Solution IAT-1

IoT &WSN [18EC752]

USN □□□□□□□□□□

CELEBRATING 25 YEARS
CMRIT
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test 1 – Nov. 2021

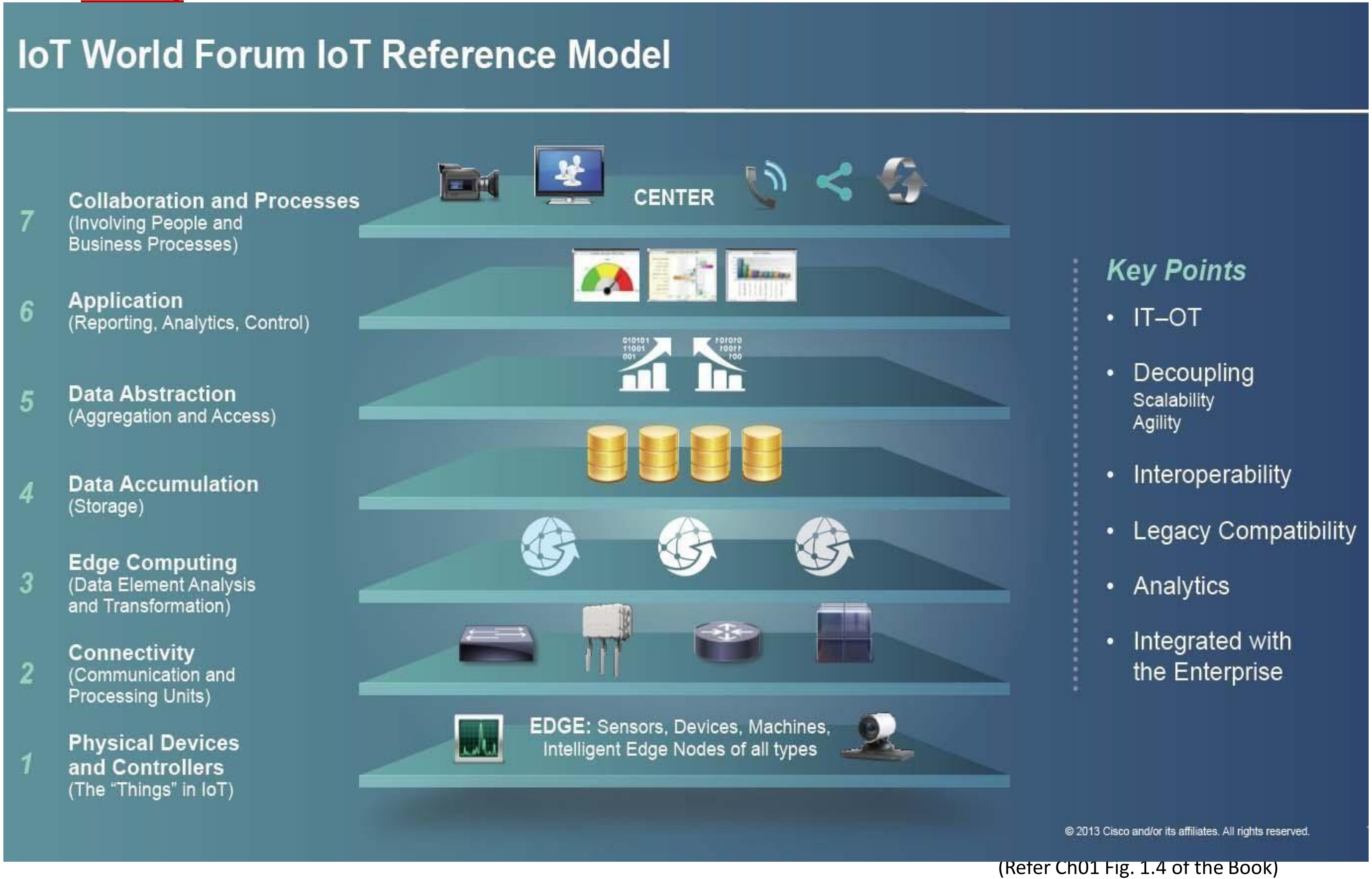
Sub:	IoT &WSN					Sub Code:	18EC752	Branch:	ECE
Date:	/11/2021	Duration:	90 min's	Max Marks:	50	Sem / Sec:	7 A,B,C,D		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	What is IoT? Explain conceptual frame work of IoT with necessary equations and reference model suggested by CISCO.	[3+7]	CO1	L1
2	What are the three architectural domain functionalities in M2M architecture? What are the differences between IoT and M2M?	[5+5]	CO2	L2
3	Explain Constrained Application Protocol (CoAP) and XMPP protocol for IoT/M2M.	[6+4]	CO2	L2
4	Explain modified OSI Model for the IoT/M2M Systems with appropriate figures.	[10]	CO1	L1
5	Explain message communication protocol MQTT with pub/sub model. Draw all the necessary figures.	[10]	CO2	L2
6	Write and explain four layer architectural frameworks developed at CISCO for a city.	[10]	CO2	L2
7	Explain about cloud service and deployment models giving examples.	[5+5]	CO2	L1
8	What is cloud computing? Explain IoT cloud based data collection, storage and computing services using Nimbits.	[3+7]	CO2	L2

1 What is IoT? Explain conceptual frame work of IoT with necessary equations and reference model suggested by CISCO.

(Refer Ch01 Fig. 1.4 of the Book)

IOT reference model suggested by CISCO (Architectural view)



IoT World Forum IoT Reference Model

7 **Collaboration and Processes** (Involving People and Business Processes)

6 **Application** (Reporting, Analytics, Control)

5 **Data Abstraction** (Aggregation and Access)

4 **Data Accumulation** (Storage)

3 **Edge Computing** (Data Element Analysis and Transformation)

2 **Connectivity** (Communication and Processing Units)

1 **Physical Devices and Controllers** (The "Things" in IoT)

CENTER
010101
11001
001
EDGE: Sensors, Devices, Machines, Intelligent Edge Nodes of all types

Key Points

- IT–OT
- Decoupling
  Scalability
  Agility
- Interoperability
- Legacy Compatibility
- Analytics
- Integrated with the Enterprise

© 2013 Cisco and/or its affiliates. All rights reserved.

(Refer Ch01 Fig. 1.4 of the Book)

Gather + Enrich + Stream + Manage + Acquire + Organize and Analyze

IEEE suggested P2413 standard for Architecture of IoT

- A reference architecture of IoT
- The IOT reference model has 7 levels called “LAYERS” OR “TIERS”.
- Each level is defined with some terminology.
- Each level perform some specific function.
- The model describes how the task at each layer should be handled to maintain simplicity and scalability.
- IN IOT the data flows in both directions i.e.

  From top to bottom (LAYER 7 to LAYER 1) – control pattern.

  From bottom to top (LAYER 1 to LAYER 7) – monitoring pattern

But Basically follows top-down approach (means consider top layer design first and then move to the lowest).

Architecture of IoT

- It defines basic architectural building blocks and their integration capability into multi-tiered systems.
- The reference model defining relation-ships among various IoT verticals, for example, transportation and healthcare
- Gives a blueprint for data abstraction
- Recommends quality ‘quadruple’ trust
- “Protection, Security, Privacy, and Safety”
- Defines no new architecture and no reinvent but existing architectures congruent with it

LAYER 1. Physical devices and controllers.

1
Physical Devices
and Controllers
(The "Things" in IoT)
EDGE: Sensors, Devices, Machines,
Intelligent Edge Nodes of all types

- They are the **physical devices** , also called as "**THINGS**" in IOT .
- Basically they are **Embedded Devices**, Embedded hardware/software like Sensors/Actuators , RFID, Hardware (Arduino, Raspberry Pi, Intel Edison, Beagle Bone Black and Wireless SoC…).
- They are **ready to send and receive the information**.
- Devices are **unlimited, diverse and no rules about the size,** location etc…. For example..
- Devices are capable of **Analog to digital conversion** and vice versa.
- Devices are capable of **generating data and being queried**.

LAYER 2. Connectivity (Communication and processing units)

Processing Units:

□Contains **Routers and Gateways**

□Main task is to deliver the right information at right time and to right machine i.e. reliable transmission.

Commu-
nication
Manage-
ment
Protocol
Handlers
Message
Router
Message
Cache

Communication:

□Includes **protocol handlers, message routers, message cache**

□It can be between smart device and network/ internet directly

□It can be through gateways then to network

□**Therefore main task involves switching and routing, enriching, transcoding, translation between protocols, security and self learning etc.**

□**Communication occurs networks – EAST-WEST communication**

Popular Communication Protocols Used from sensors to gateways are:

- ZigBee, 6LOWPAN , Bluetooth, RFID
- WiFi, WiMax, 2G/3G/4G/5G

Popular IoT Development Boards (which acts as gateways) used for connectivity are

- **Arduino Boards(e**.g. Arduino Yún), **Intel Galileo board, Beagle Board,   Raspberry Pi.**
- **They are also called as Sources of IOT.**

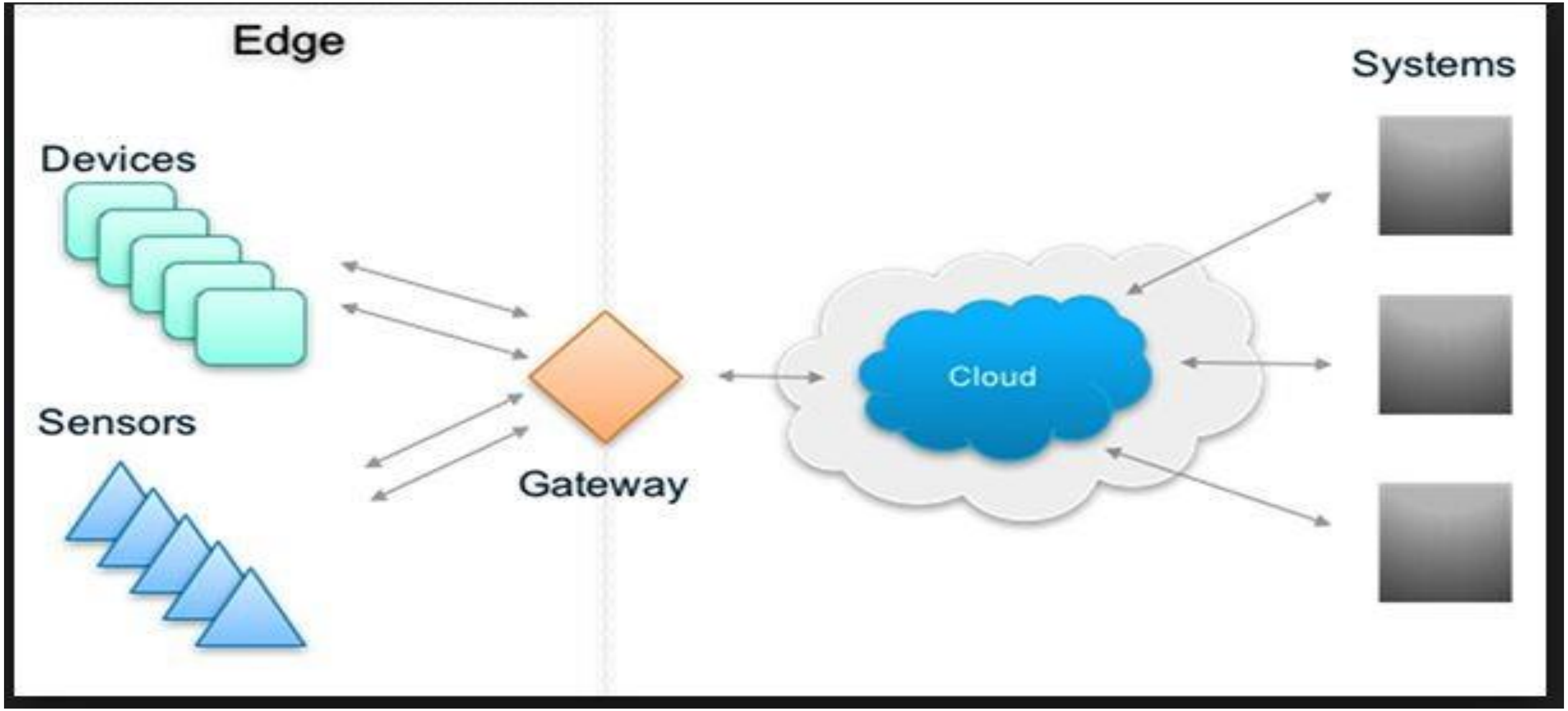
Layer 3 [Edge Computing Or Fog Computing]

□Edge computing is an architecture that uses **edge devices / network edge** like **routers, gateways, switches**, multiplexers, integrated access devices to do some **preprocessing of data**.

□**Preprocessing** includes data aggregation, storage, data filtering, cleanup, analysis, transformation (formatting, decoding, distillation) , Threshold(alert), event generation etc.

□Finally the data is routed to web servers/cloud.

Then the data from gateways are send to higher layers i.e. to backend server/cloud or data base centres using some protocols like CoAP, RESTful HTTP, MQTT, XMPP (Extensible Messaging and Presence Protocol**)**



Layer 4 [Data Accumulation and storage]

Data management is done at backend server/cloud or data base centres

Main roles of layer 4 are:

- □**Convert data in motion to data at rest.**
- □**Convert format from network packets to database relational tables.**
- □Convert Event based data to query based data (it bridges the gap between real time networking and non real time)
- □The concept of BIG DATA is used at layer 4.

<u>Layer 5 Data Abstraction</u>

Data abstraction is done at backend server/cloud or data base centres

Abstraction **means providing the essential and relevant information** of   the data by hiding the irrelevant one.

Main roles are:

1) **<u>Provide multiple storage systems</u>** to accommodate data from different IOT devices.
2) **<u>Reconciling multiple data format</u>** from different sources.
3) **<u>Combining data from multiple sources</u>** and simplifying the application i.e <u>consolidating the data</u> into one place.
4) **<u>Filtering, selecting, projecting and reformatting the data</u>** to serve client application.
5) **<u>Protecting the data</u>** with appropriate authentication and authorization.

Layer 6 Application

Application
(Reporting, Analytics, Control)

- Layer 6 deals with reporting, analysis and control
- i.e. the data is analysed and then send to controlling device like actuator.
- And then the data is passed to specific application like mobile application or webpage or to the business enterprise which require that data.

Applications

Some of the main applications are

□ Mobile, tablets, IP-TV, VOIP telephony, video-conferencing, video-on-demand, videos surveillance,

□ Wi-Fi and Internet,

□ Home security: access control and security alerts

□ Wi-Fi and Internet

□ Lighting control

□ Home health care

□ Fire detection: Leak detection

□ Energy efficiency Solar panel monitoring and control, Temperature monitoring and HVAC control

Layer 7 Collaboration and processes.

□ It means involving people and business process.

□ Basically multiple people are using same applications for a range of different purpose

□ But in IOT the main objective is to empower people to do their work better, not the application.

2 What are the three architectural domain functionalities in M2M architecture? What are the differences between IoT and M2M?

M2M communication

□M2M refers to the process of **communication of the physical devices or machines or smart devices with the other machines of same type without intervention of humans**.

