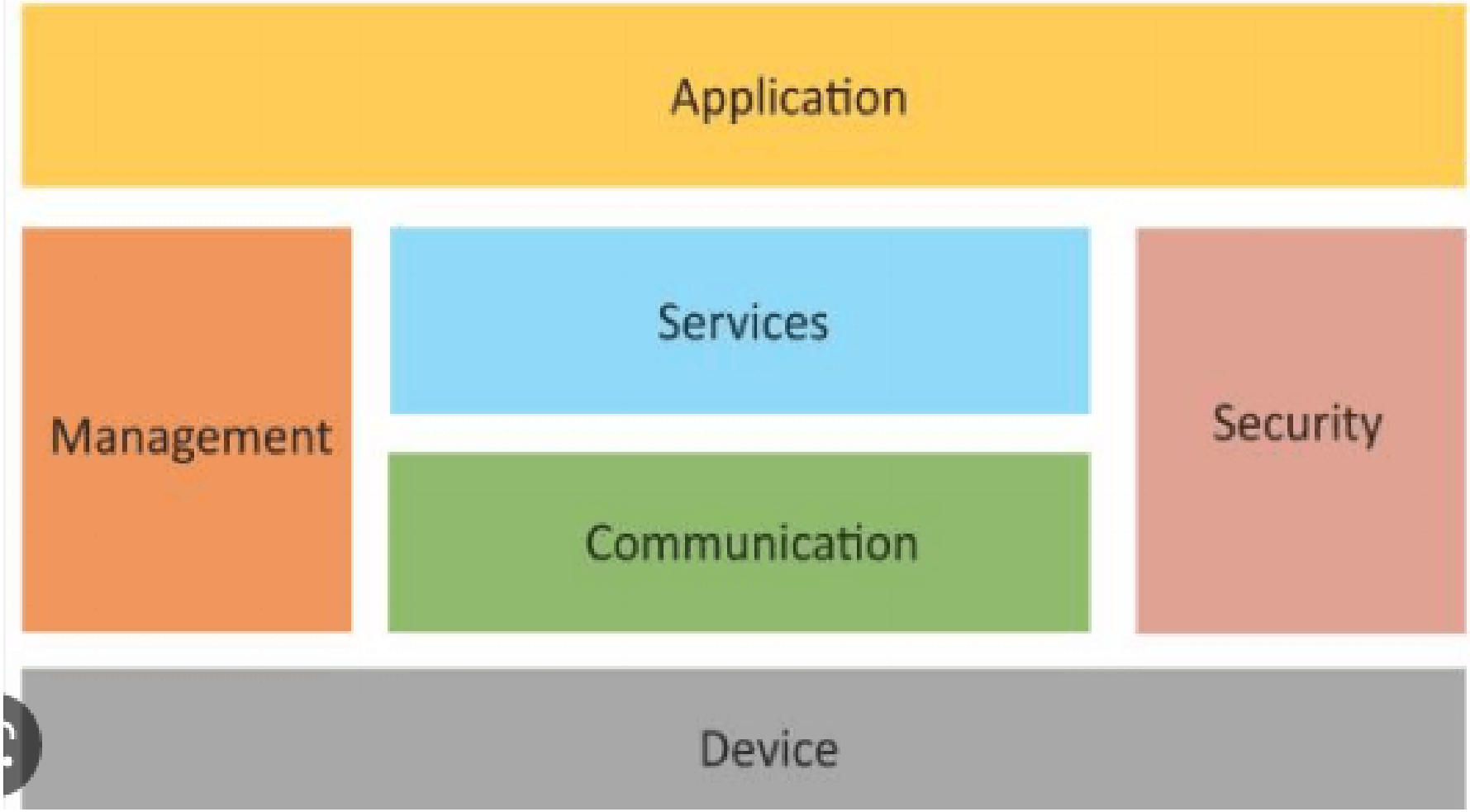
DDS

- DDS is another publish-subscribe protocol but it is different from MQTT which connects them to server but here DDS protocol is a Broker-less architecture that's why it is a high speed and high-performance protocol than MQTT as it is not dependent upon any intermediary system.
- DDS protocol uses broker less architecture in IoT (Internet of Things). DDS stands for Data Distribution Service. It is an IoT protocol developed for M2M (Machine to Machine) Communication by OMG (Object Management Group). It **enables data exchange via publish-subscribe methodology**.


DATA WRITER
QoS
DATA WRITER
QoS
DATA WRITER
QoS
DATA WRITER
QoS
FILTER
Topic A
QoS
Topic B
QoS
DDS DOMAIN
Topic C
QoS
Topic D
QoS
FILTER
DATA READER
QoS
DATA READER
QoS
DATA READER
QoS
DATA READER
QoS
DATA READER
QoS

- DDS is the proven protocol-of-choice when IoT system performance, scalability, robustness, reliability and quality-of-service are paramount
- OMG Data Distribution Service (DDS) is a proven international standard for data-connectivity that is ideal for business-critical Internet of Things (IoT) systems.
- The DDS is used in various applications such as **smart grid management, medical applications, robotics, power generation, simulation, testing,** defense sector include **navigation systems, weaponry and management, radar**.

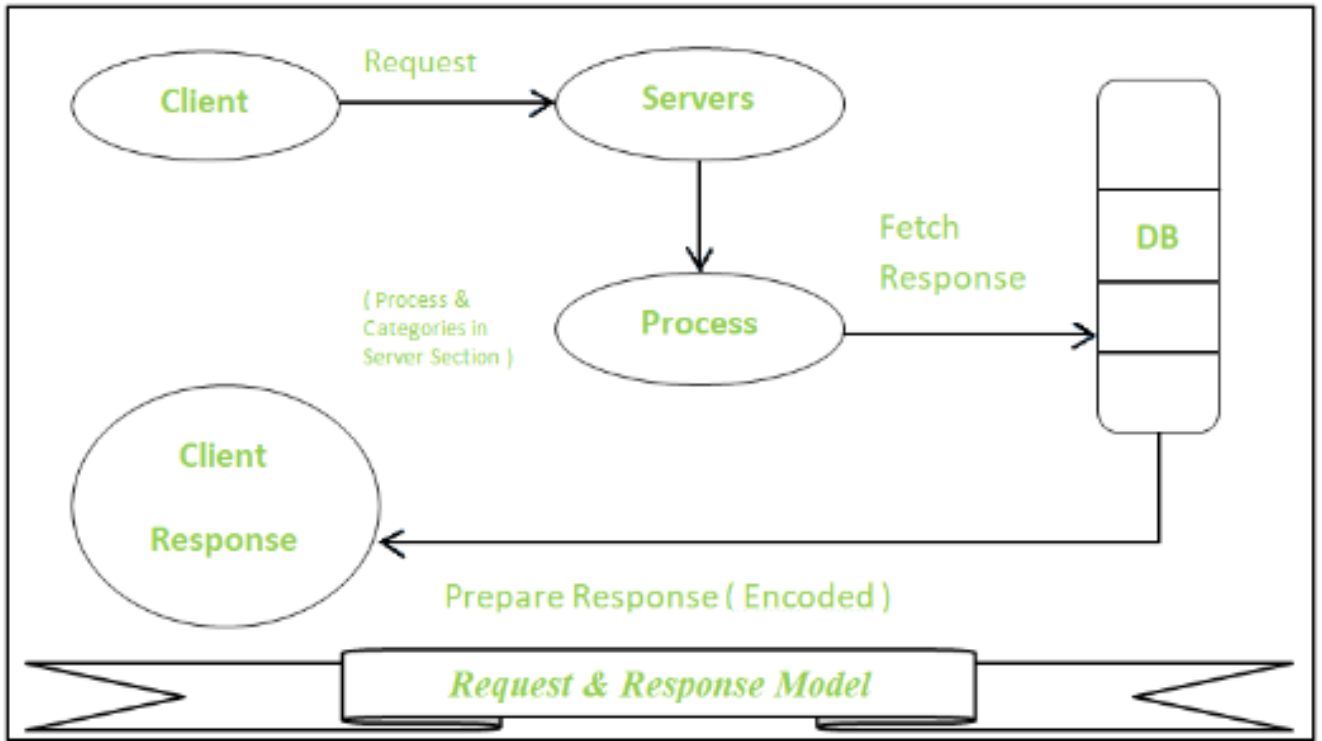
Logical Design of IoT: **IoT Functional Blocks**