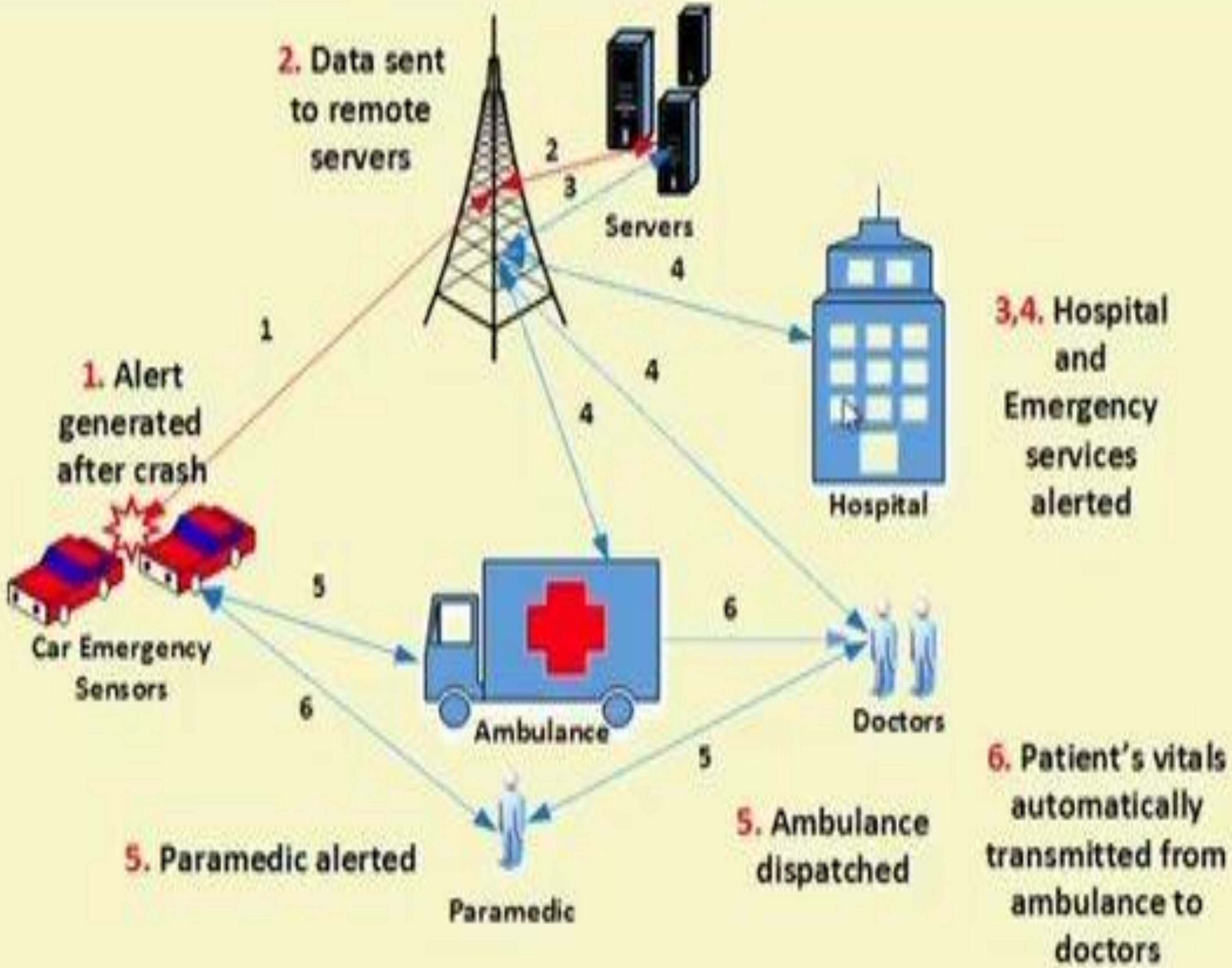
□**Smart devices collect the data, monitor it, do some necessary computations, and perform communication to the remote devices using internet**

□It also uses servers or cloud end applications, services and processes.

□**M2M is used in many applications like home automation, industrial automation, smart cities, healthcare etc** .

□For-example using robots interacting with machines at home.

□IT is similar to SCADA (Supervisory control and data acquisition system)



1. Alert generated after crash

2. Data sent to remote servers

3,4. Hospital and Emergency services alerted

5. Paramedic alerted

5. Ambulance dispatched

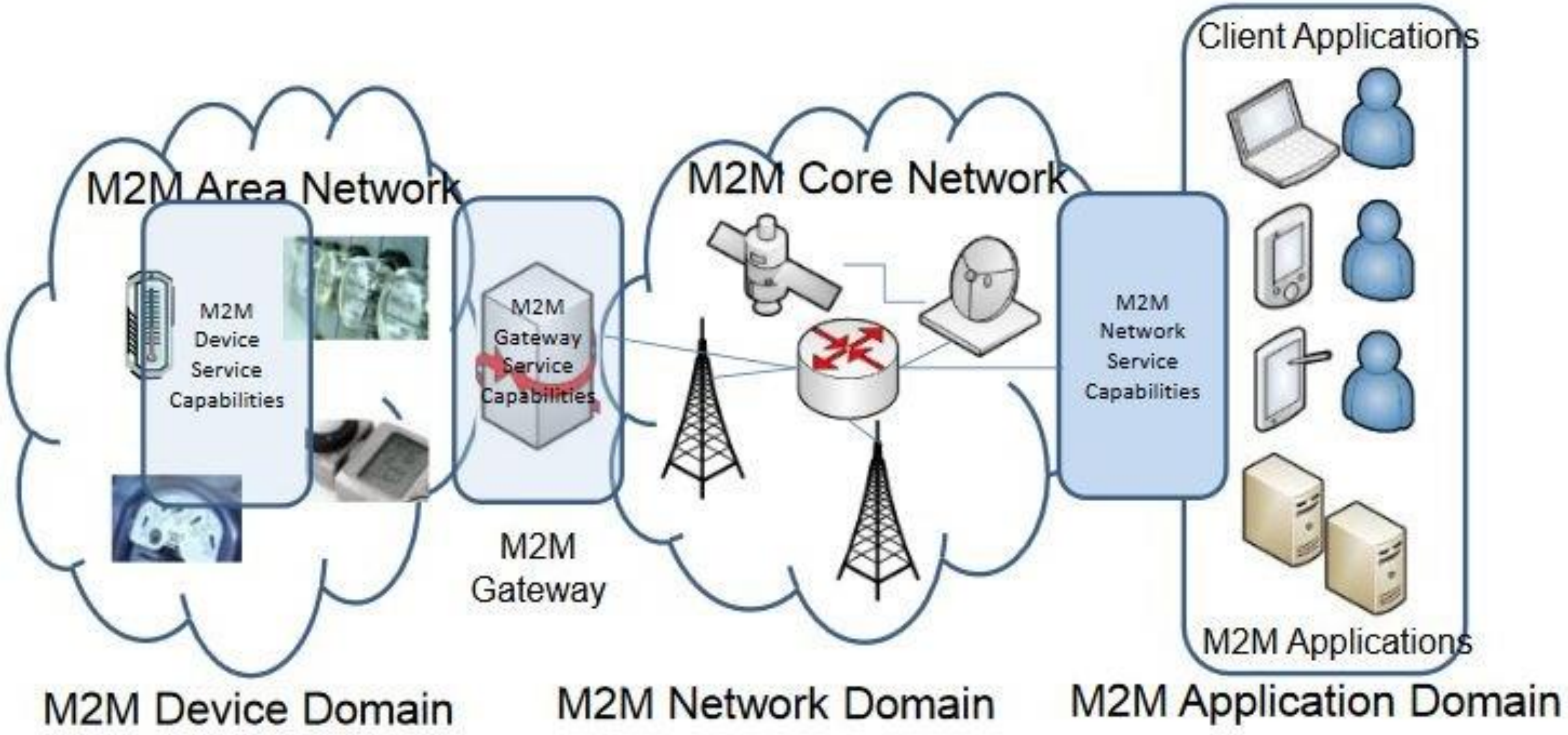
6. Patient's vitals automatically transmitted from ambulance to doctors

M2M Architecture:

Divided into three domains

1) M2M device domain
2) M2M network domain
3) M2M application domain

ETSI M2M Network



M2M device domain

It consists of three subparts

a) **Physical devices and controllers**
b) **Communication interface**
c) **Gateway (BS)**

a) **Physical devices and controllers:**

□ They are **sensors, physical devices, controllers, machines which are capable of transmitting data autonomously.**

□ **Two types of devices –**ones capable of directly connecting to the network and the others requires an M2M gateway in order to connect to the network

□ Generally, devices can connect directly to an M2M devices via embedded SIM, TPM and radio stack or fixed line access but some connect through gateway.

a) **Communication interface:**

It is the port or processing unit that receives data from one interface and transmit it to other interface.

c) **Gateway**

Gateways and routers are the endpoints of the operator’s network in scenarios where sensors and M2M devices do not connect directly to the network

2) **M2M Network Domain (Communication Networks)**

It consists of ***M2M core and M2M service*** *capabilities.*

□***M2M core*** covers the communications between the M2M Gateway(s) and M2M application(s), e.g. LTE, WiMAX, and WLAN.

□**M2M service** capabilities include network functions to support M2M applications. It also deals with management functions like device identity management, data storage, data collection, analysis, aggregation etc.

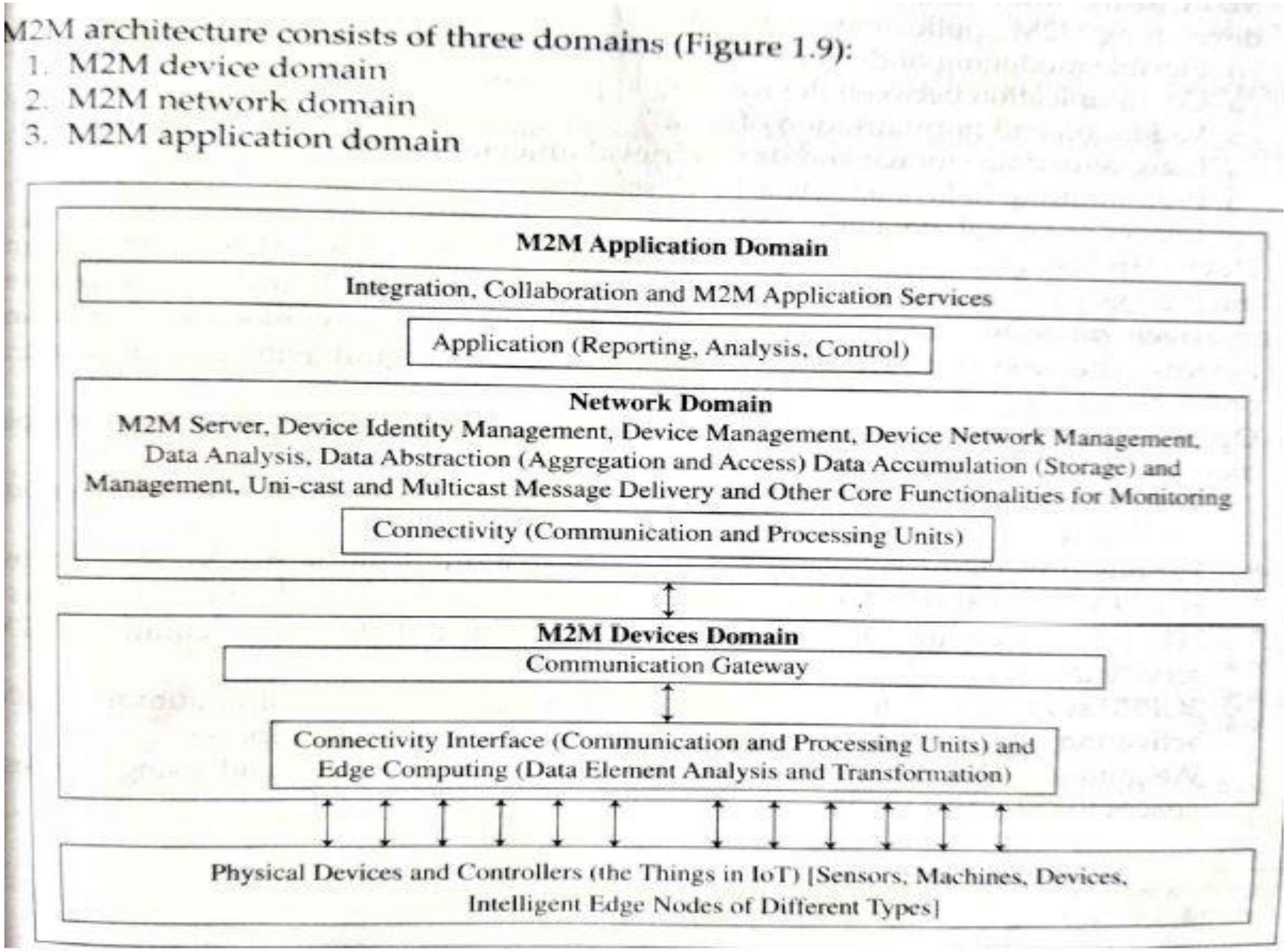
3) M2M application domain

Two types of applications -**M2M applications and client applications**

M2M applications: **These applications are located on the servers, interacts with M2M devices.**

Client applications: **These used to serve end-users; either receive services from M2M applications or directly from M2M devices**.

Here the received data is further analyzed, reports are generate and based on the analysis further actions are taken for controlling the network.



M2M architecture consists of three domains (Figure 1.9):

1. M2M device domain
2. M2M network domain
3. M2M application domain

M2M Application Domain

Integration, Collaboration and M2M Application Services

Application (Reporting, Analysis, Control)

Network Domain

M2M Server, Device Identity Management, Device Management, Device Network Management, Data Analysis, Data Abstraction (Aggregation and Access) Data Accumulation (Storage) and Management, Uni-cast and Multicast Message Delivery and Other Core Functionalities for Monitoring

Connectivity (Communication and Processing Units)

M2M Devices Domain

Communication Gateway

Connectivity Interface (Communication and Processing Units) and Edge Computing (Data Element Analysis and Transformation)

Physical Devices and Controllers (the Things in IoT) [Sensors, Machines, Devices, Intelligent Edge Nodes of Different Types]