IOT communication models

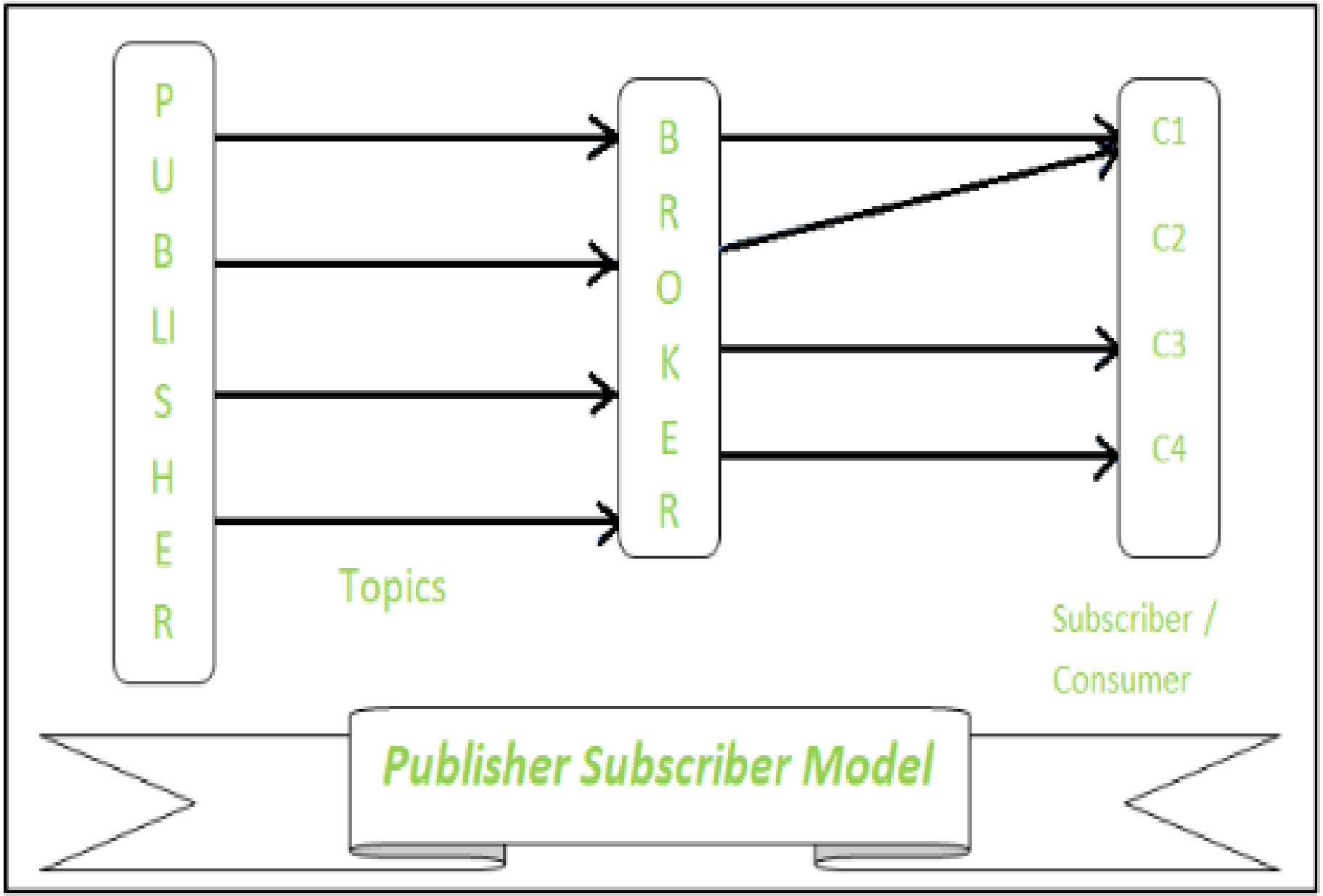
- it is important and useful to understand how various IoT devices communicate with each other.
- Request and Response model
- This model is stateless since the data between the requests is not retained and each request is independently handled.

In **Request-Response** communication model client sends a request to the server and the server responds to the request. When the server receives the request it decides how to respond, fetches the data retrieves resources, and prepares the response, and sends it to the client.



Publisher-Subscriber Model

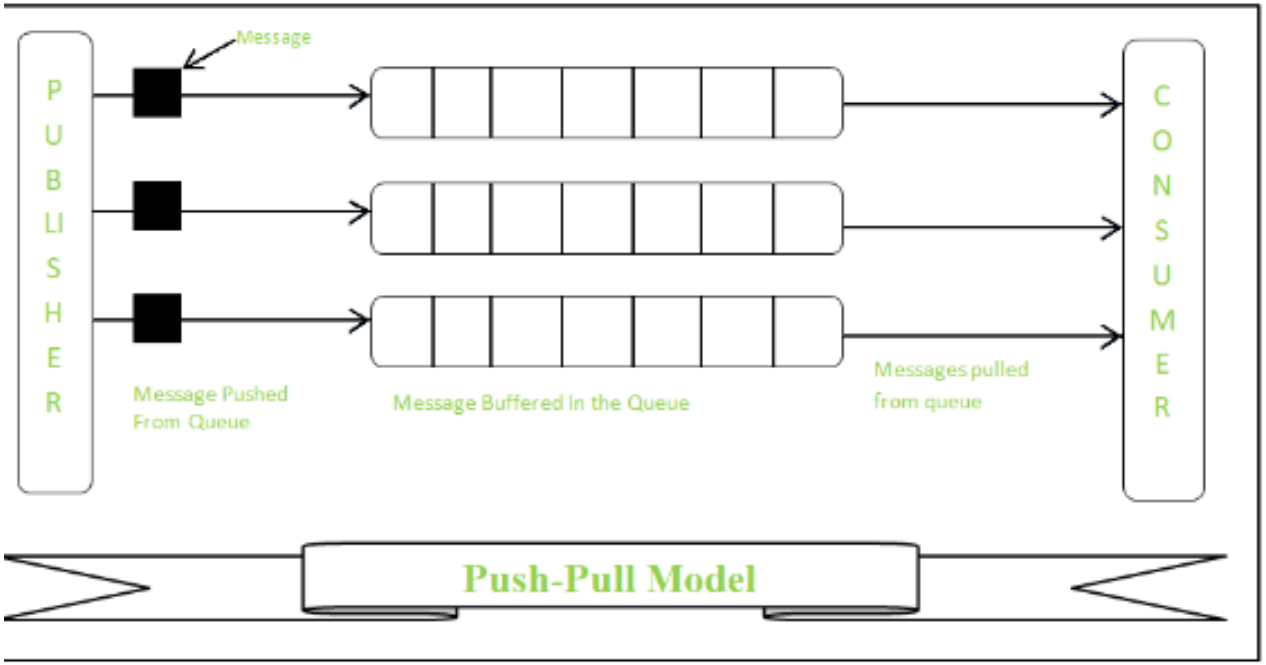
- **Publishers** are the source of data. It sends the data to the topic which are managed by the broker. They are not aware of consumers.
- **Consumers** subscribe to the topics which are managed by the broker.
- Hence, **Brokers** responsibility is to accept data from publishers and send it to the appropriate consumers.


PUBLISHER
BROKER
C1
C2
C3
C4
Topics
Subscriber / Consumer
Publisher Subscriber Model

Push-Pull Model

- **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.