Examples of IOT

1) **Wearable smart watch**

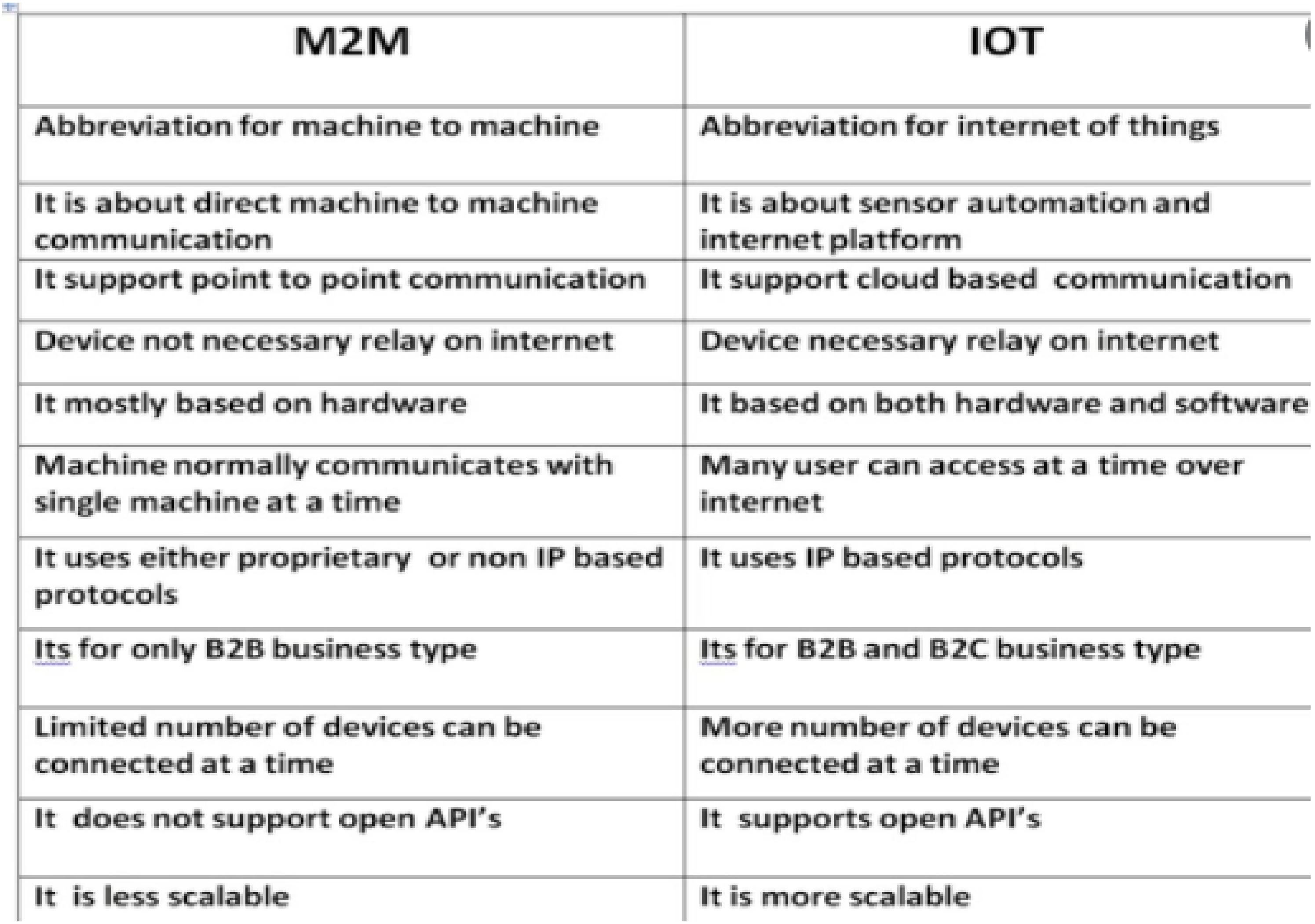
- Samsung Galaxy Gear S
- Apple watch
- Microsoft Wrist Band 2

2) Smart Home

- Intel based intelligent gateway enables creation of smart home offered by service provider for telephony, mobile, cable, broadband
- OpenHAB (Open home automation bus) enables smart home devices to communicate and controlled via home.

3) Smart cities

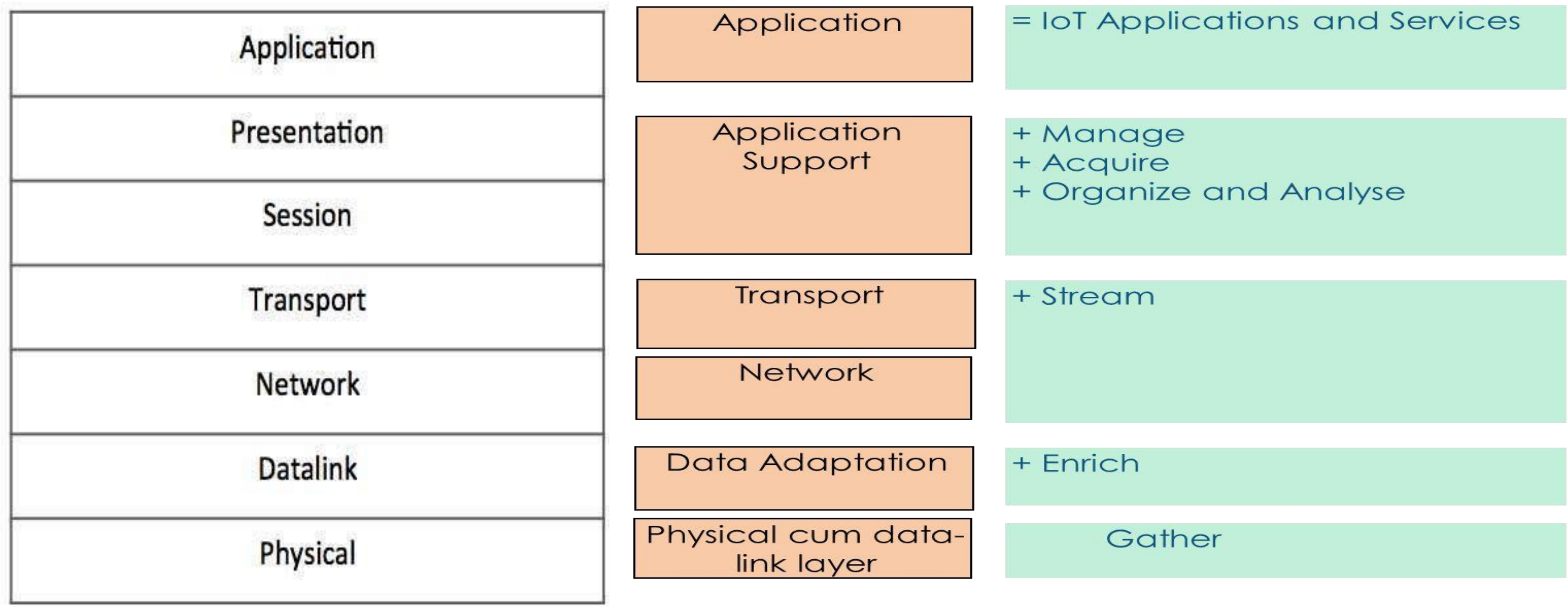
Difference between IoT and M2M



M2M	IOT
Abbreviation for machine to machine	Abbreviation for internet of things
It is about direct machine to machine communication	It is about sensor automation and internet platform
It support point to point communication	It support cloud based communication
Device not necessary relay on internet	Device necessary relay on internet
It mostly based on hardware	It based on both hardware and software
Machine normally communicates with single machine at a time	Many user can access at a time over internet
It uses either proprietary or non IP based protocols	It uses IP based protocols
Its for only B2B business type	Its for B2B and B2C business type
Limited number of devices can be connected at a time	More number of devices can be connected at a time
It does not support open API's	It supports open API's
It is less scalable	It is more scalable

4	Explain modified OSI Model for the IoT/M2M Systems with appropriate figures.

Modified OSI model for the IOT/M2M Systems



Application

Presentation

Session

Transport

Network

Datalink

Physical

Application

Application Support

Transport

Network

Data Adaptation

Physical cum data-link layer

= IoT Applications and Services

+ Manage
+ Acquire
+ Organize and Analyse

+ Stream

+ Enrich

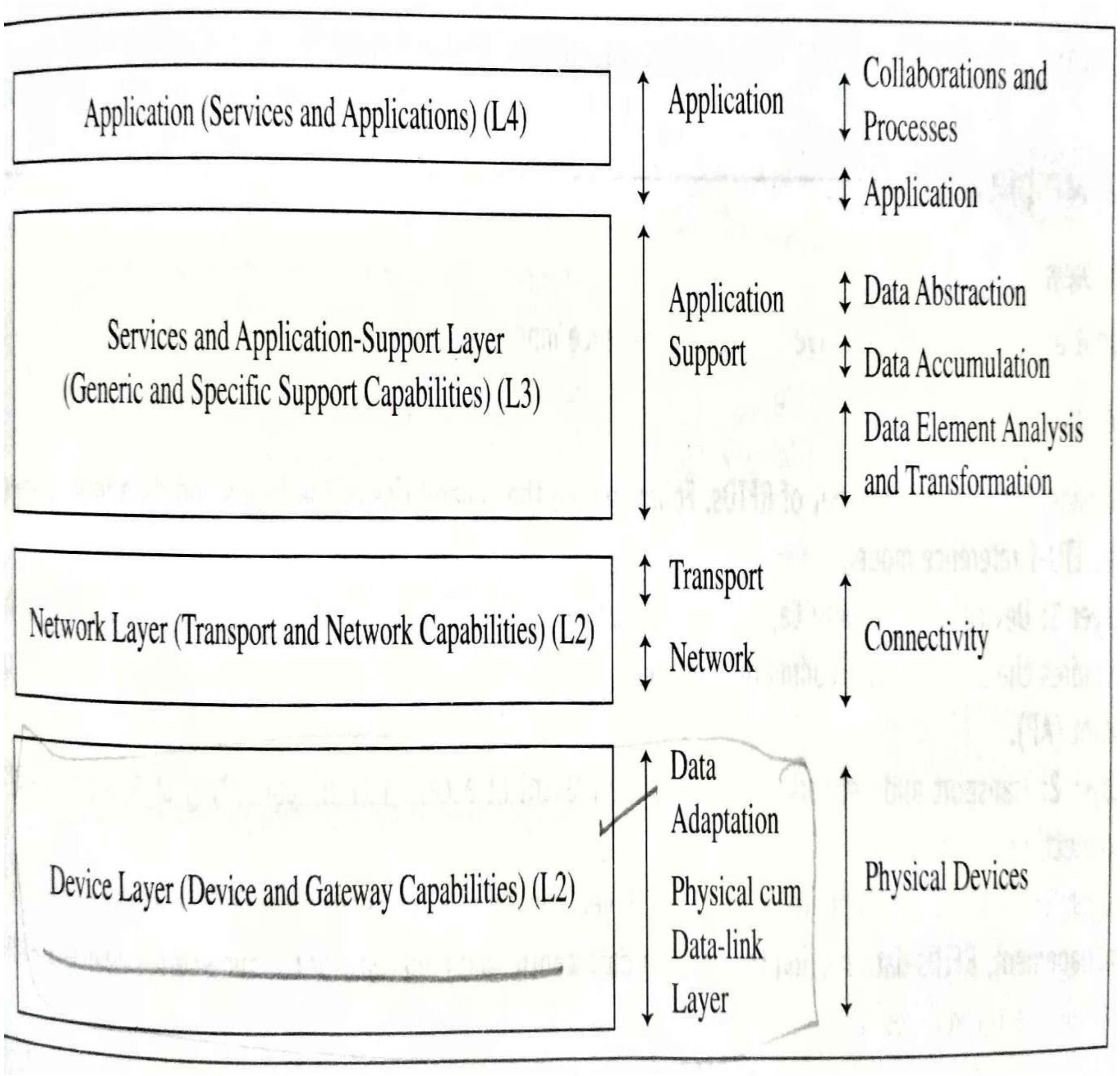
Gather

7-Layer Generalised OSI Model

IETF 6-Layer modified OSI Model

IETF: Internet engineering task force

ITU-T reference model


Application (Services and Applications) (L4)
Services and Application-Support Layer
(Generic and Specific Support Capabilities) (L3)
Network Layer (Transport and Network Capabilities) (L2)
Device Layer (Device and Gateway Capabilities) (L2)
Application
Application Support
Transport
Network
Data Adaptation
Physical cum Data-link Layer
Collaborations and Processes
Application
Data Abstraction
Data Accumulation
Data Element Analysis and Transformation
Connectivity
Physical Devices

Physical cum data link layer

❑ It is also called perception layer or object layer or device layer.

❑ The physical layer consists of the physical device called as ‘Things’ which can be sensors, actuators, RFID tags.

❑ These devices collects and **gather** the raw data from the environment, do some processing and have computational capabilities.

❑ The various protocols which works and allows the smallest devices to transmit information seamlessly over internet are:
802.15.4 –6LoWPAN/IPV6, Zigbee, 802.11 (wi-fi),802.11ah, wirelessHART, DASH-7, LoRaWAN, RFID,
EC-GSM-IOT, X-10, RS-232, Z-wave, 3G/LTE, 3GPP

802.15.4 –6LoWPAN/IPV6: (Low-Power Wireless Personal Area Networks)

- This protocol allows the low power smallest devices like sensors to communicate across internet using internet

protocol.

Zigbee: IEEE802.11.4 standard

- It is low cost, low power, wireless mesh network standard used for low power battery operated devices.

802.11 (wi-fi):

- It is most widely used computer networking standard used in home, offices etc.

802.11ah: Wi-Fi hai-low:

- Operate in 1 GHz licence band
- Used for sensor network.

wirelessHART: Highway Addressable Remote Transducer Protocol (HART)

The protocol is designed for the harshest industrial environments, where low power, reliability, resilience, and scalability are key—making them well suited for general industrial applications,

DASH-7:

- IT is an open source wireless sensor and actuator network protocol operates at 915 MHz band.
- Provide multilayer battery life with high data rate.

LoRaWAN: The *LoRaWAN* (Low Power WAN *Protocol* for Internet of Things)

- It is designed to allow low powered devices to communicate with the internet connected applications over a long range.
- It supports bidirectional communication between device and gateway.

EC-GSM-IOT: Extended coverage Global system for mobile communication

- This standard is based Low Power Wide Area technology.
- It is based on eGPRS and designed as a high capacity, long range, low energy and low complexity cellular system for IoT communications

X-10:

X10 is a protocol for communication among electronic devices used for home automation

Z-wave:

Z-Wave is a wireless communications protocol used primarily for home automation

3G/LTE

In telecommunication, **Long-Term Evolution** (**LTE**) is a standard for high-speed wireless communication for mobile devices and data terminals, based on the GSM/EDGE and UMTS/HSPA technologies

EDGE: Enhanced Data rates for GSM Evolution
UMTS: Universal Mobile Telecommunications System
HSPA: High Speed Downlink Packet Access

Data Adaptation layer

□ The main function of data adaptation layer is **data enrichment**.

□ Enrichment is a value adding process where external data from multiple sources is added to the existing data to enhance the quality and richness of data.

□ Gateway is considered to work at data adaptation layer.

<u>Gateway and network layer and transport layer</u>

Both the layers deals with Data streaming i.e. transfer of data at a steady high-speed rate sufficient to support such applications as high-definition television (HDTV)

Gateway and network layer

The main functions of this layer are:

1) Routing the data coming from sensors to the next upper management layer
2) This layer is responsible for managing the traffic between networks that uses the different protocols.
3) This layer is responsible for protocol translation and other interoperability tasks
4) The IOT gateway device is employed because some of the devices can’t directly communicate to the front end applications as there is no network stack present for internet connectivity, hence gateways acts as proxy.

Transport layer

The main functions of this layer are:

□ Forwarding the data coming from sensors to the next upper management layer.

□ It provides sufficient security features, bandwidth management etc.

Application support layer/middleware layer:

□This layer is also called as service management layer or processing layer.

□It is called as application support as it allows the IOT application programmer to work with heterogeneous objects without consideration to specific hardware.

□The layer deals with the device management, device modeling, device configuration, security, data abstraction, analysis of data, managing data flow, data mining, big data processing analysis of information etc.

□The layer also deals with cloud computing, web service, database management.