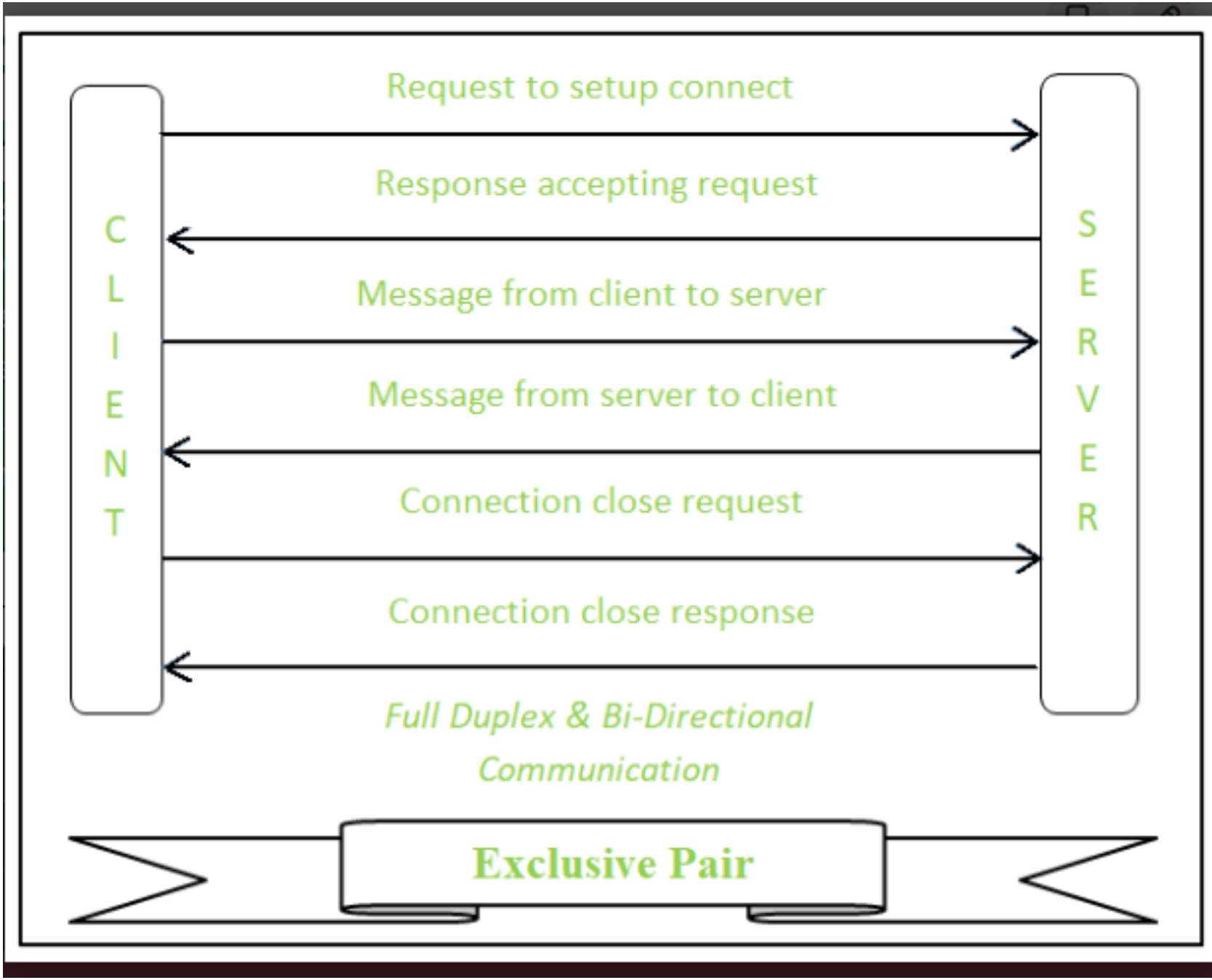
Exclusive Pair

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.


Request to setup connect
Response accepting request
Message from client to server
Message from server to client
Connection close request
Connection close response
CLIENT
SERVER
Full Duplex & Bi-Directional Communication
Exclusive Pair

IOT communication API (application programming interfacing)

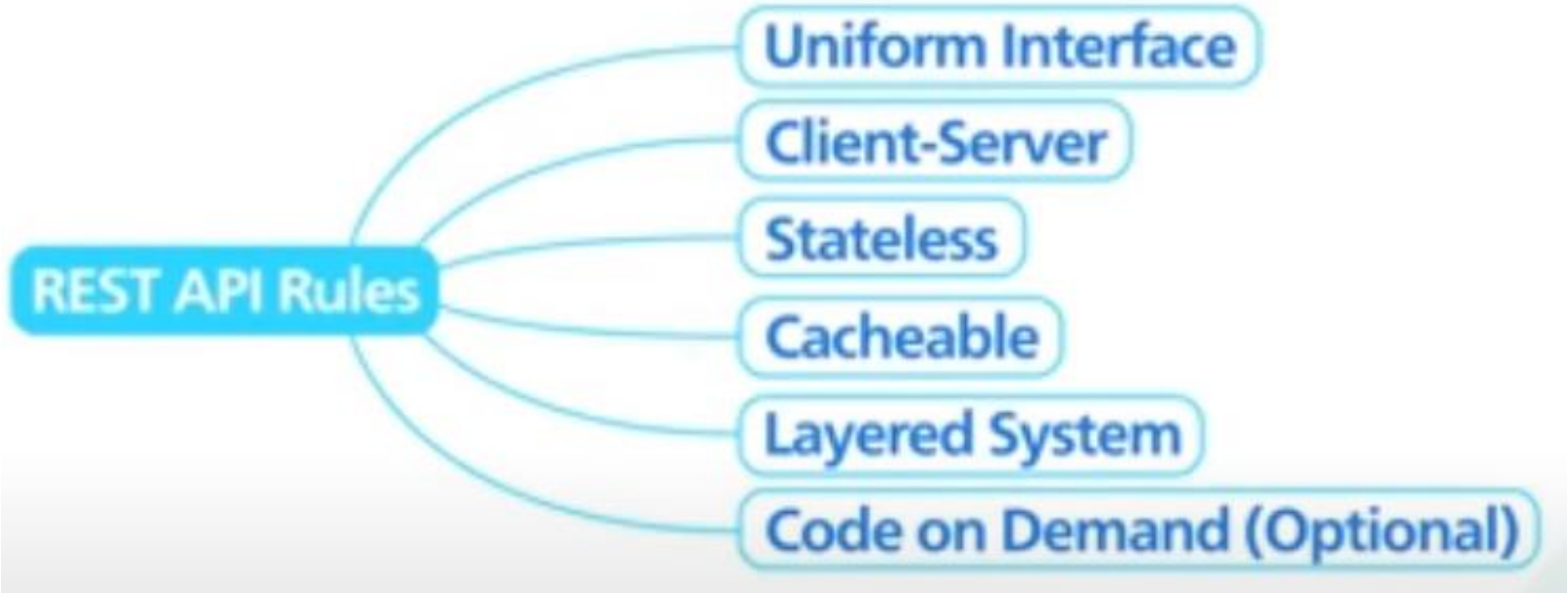
- API-set of function, protocols, routines and tools for building software application
- API Application In IoT. Application Programming Interface (API) is **an interfacing software platform that allows the exchange of any information or data and supports the interaction among different applications or any such intermediaries**