Application layer/business layer:

□This layer is responsible for building the business model, making reports, graphs, flowcharts,

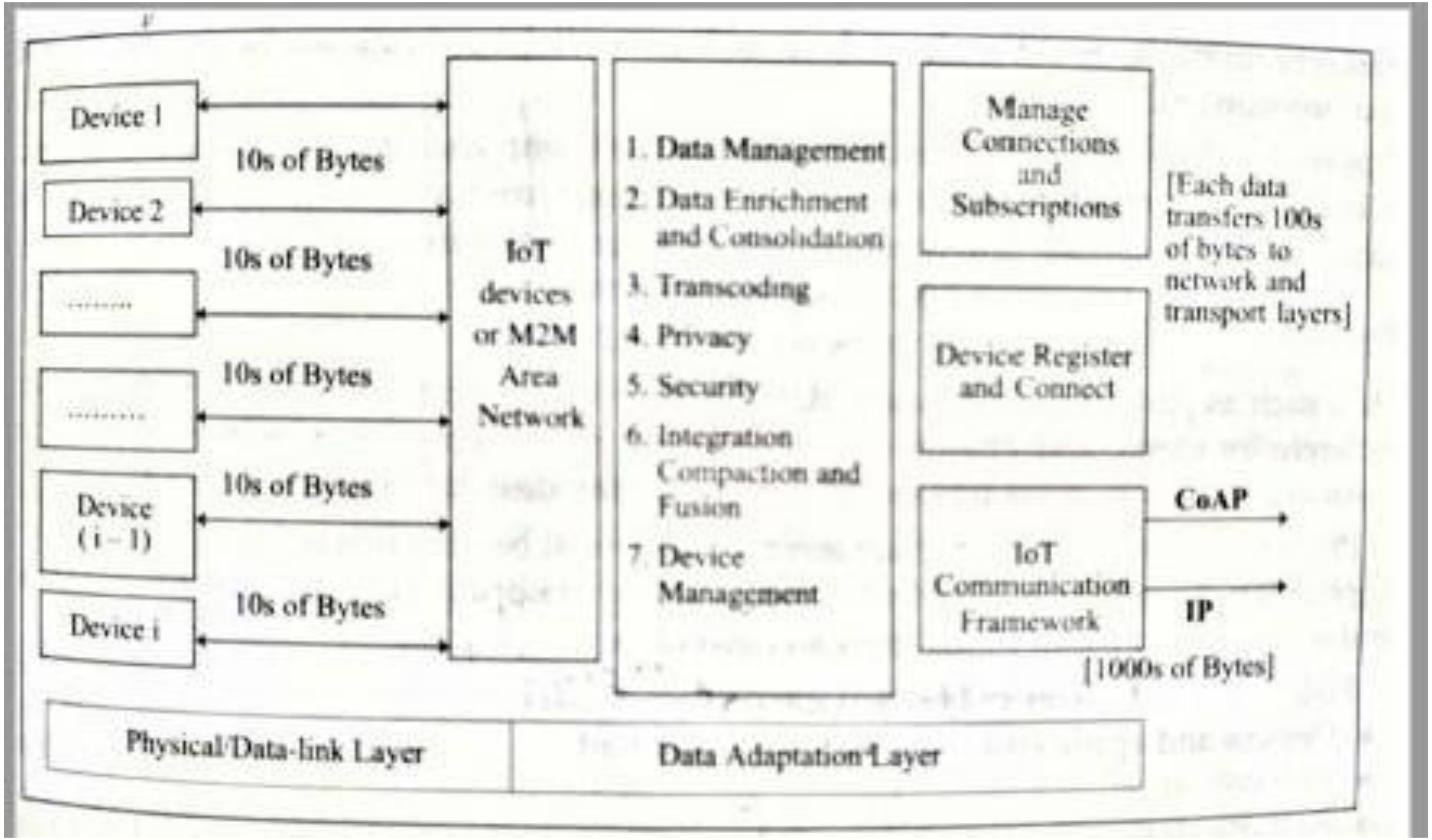
□This layer also supports decision making process based on big data analysis.

□The various protocols which work at the App layer are HTTP/COAP, MQTT, AMQP etc

Example: Data Interchange in Streetlight

Layer 1	• Smart sensing and data-link circuit with each streetlight for transferring the sensed data to the layer 2
Layer 2	• Group controller controls a group of streetlights as per the program-commands from a Central station • Data Adaptation the group-controller receives data of each group through Bluetooth or ZigBee, then aggregates and compacts the data for communication to Internet,
Layer 3	• Network stream on the Internet to next layer
Layer 4	• Transport layer for device identity management, identity registry and data routing to next layer
Layer 5	• Application support by data managing, acquiring, organising and analysing
Layer 6	• Application a remotely stored service program which issues the commands or programs the firmware at the service controllers • Service controllers switch on-off, and monitor each group of streetlights in whole of the city.

Data Enrichment, data consolidation and device management at gateway


Device 1
10s of Bytes
Device 2
10s of Bytes
....
10s of Bytes
....
10s of Bytes
Device (i – 1)
10s of Bytes
Device i
IoT devices or M2M Area Network
1. Data Management
2. Data Enrichment and Consolidation
3. Transcoding
4. Privacy
5. Security
6. Integration Compaction and Fusion
7. Device Management
Manage Connections and Subscriptions
[Each data transfers 100s of bytes to network and transport layers]
Device Register and Connect
IoT Communication Framework
CoAP
IP
[1000s of Bytes]
Physical/Data-link Layer
Data Adaptation Layer

IOT Gateway:

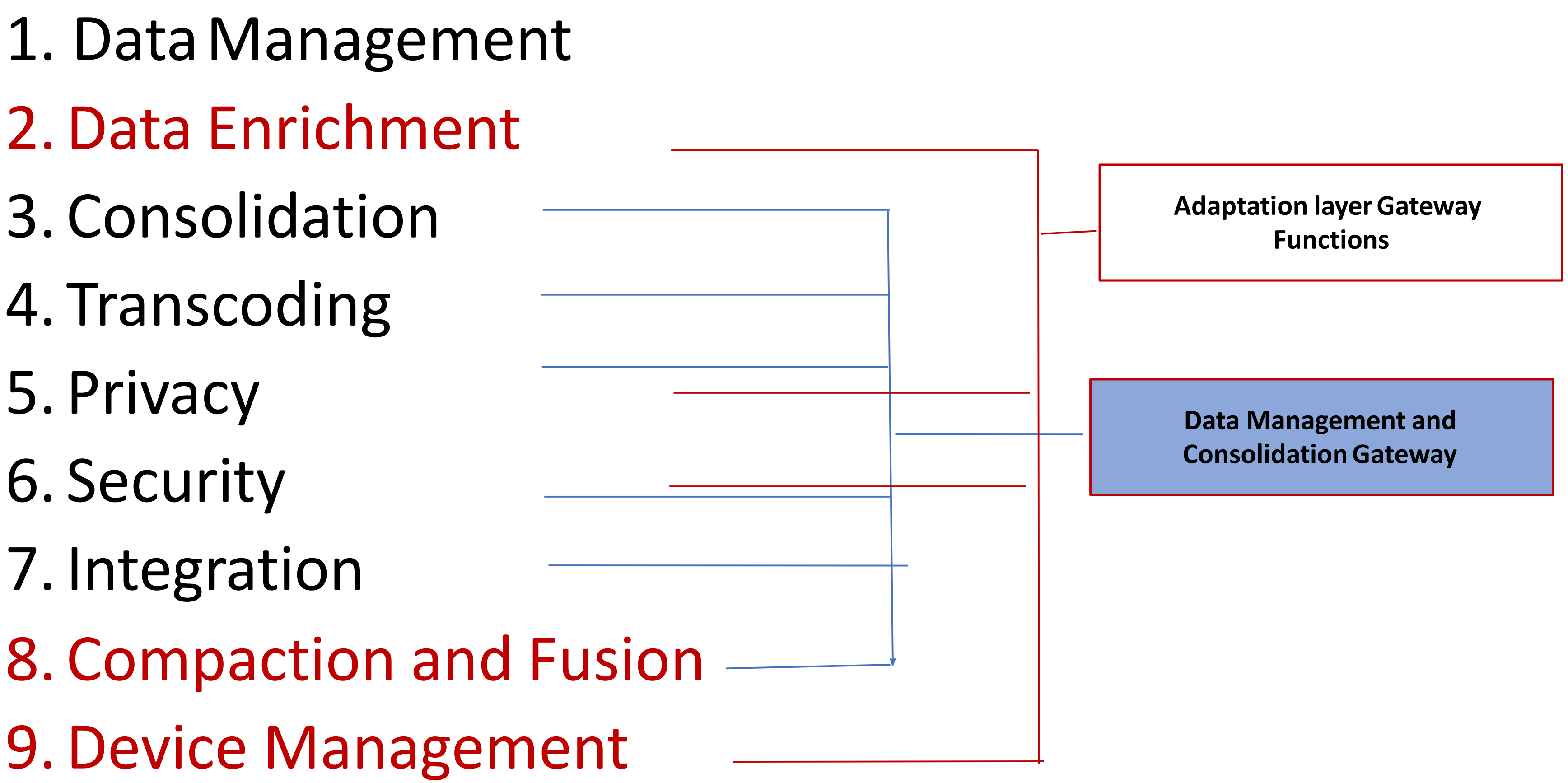
□ **The IOT gateway is considered to work at the adaptation layer of M2M architecture.**

□It can be hardware or software or combination of both.

□The main functions of gateway at adaptation layer are:

1. Data management
2. Data enrichment and consolidation
3. Transcoding
4. Privacy
5. Security
6. Integration, compaction and fusion
7. Device management

Data Enrichment, data consolidation and device management at gateway



1. Data Management
2. Data Enrichment
3. Consolidation
4. Transcoding
5. Privacy
6. Security
7. Integration
8. Compaction and Fusion
9. Device Management

Adaptation layer Gateway Functions

Data Management and Consolidation Gateway

Data Management and Consolidation Gateway functions

Transcoding:

□ It involves change of data, protocol, format or code received form the IOT device in the format which is acceptable at the server and vice versa.

□ It also involves filtering, compression, decompression etc.

Privacy:

□ It defines that the as the data must be protected from conscious or unconscious transfer to untrustworthy destination using the internet.

□ And it ensures that the stakeholder in future should not misuse the device end data or application data.

- Examples: Patient medical data, data for a company supplies from and to different locations, and changes in inventories

Privacy depends on following components:

I. Device and Applications Identities management
II. Authentication
III. Authorization
IV. Trust and
V. Reputation

Security:

□ The access to the data needs to be secure.

□ It deals with providing proper authentication of request and authorization for a response and service

Integration and Data consolidation :

□*Data consolidation* refers to the collection and integration of *data* from multiple sources into a single destination.

□ In IOT millions of devices are transmitting huge amount of data therefore attributes of devices and data belongs to different devices are correlated and integrated.

□ But there are some concerns related to Integration like location tracking, profiling threats etc.

Device Management Functions

It involves **:**

- □ Device and application identity management like device ID or address, managing device, parameters and settings.
- □ Device activation, deactivation, device configuring, device registration, device de-registration, device attaching, device detaching etc.
- □ Accepting subscription for its resources.
- □ Fault management: Course of actions and guidelines to be followed in case of a fault
- □ Forwarding function when DM server and device can interact without reformatting or structuring
- □ Protocol conversion when device and DM server using distinct protocols
- □ Use of proxy function in case the an intermediate pre-fetch required in lossy environment or network environment needs

Adaptation layer and Gateway function

1. Data Enrichment

2.Compaction and Fusion

Data Enrichment

- It is a value adding process where the external data from multiple sources is added to the existing data to enhance the quality and richness of data.

- There are three types of enrichment

1) Aggregation

2) Compaction

3) Fusion

Aggregation:

- It is the process of joining together previous and present data frames after removing redundant and duplicate frame.

Compaction:

- It means making information short without changing the context of information

Fusion:

It is a process of forwarding the formatted data after removing the redundant information from the received data.

Data Gathering

Data gathering means data- acquisition from the device(s)

Four modes of data gathering are:

1. Polling– means data sought from a device by addressing the device
2. Event based– means data gathered from device based upon some events (like smart watch)
3. Scheduled interval– means data gathered from device based upon some scheduled intervals (like in healthcare)
4. Continuous monitoring: means data gathered from device based upon continuous monitoring (like in weather monitoring)

URI: Uniform resource Identifier

- It identifies a resource either by location or name or by both.
- Resource means web resources like webpage, images, files, documents that are part of web architecture.

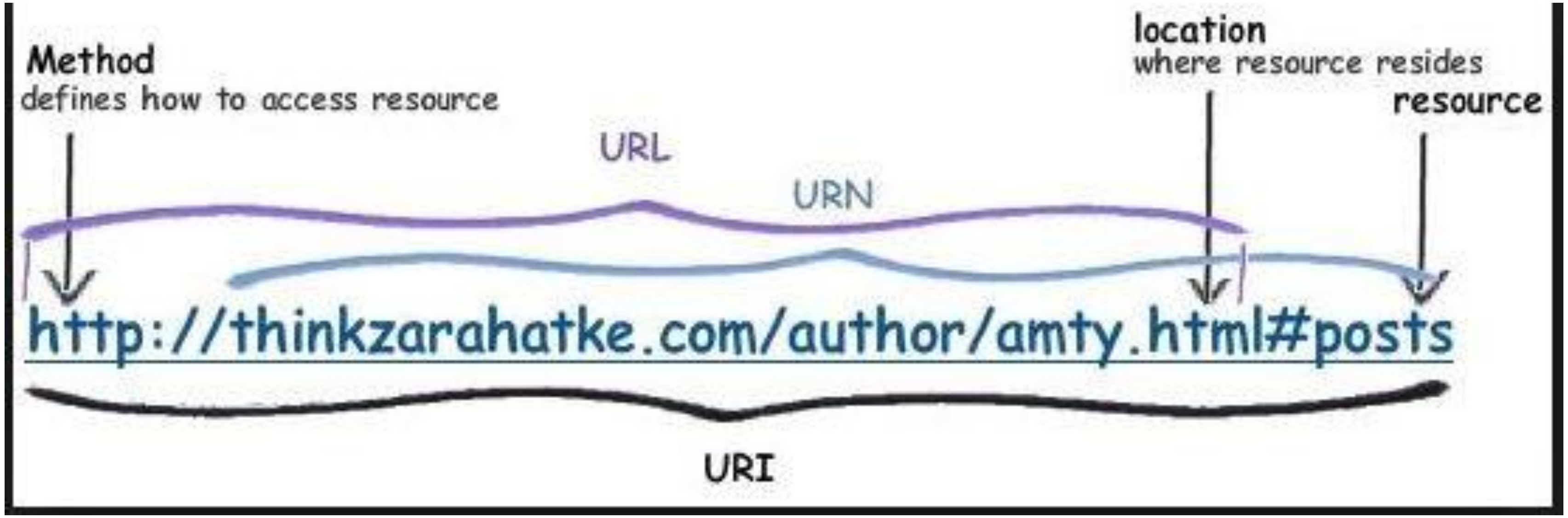
URI: URL+URN

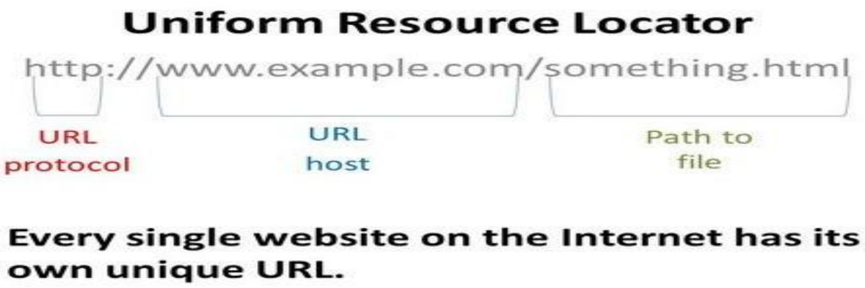
URL: Uniform resource locator

- It identifies the network location of the specific resource
- It identifies how to access the resource.

URN:

- It identifies the resource by name in a given namespace.
- But it does not specify the location of the specific representation.


Method
defines how to access resource
location
where resource resides
resource
URL
URN
http://thinkzarahatke.com/author/amty.html#posts
URI



Every single website on the Internet has its own unique URL.

urn:isbn:0451450523

to identify a book by its ISBN number.