IOT communication API

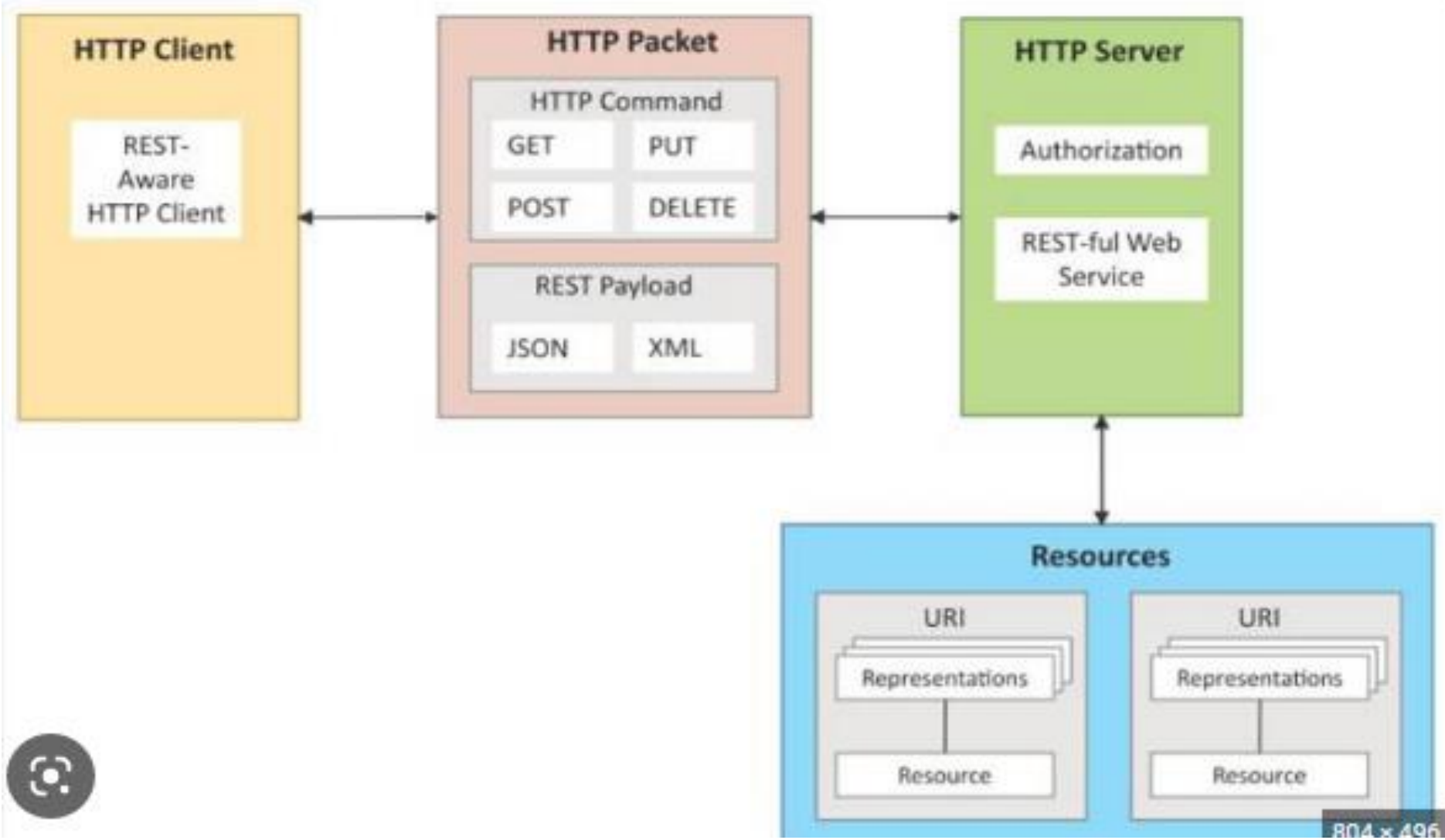
- REST based communication API
- Web Socket based communication API
- **REST Based Communication API** : Representational State Transfer (REST) is a set of architectural principles by which you can design web services and web APIs that focus on a system's resources and how resource states are addressed and transferred. REST APIs follow the request-response communication model.