ISBN: The International Standard Book Number

API: Application programming interface

□ APIs is a code or set of routines, protocols and tools that allows different kind of applications to talk to each other.

□ **e.g. google login, booking the airline tickets.**

□ **APIs can be free and can be paid**

Some different kind of APIs in IOT services are:

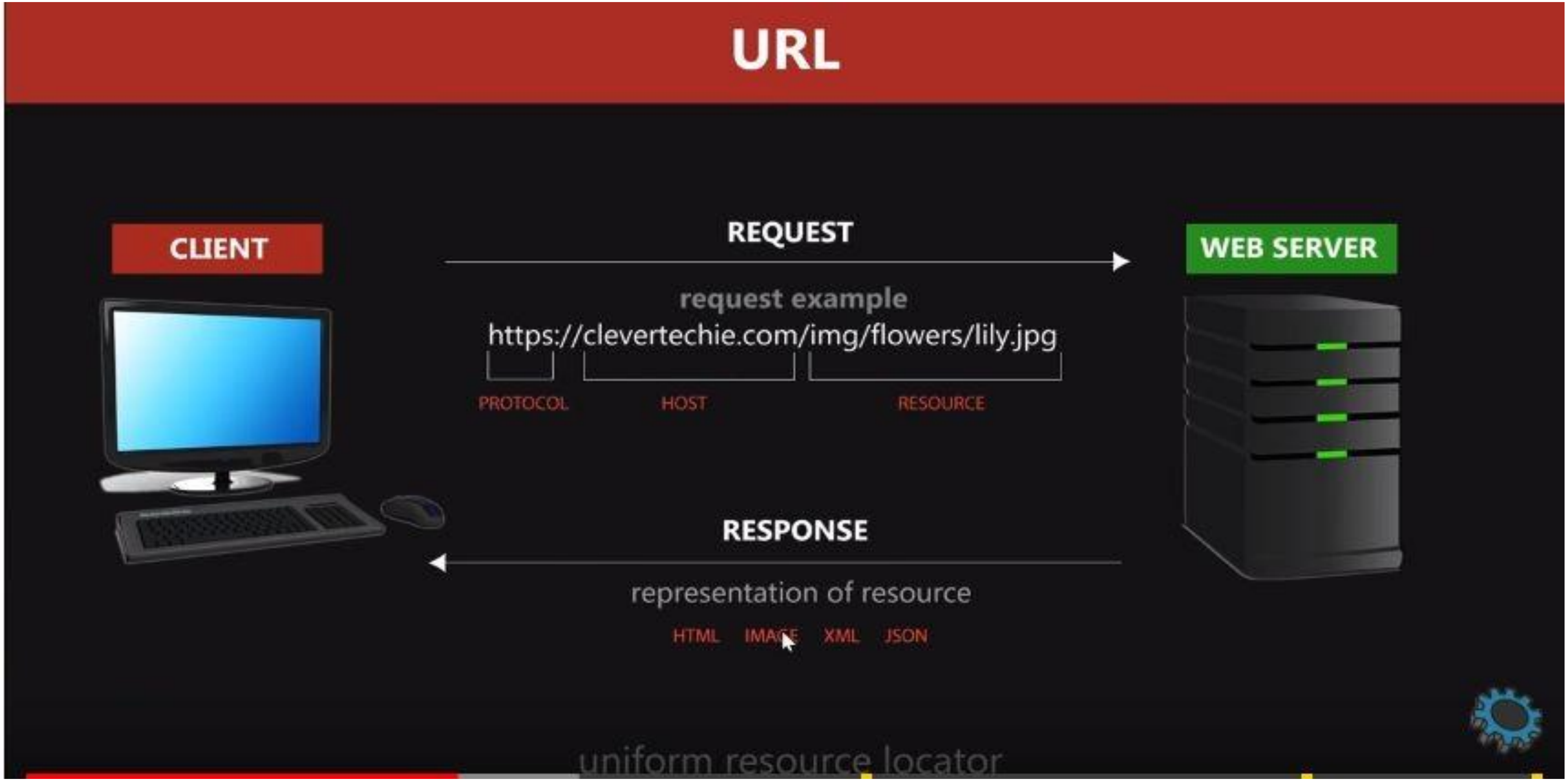
REST

XML/JSON

SOAP

REST : Representational State Transfer

- □There is the transfer of representation of the resources.
- □**Resources** can be anything like image, photos, list of photos, comments, articles, books, profiles etc.
- □When you type some URL in web browser or click some link in webpage then you send a request to the webserver using HTTP to deliver the resource and webserver returns the data in the format which is readable or required by the client called **representation of resource.**
- □ *The webserver actually delivers the representation of the resource in the format that is readable by the user e.g. in HTML, JSON, txt, XML form.*



URL

CLIENT

REQUEST

WEB SERVER

request example

https://clevertechie.com/img/flowers/lily.jpg

PROTOCOL HOST RESOURCE

RESPONSE

representation of resource

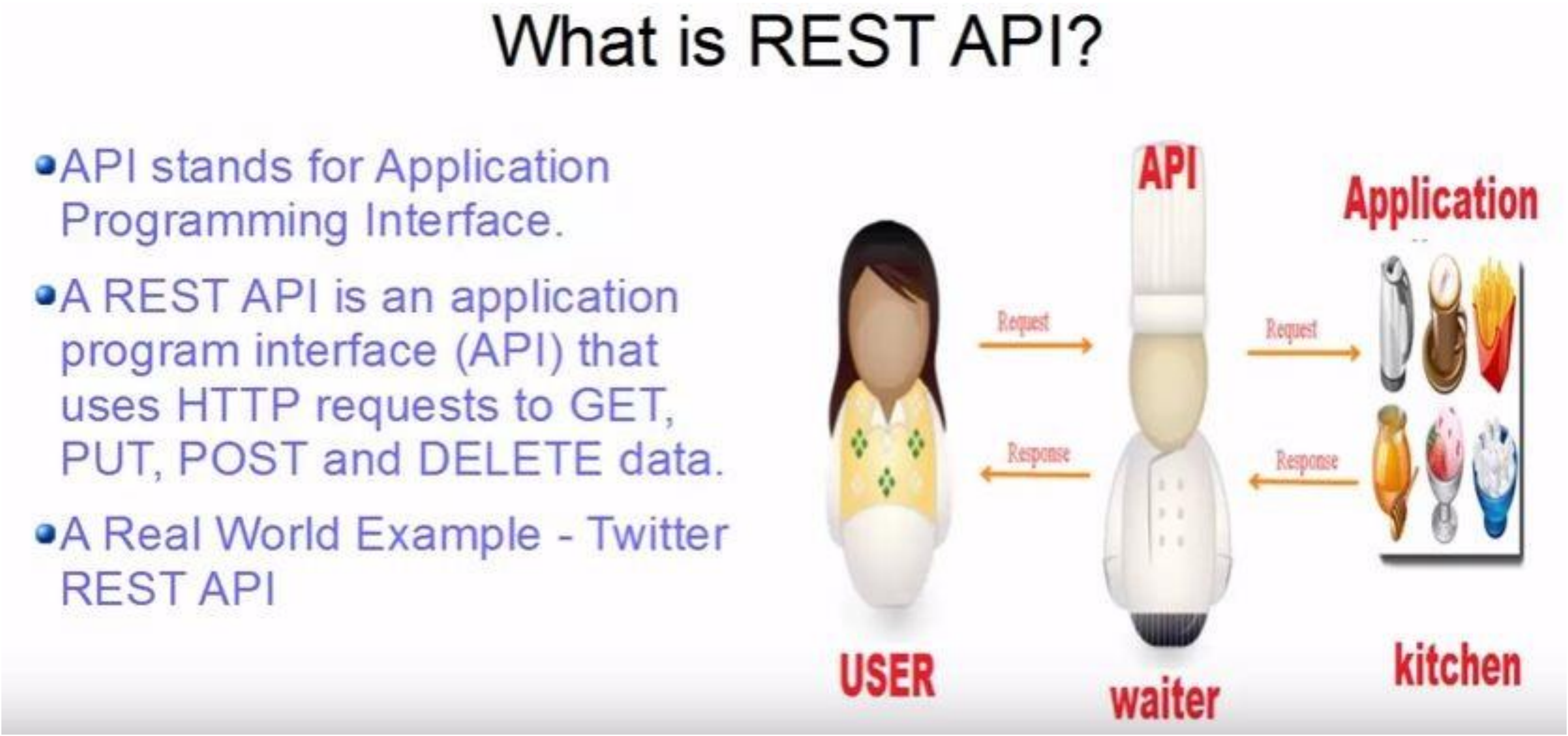
HTML IMAGE XML JSON

uniform resource locator

REST : Representational State Transfer

- □ It is an architectural style
- □ The resources are accessed by using URI (by using HTTP).
- □ URI is called the address of resource.
- □ The current state of the resource from the server (also called as constant **resource state**) is captured and transmitted to client by using a representation in an understandable format (i.e. **application state** of the client) called as representation state transfer.

- □ So, this resource state on server remains constant therefore called as stateless, and only the representation of the resource will change, which in turn would change the application state called as REST



What is REST API?

- API stands for Application Programming Interface.
- A REST API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data.
- A Real World Example - Twitter REST API

API
Application
Request
Request
Response
Response
USER
waiter
kitchen

```
GET "http://localhost:8080/confluence/rest/api/content?type=blogpost&start=0
    &limit=10&expand=space,history,body.view,metadata.labels"
```

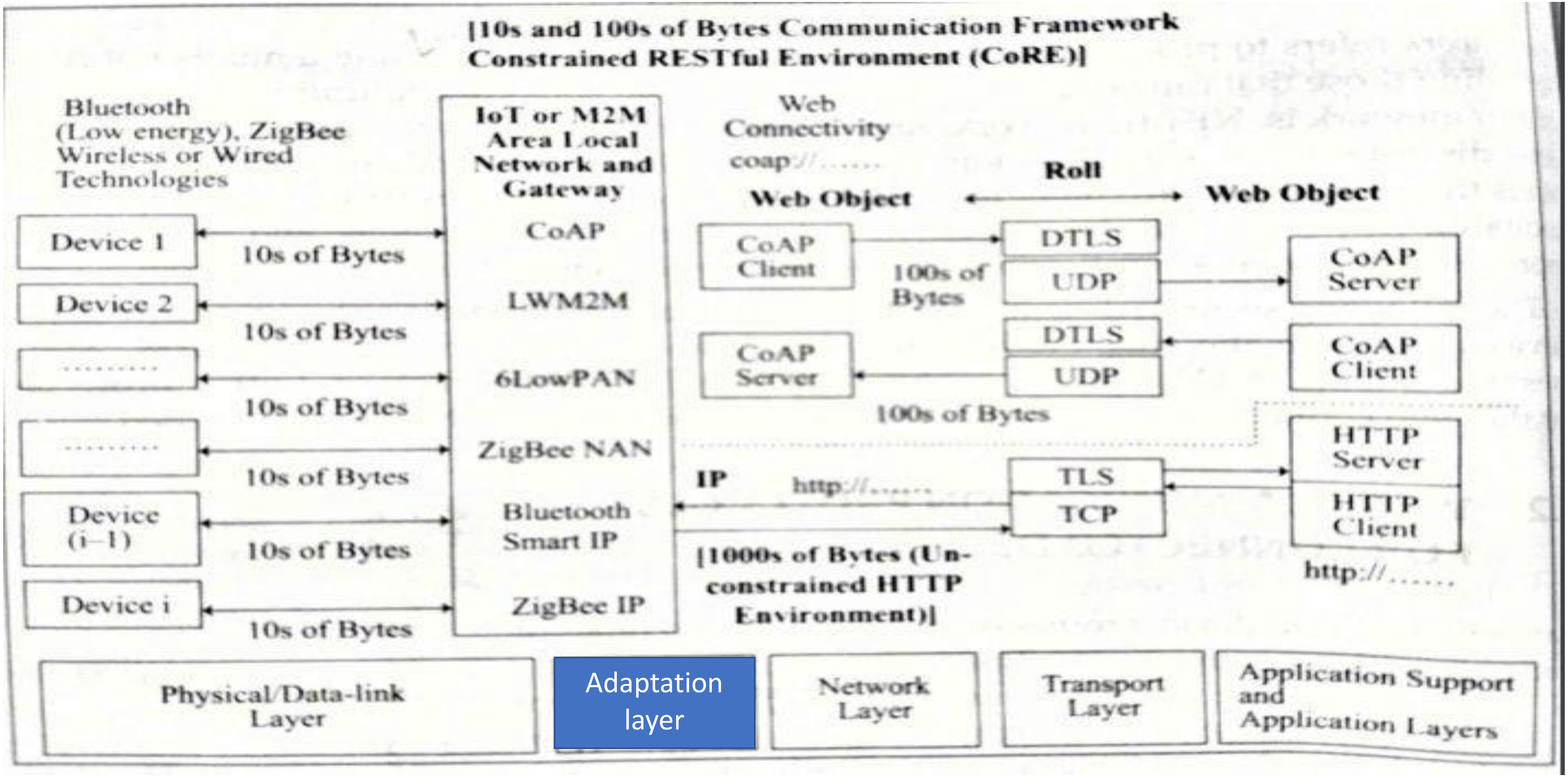
REST API

- *It acts as a waiter between client and server*
- *It access the resources on the client request and then respond this representation of the result to the client.*
- *REST API is a n intermediate service to access the resources.*

RESTful Web Services

1) ***RESTful webservices are webservices which are Rest based.***
2) ***Stateless and cacheable.***
3) ***Used URI and HTTP method to access the resource***
4) ***REST is based on HTTP***
5) ***Provide the communication between client and server over HTTP protocol.***
6) ***Most of the web browser uses REST API like Google APIs, Twitter APIs are all REST APIs***

Web Communication Protocol for connected devices



Web Communication Protocol for connected devices

The data of the connected devices routes over the web in two types of communication environment.

1) **Constrained RESTful Environment (CORE)**
2) **Unconstrained Environment**

3 Explain Constrained Application Protocol (CoAP) and XMPP protocol for IoT/M2M.

Constrained RESTful Environment (CORE)

- □ In Constrained environment (i.e. in CORE), the IOT devices sends small data (e.g. 10s of bytes), have limited resource, and runs on battery which powers off soon.