REST API


{...}
REST API
Client
Server

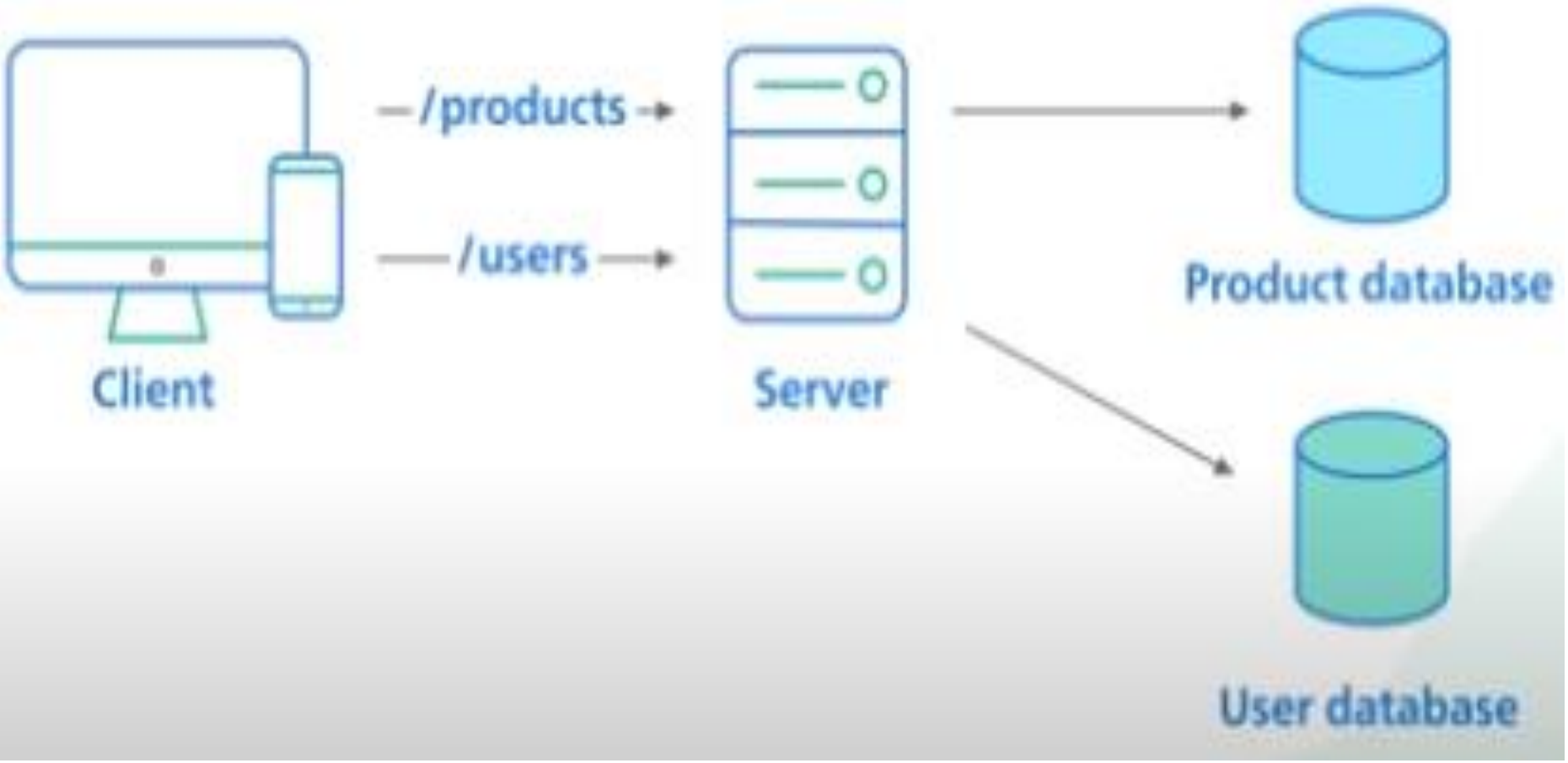

REST API Rules
Uniform Interface
Client-Server
Stateless
Cacheable
Layered System
Code on Demand (Optional)

Communication with REST



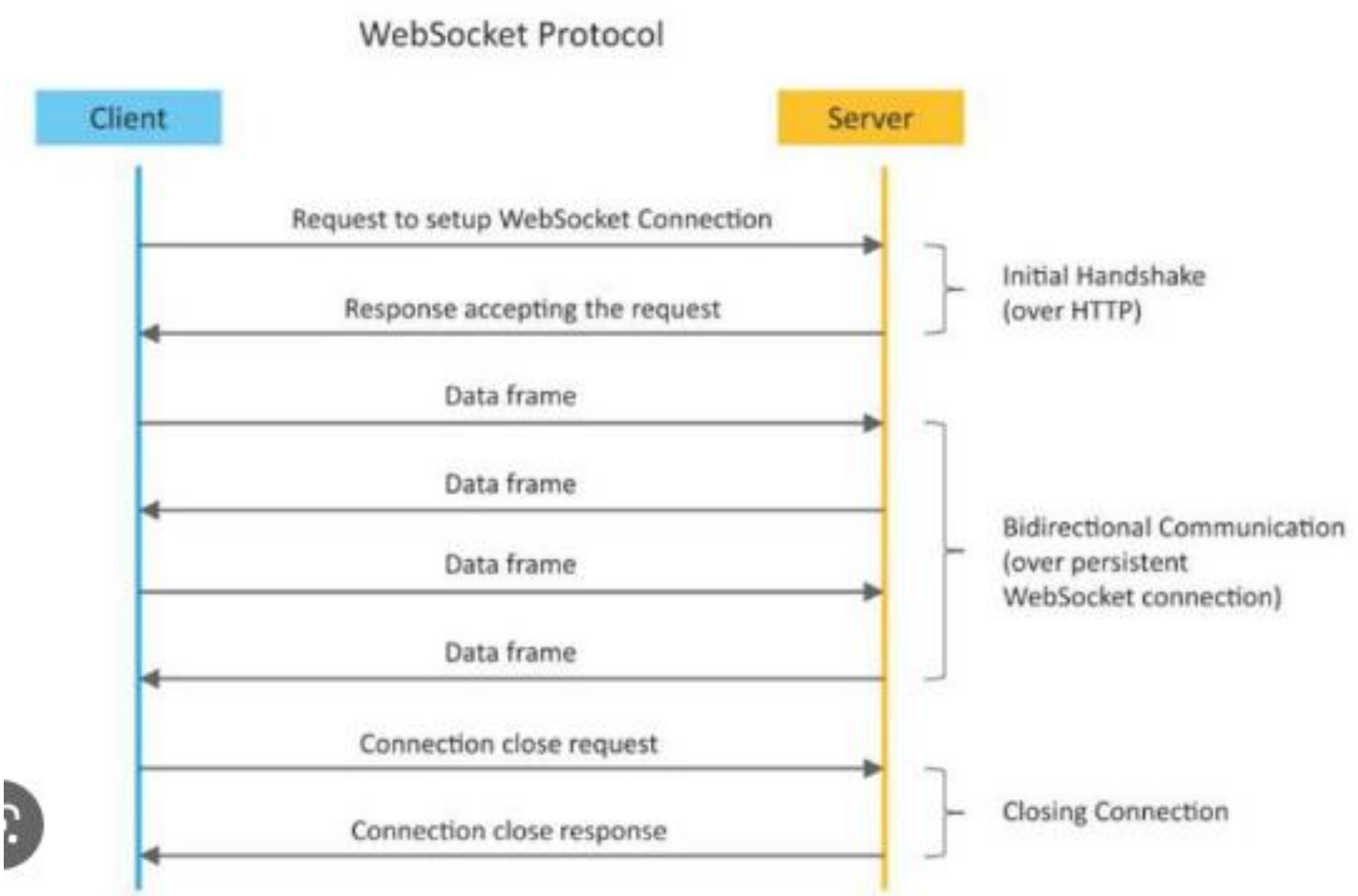
- API follow the REST standard is called RESTFUL API
- Example :Google Maps
- Basics of REST
- REST API –organize the resource using unique URIs which are worked as a resource identifiers

https://example.com/api/v3/products
https://example.com/api/v3/users



WEB Socket based communication APIS

- Web Socket APIs allow bi-directional, full-duplex communication between clients and servers. It follows the exclusive pair communication model. This Communication API does not require a new connection to be set up for each message to be sent between clients and servers. Once the connection is set up the messages can be sent and received continuously without any interruption. WebSocket APIs are suitable for IoT Applications with low latency or high throughput requirements.


WebSocket Protocol
Client
Server
Request to setup WebSocket Connection
Response accepting the request
Initial Handshake
(over HTTP)
Data frame
Data frame
Data frame
Data frame
Bidirectional Communication
(over persistent
WebSocket connection)
Connection close request
Connection close response
Closing Connection