- □ In CORE environment when the data is routed on the network lead to the packet loss such a network is called as ROLL network (Routing over a network with low power and data loss)

- □ Another constrained of the CORE is low power devices can powered off soon and can have wake up time and sleep time like smart watches

Two protocols mainly work in CORE:

a) COAP (Constrained Application Protocol)
b) LWM2M (Light Weight Machine To Machine Protocol)

a) COAP (Constrained Application Protocol)

□ It is lightweight application layer protocol and web transfer protocol

□ Used for the constrained network i.e. it is designed for the transportation of small data between resource constrained nodes

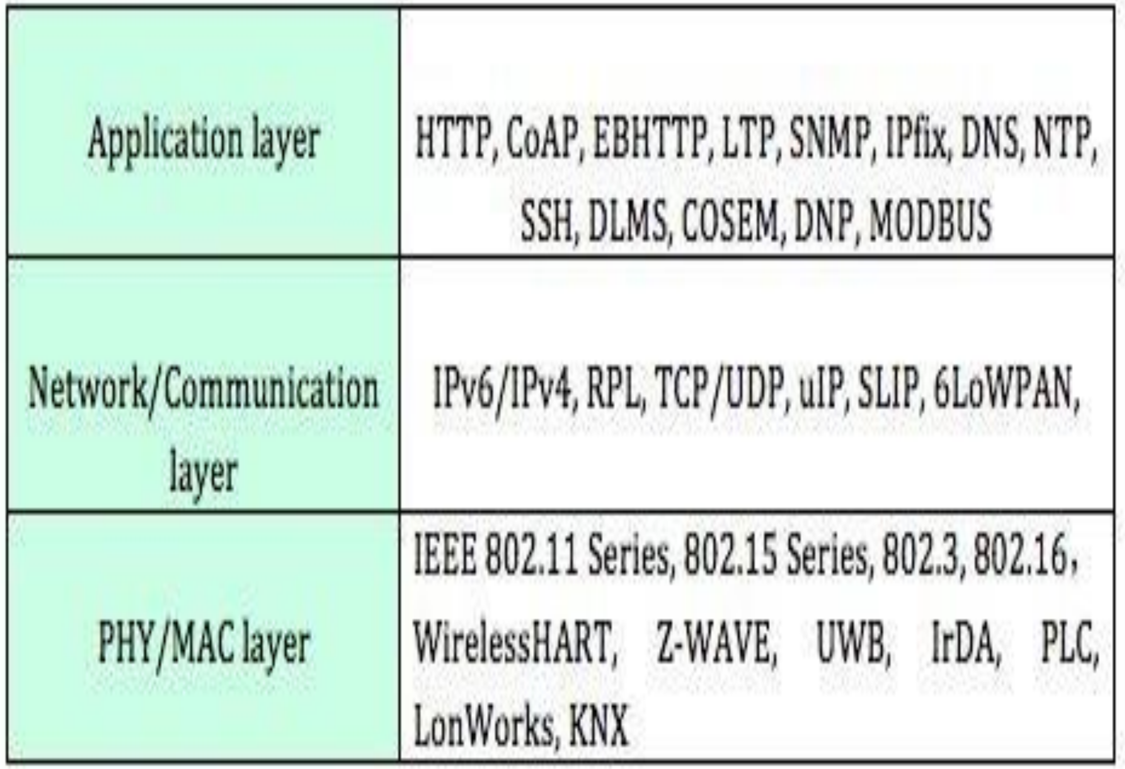
□ COAP is used for CORE using ROLL network.

□ Since COAP is a network oriented protocol therefore has similar features as http like
coap://........
http://.......

□ http is based on TCP for PTP communication.

□ CoAP is based on UDP for transmission over constrained network.

Table 1 protocols in different layers



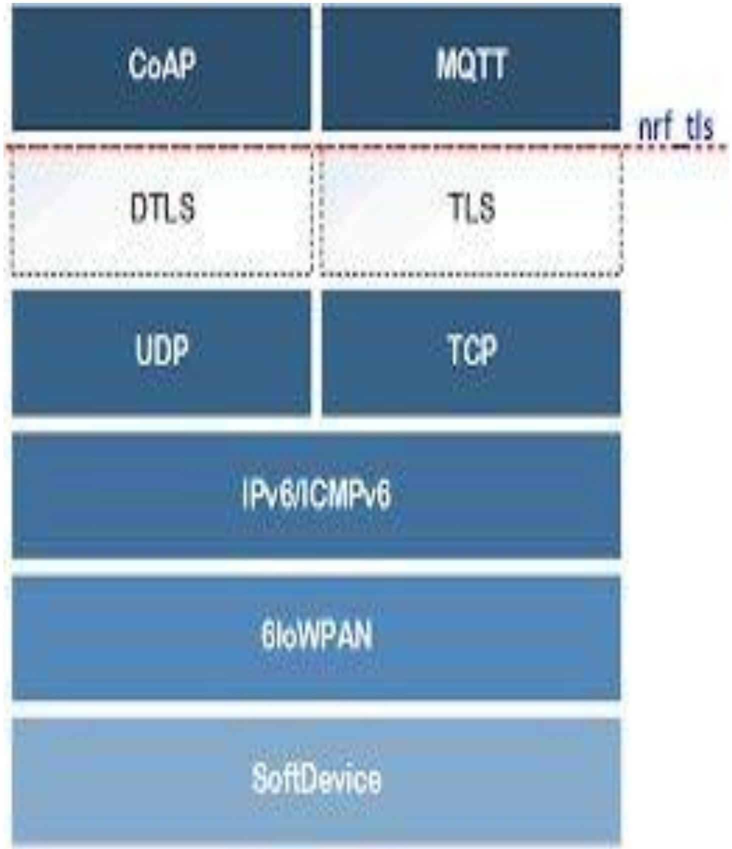
Application layer	HTTP, CoAP, EBHTTP, LTP, SNMP, IPfix, DNS, NTP, SSH, DLMS, COSEM, DNP, MODBUS
Network/Communication layer	IPv6/IPv4, RPL, TCP/UDP, uIP, SLIP, 6LoWPAN,
PHY/MAC layer	IEEE 802.11 Series, 802.15 Series, 802.3, 802.16, WirelessHART, Z-WAVE, UWB, IrDA, PLC, LonWorks, KNX

DTLS: (Datagram Transport Layer Security)

CoAP works with the UDP which is a unreliable protocol.

DTLS is application layer protocol which binds with UDP and provide

1) Security, integrity, authentication, confidentiality.
2) Packet retransmission
3) Assigning sequence number with handshake
4) Replay detection.



Message format of COAP

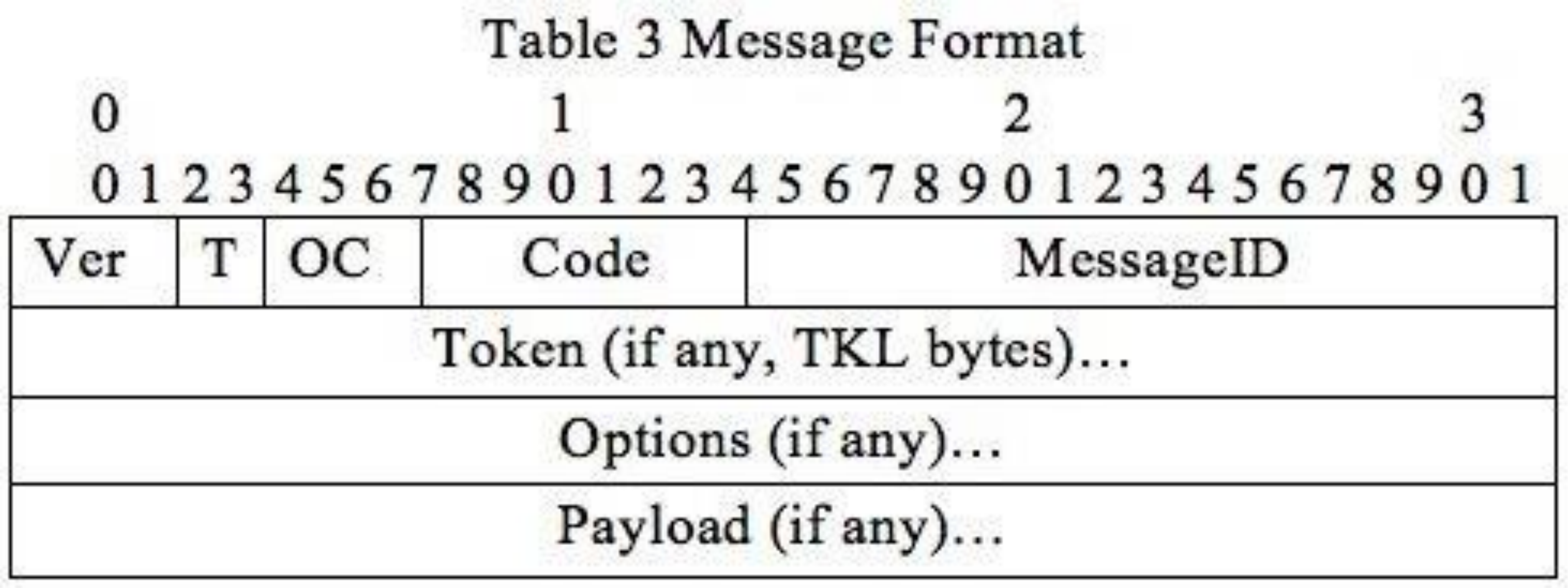
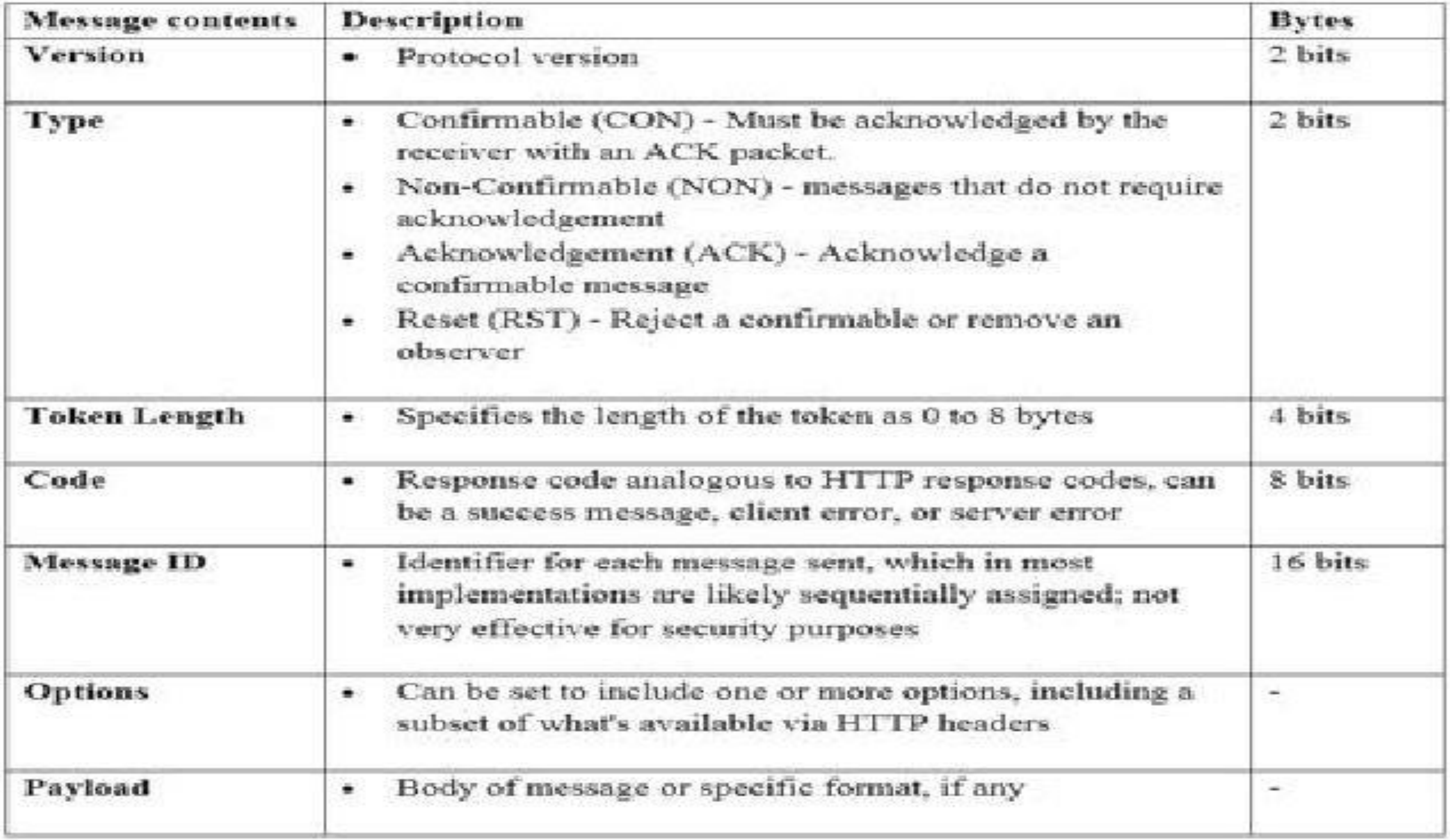


Table 3 Message Format

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 6 7 8 9 0 1

Ver	T	OC	Code	MessageID
Token (if any, TKL bytes)…				
Options (if any)…				
Payload (if any)…				



Message contents	Description	Bytes
Version	• Protocol version	2 bits
Type	• Confirmable (CON) - Must be acknowledged by the receiver with an ACK packet. • Non-Confirmable (NON) - messages that do not require acknowledgement • Acknowledgement (ACK) - Acknowledge a confirmable message • Reset (RST) - Reject a confirmable or remove an observer	2 bits
Token Length	• Specifies the length of the token as 0 to 8 bytes	4 bits
Code	• Response code analogous to HTTP response codes, can be a success message, client error, or server error	8 bits
Message ID	• Identifier for each message sent, which in most implementations are likely sequentially assigned; not very effective for security purposes	16 bits
Options	• Can be set to include one or more options, including a subset of what's available via HTTP headers	-
Payload	• Body of message or specific format, if any	-

Fig. 3: Details of CoAP message components

Unconstrained Environment

- □ Web application uses HTTP and RESTful HTTP for web client and web server communication.
- □ TCP/IP protocol is used by web services and web application.
- □ A huge amount of data (1000s) can be transmitted using HTTP protocol.

XMPP (Extensible Messaging and Presence Protocol)

□ ***XMPP uses XML technology for real time communication includes***
instant messaging (used in multiuser chat)
Presence
Collaboration.

□ The protocol is *used in constrained environment* for messaging.

□ It is also used for publish-subscribe systems, signaling for VoIP, video, file transfer, gaming etc.

XMPP (Extensible Messaging and Presence Protocol)

X- Extensible:

XMPP is designed to be *extensible*, in has been designed to *grow and accommodate changes*.

M-Messaging:

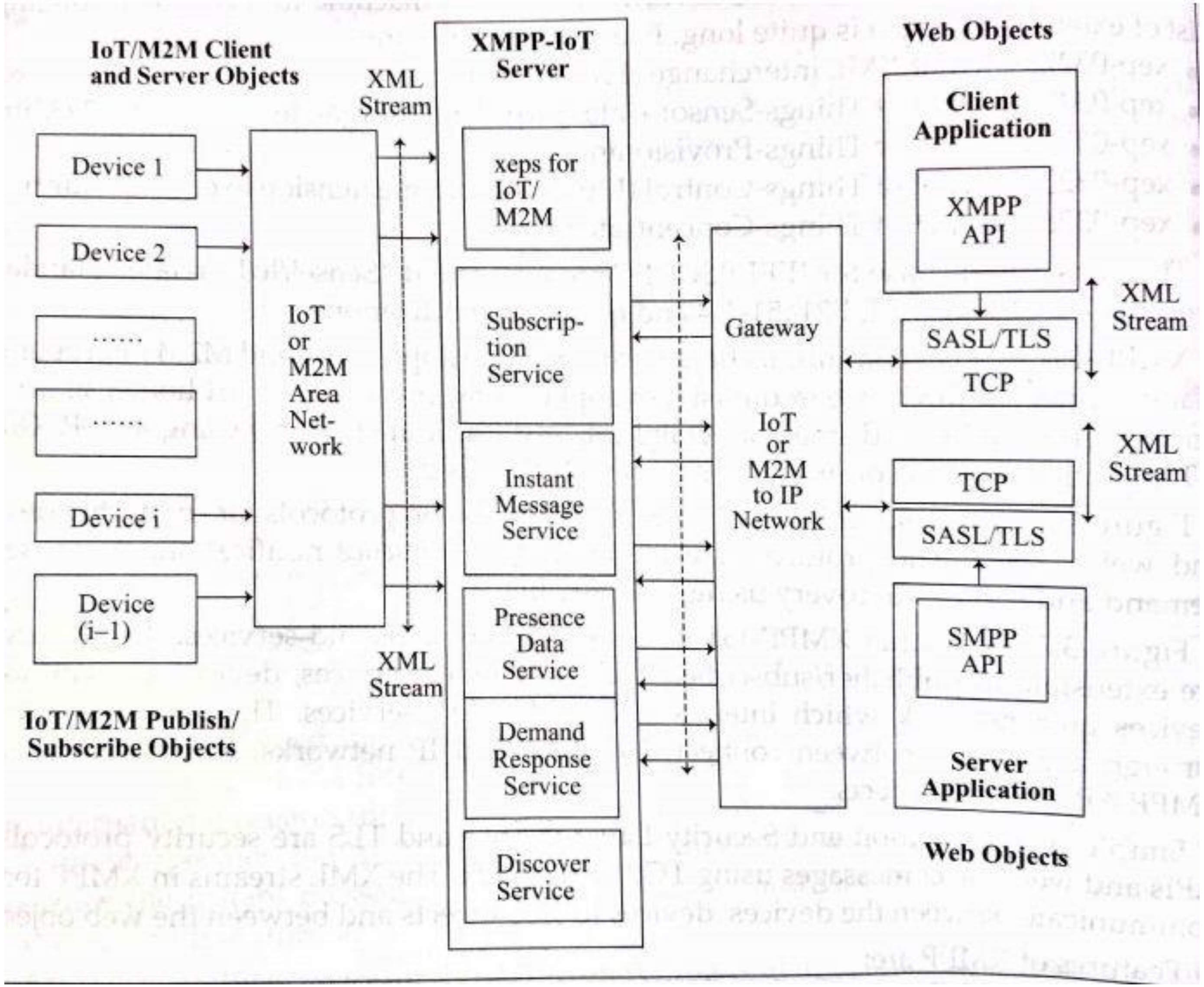
XMPP has been designed to send instant message.

P-Presence:

The presence indicator tells the server that you are online/offline/busy.

Protocol:

XMPP is a protocol; a set of standards to talk to each other. It is widely used across web but is unadvertised.



XMPP-IOT server/ XMPP M2M server is used for exchanging the messages between machines.
SASL (Simple Authentication and Security Layer)

XMPP

□ XMPP –IOT server consists of xeps and services like.

□ xeps are called as XMPP extension protocols. Examples of xeps are

- xep-DataForms Format,
- xep-XHTML-IM (instant messaging)
- xep-Service Discovery
- xep-MUC (Multi User Chat)
- xep-Publish-Subscribe and Personal eventing Protocol
- xep-File transfer
- xep-Jingle for voice and video

□ Services are like publish, subscribe, instant messaging, MUC etc

□ Service discovery identifies the protocols, features supported by client and servers as well as the items associated with an entity, such as the list of rooms hosted at a multi-user chat service.

XMPP Working

□ The IOT/M2M Client and Server objects communicate to XMPP-IoT server through the IoT/M2M Area Network.

□ Server provides necessary services like publish subscribe, instant messaging, multiuser chat etc.

□ Then XMPP server communicates to the XMPP APIs in the web objects through IOT gateway.

□ SASL (Simple Authentication and Security Layer) and TLS (Transport layer security) are security protocols for APIs and WEB Objects messages using TCP/IP network.

5 Explain message communication protocol MQTT with pub/sub model. Draw all the necessary figures.

MQTT (Message Queuing Telemetry Transport)

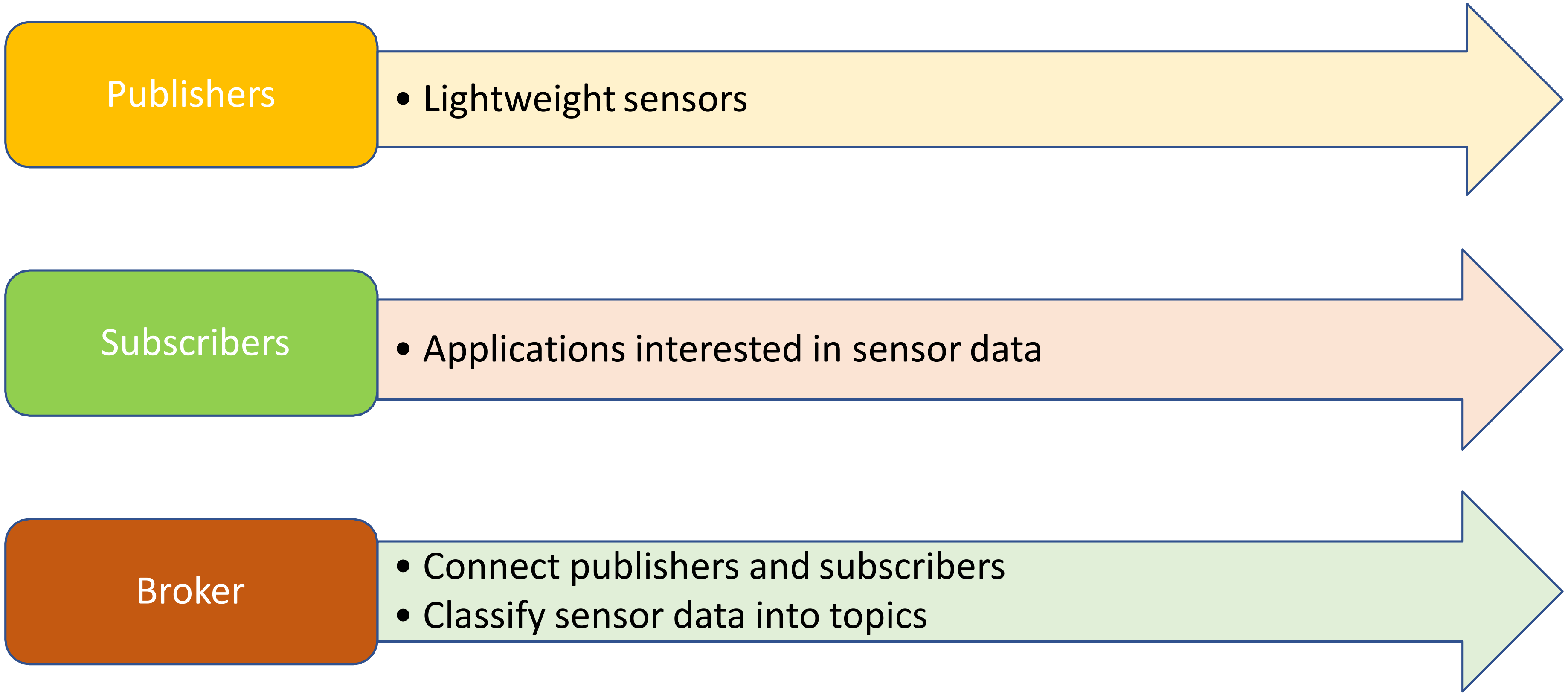
- An **open source protocol for machine-to-machine (M2M**)/"Internet of Things" connectivity

- **(Telemetry dictionary meaning is measuring and sending values or messages to far off places by radio or other mechanism)**

- Created by IBM IN 1999, as a constrained environment protocol.

- Designed to **provide connectivity** (mostly embedded) **between applications** and middle-wares **(M2M/IOT objects)** on one side **and networks and communications (WEB Objects)** on the other side.

MQTT Components



Publishers

- Lightweight sensors

Subscribers

- Applications interested in sensor data

Broker

- Connect publishers and subscribers
- Classify sensor data into topics

(Pub/Sub) Publish/Subscribe method is used for communication in place of request-response client-server architecture

MQTT Methods

Connect

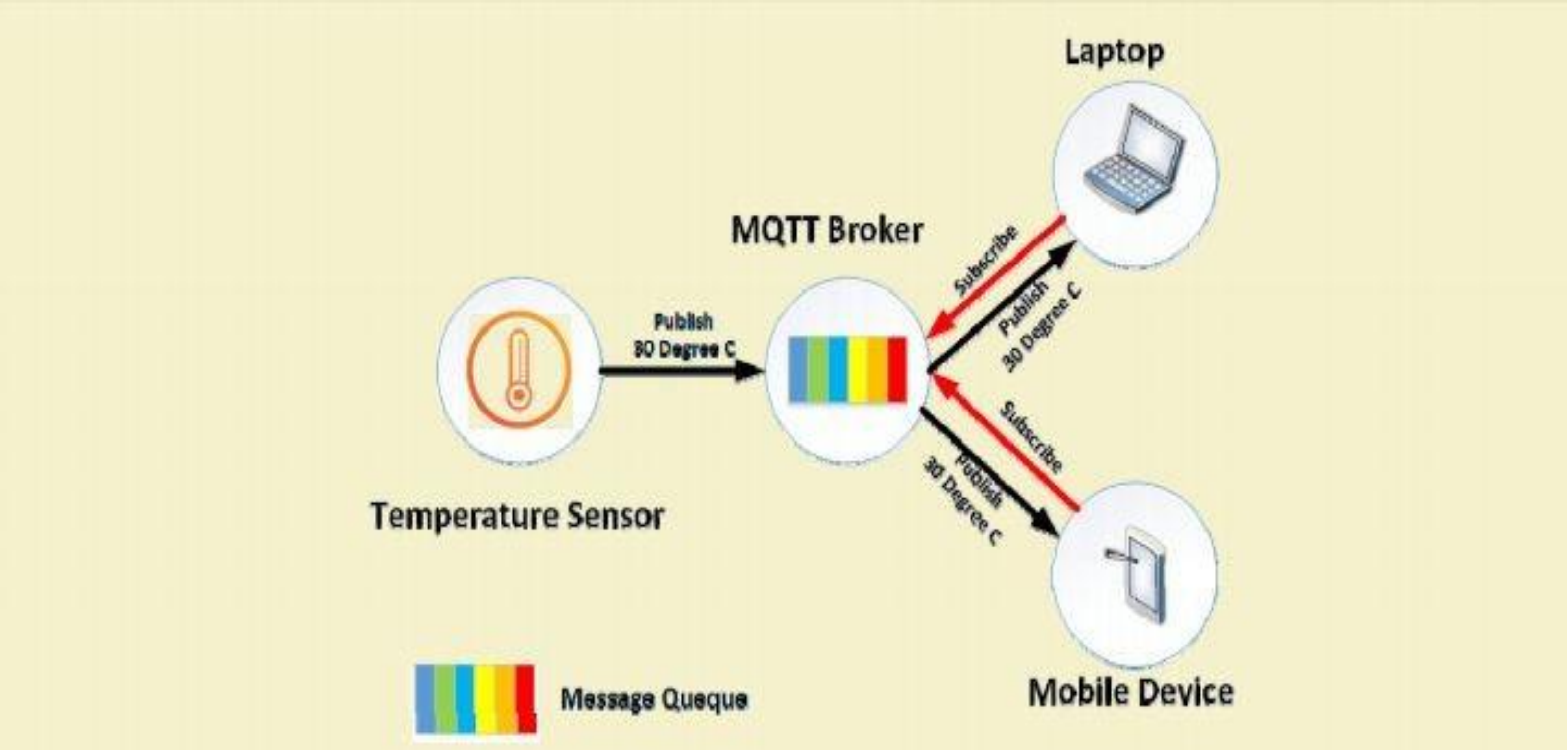
Disconnect

Subscribe

Unsubscribe

Publish

Communication



Communication

1) **Publisher:**

- They are the **light weight sensors also called as clients**
- These **clients first make connections to the Broker and then publish a message to the broker.**
- **The message include the topic**. The topic is the routing information for the broker.

2) **Broker:**

- Perform **store and forward** operation
- **Receives the topics** from **publishers**
- **Each client** that wants to receive messages **first subscribes to a certain topic** and then **the broker delivers all messages with the matching topic to the client**.

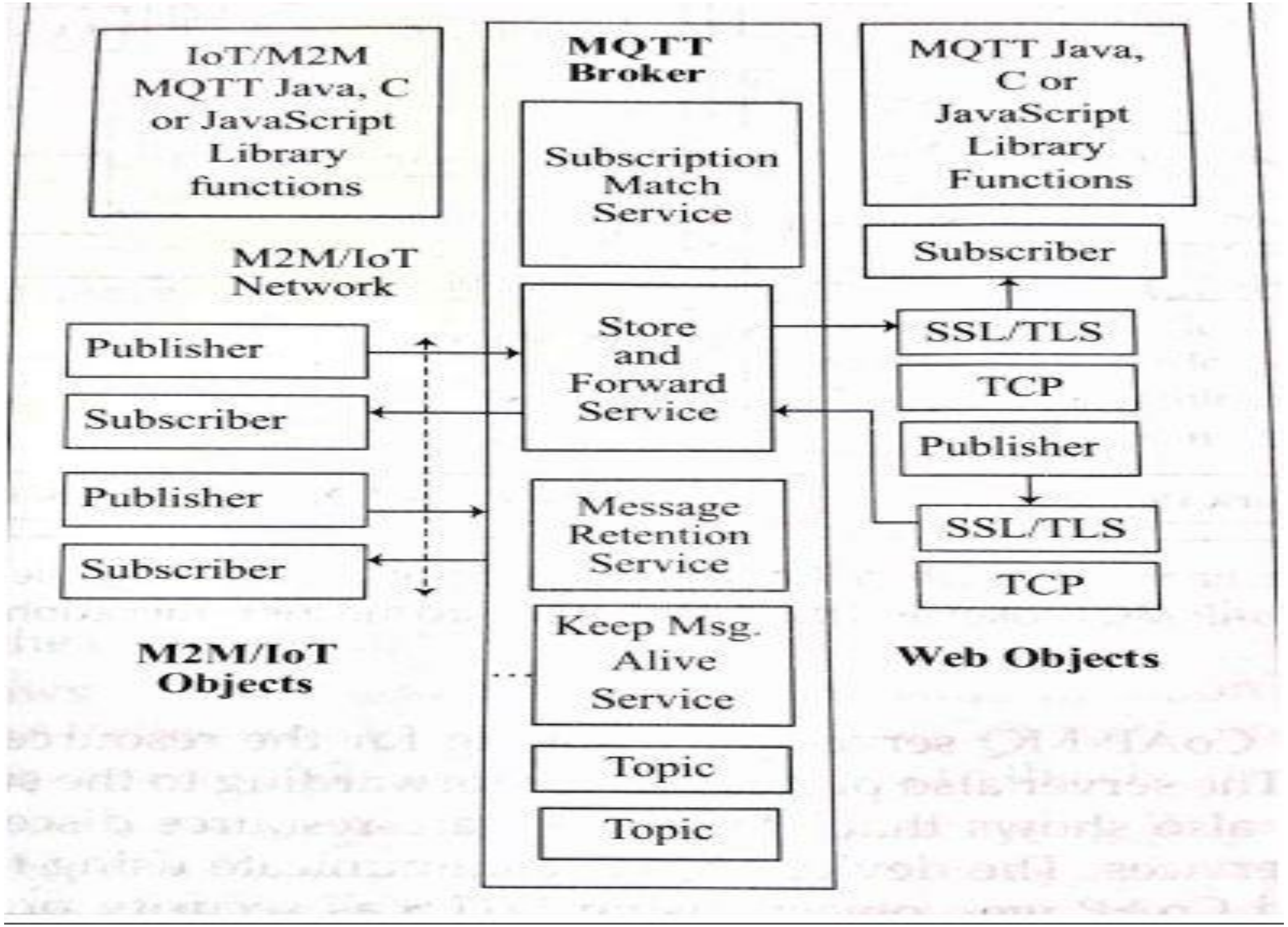
Therefore the clients don't have to know each other. They only communicate over the topic.

Communication

3) **Subscribers:**

- They are the clients that require the information from publishers

MQTT (Message Queuing Telemetry Transport)


IoT/M2M MQTT Java, C or JavaScript Library functions
M2M/IoT Network
Publisher
Subscriber
Publisher
Subscriber
M2M/IoT Objects
MQTT Broker
Subscription Match Service
Store and Forward Service
Message Retention Service
Keep Msg. Alive Service
Topic
Topic
MQTT Java, C or JavaScript Library Functions
Subscriber
SSL/TLS
TCP
Publisher
SSL/TLS
TCP
Web Objects

Designed to provide connectivity of M2M/IOT objects, web objects using different protocols.

MQTT BROKER :

Store and forward service:

- □ It performs store and forward operation, **stores the topics from publishers and forward it to subscriber**
- □ **Publishers/ Subscribers can be M2M/IOT objects** which communicate by using network protocols like Zigbee **Or it can be Web Objects** which communicate over IP network using SSL and TLS security protocols.
- □ The figure shows device objects and Web objects uses MQTT Java, C or JavaScript Library functions.

Subscription Match Service:

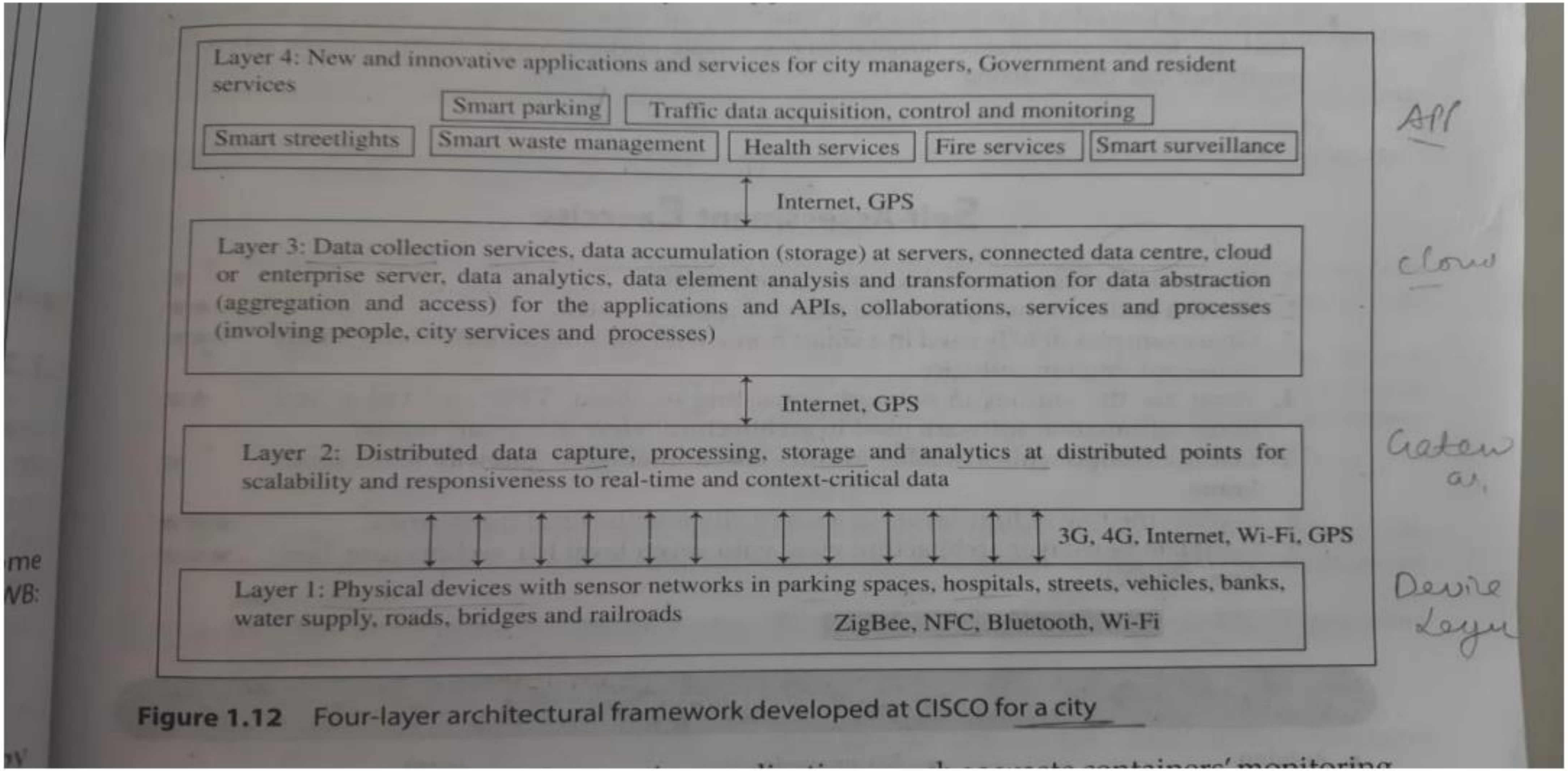
It matches the subscriptions and forward it to right subscribers in order to route the message to the right endpoints.

Message Retention Service:
When this field is set the **broker retains the last receives message from the publisher for the new connected subscriber on the same topic.**

Keep Message Alive:
Until the DISCONNECT message is received the message is saved by setting the **“Keep Message Alive Service”**

6 Write and explain four layer architectural frameworks developed at CISCO for a city.

Four Layer architecture framework developed by CISCO



Layer 4: New and innovative applications and services for city managers, Government and resident services

Smart parking | Traffic data acquisition, control and monitoring

Smart streetlights | Smart waste management | Health services | Fire services | Smart surveillance

Internet, GPS

Layer 3: Data collection services, data accumulation (storage) at servers, connected data centre, cloud or enterprise server, data analytics, data element analysis and transformation for data abstraction (aggregation and access) for the applications and APIs, collaborations, services and processes (involving people, city services and processes)

Internet, GPS

Layer 2: Distributed data capture, processing, storage and analytics at distributed points for scalability and responsiveness to real-time and context-critical data

3G, 4G, Internet, Wi-Fi, GPS

Layer 1: Physical devices with sensor networks in parking spaces, hospitals, streets, vehicles, banks, water supply, roads, bridges and railroads

ZigBee, NFC, Bluetooth, Wi-Fi

API

cloud

Gateway

Device Layer

Figure 1.12 Four-layer architectural framework developed at CISCO for a city

Layer 1 consists of sensors, sensor networks and device networks in parking spaces, hospitals, streets, vehicles, banks, water supply, roads, bridges and rail roads. Bluetooth, Zigbee, NFC, WiFi and other short range communications protocols are used at the bottom layer.

Layer 2 captures data at distributed computing points where data is processed, stored and analyzed.

Layer 3 is meant for central collection services, connected services, connected data centers, cloud and enterprise servers for analytics applications.

Layer 4 consists of new innovative applications, such as waste containers monitoring, WSNs for power loss monitoring, bike sharing management, smart parking, smart surveillance and much more.

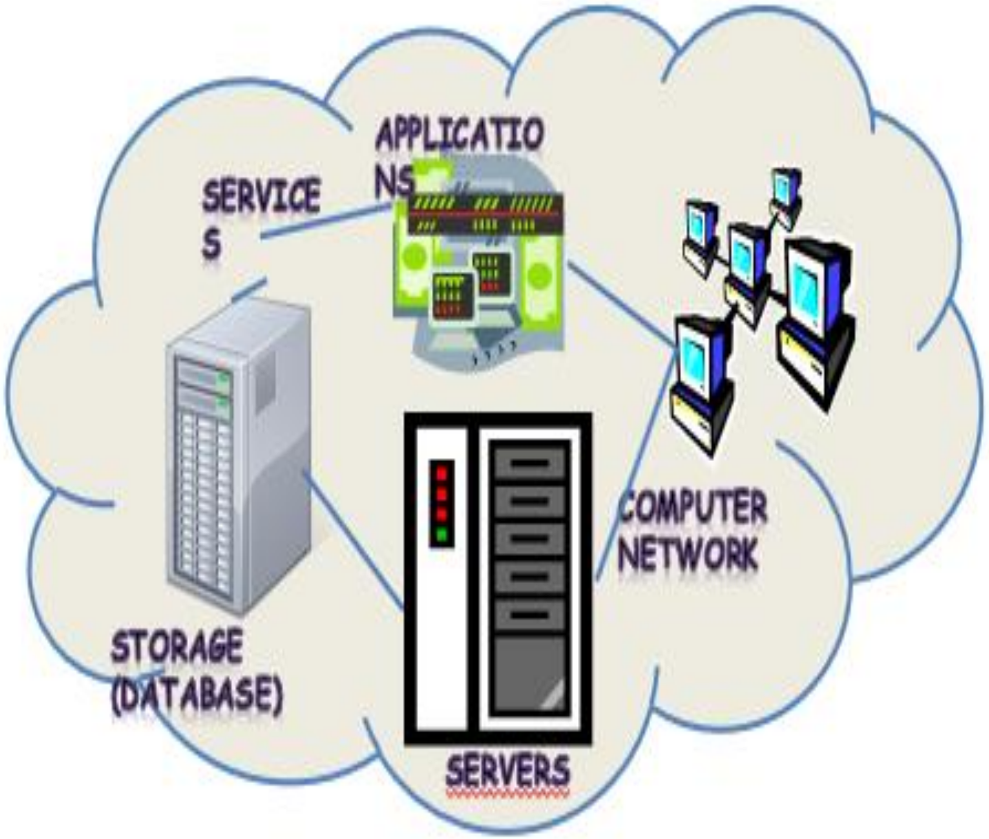
7 Explain about cloud service and deployment models giving examples.

What is cloud computing

- A collection of integrated and networked hardware, software, Internet infrastructure (called a platform) with a huge storage, computing capabilities.

- It provides hardware, software and networking services to clients on rent.

- These platforms hide the complexity and details of the underlying infrastructure from users and applications using API



Cloud Computing Paradigm

- Cloud computing is a method in which resources are retrieved from the Internet through web-based tools and applications, without using direct connection to a server.

- Cloud-based storage makes it possible to save the files to a remote database instead of keeping them on a hard drive or local storage device.

- It's called cloud computing because it does not require a user to be in a specific location to gain access to it. This type of system allows the users to store files and applications on remote servers, and then access It via the internet any time.

Cloud Computing Paradigm

- It also allows us to create, configure, and customize applications online.

- With Cloud Computing users can access database resources via the internet from anywhere for as long as they need without worrying about any maintenance or management of actual resources

Cloud computing Features and advantages

- Provide provision of **unlimited storage, computing servers, software delivery and servers** to the users on rent i.e. on demand.
- Provide **resource pooling** in multi-tenant model.
- Provide **broad network and remote access to heterogeneous users**, clients, systems and devices in virtualised environment.
- More **flexible and efficient allocation of resources**.
- Provides **elasticity, scalability, maintainability of resources**.
- It **lowers the cost** of IT infrastructure.
- Advanced **security and reliability**.
- Enables running **multiple operating system**.
- Always **Latest version availability**

Working model of cloud computing

There are two kinds of working models:

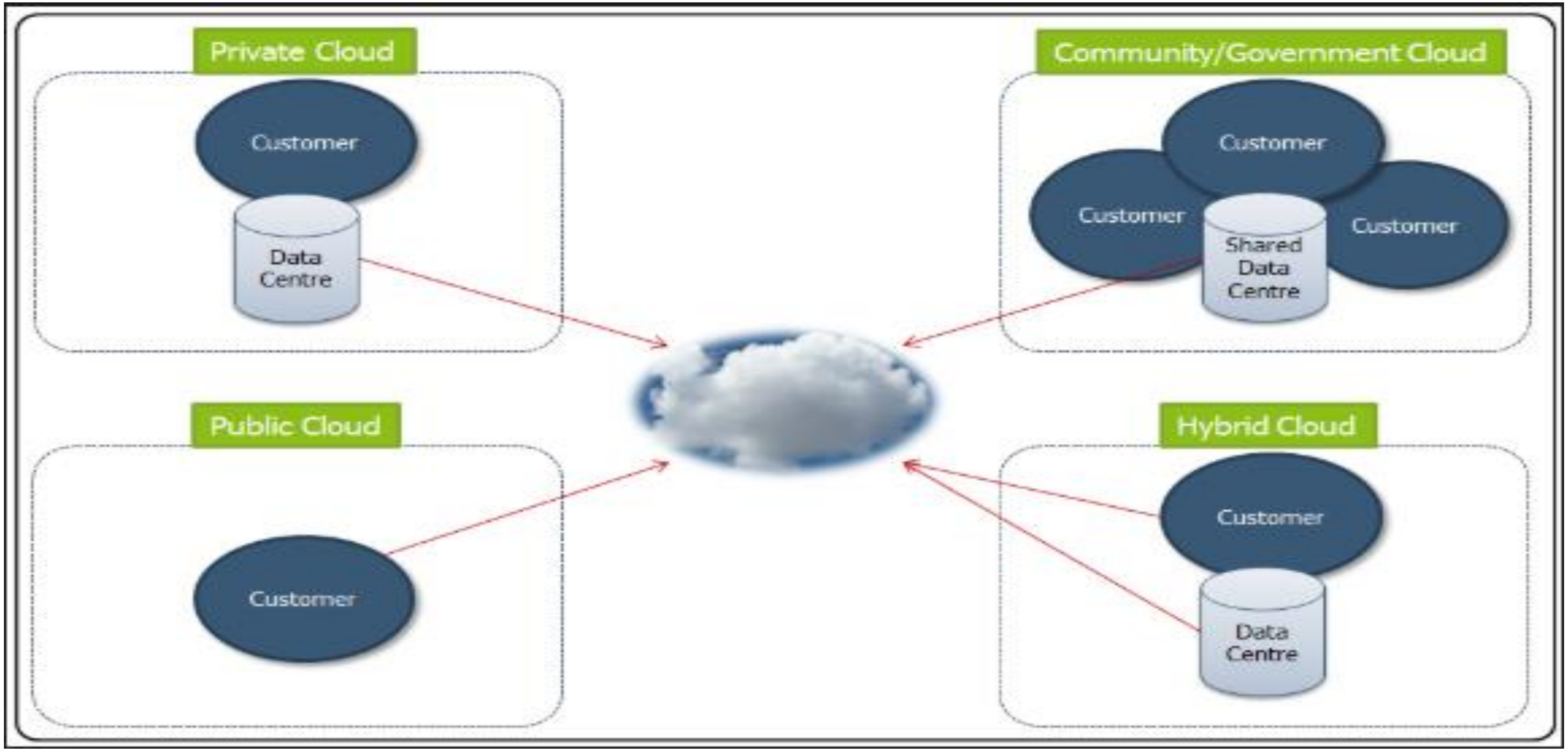
1. **Deployment Models**
2. **Service Models**

Deployment Models

- Deployment models define **access method** of the cloud.
- Depending upon the access method the cloud are defined as.

- Public
- Private
- Hybrid
- Community

Deployment Model



PUBLIC CLOUD :

The Public Cloud allows **systems and services to be easily accessible to the general public.** Public cloud may be less secure because of its openness, e.g., e-mail. Amazon Elastic Compute **Cloud** (EC2), IBM's Blue **Cloud**, Sun **Cloud**, Google AppEngine and Windows Azure Services Platform

PRIVATE CLOUD :

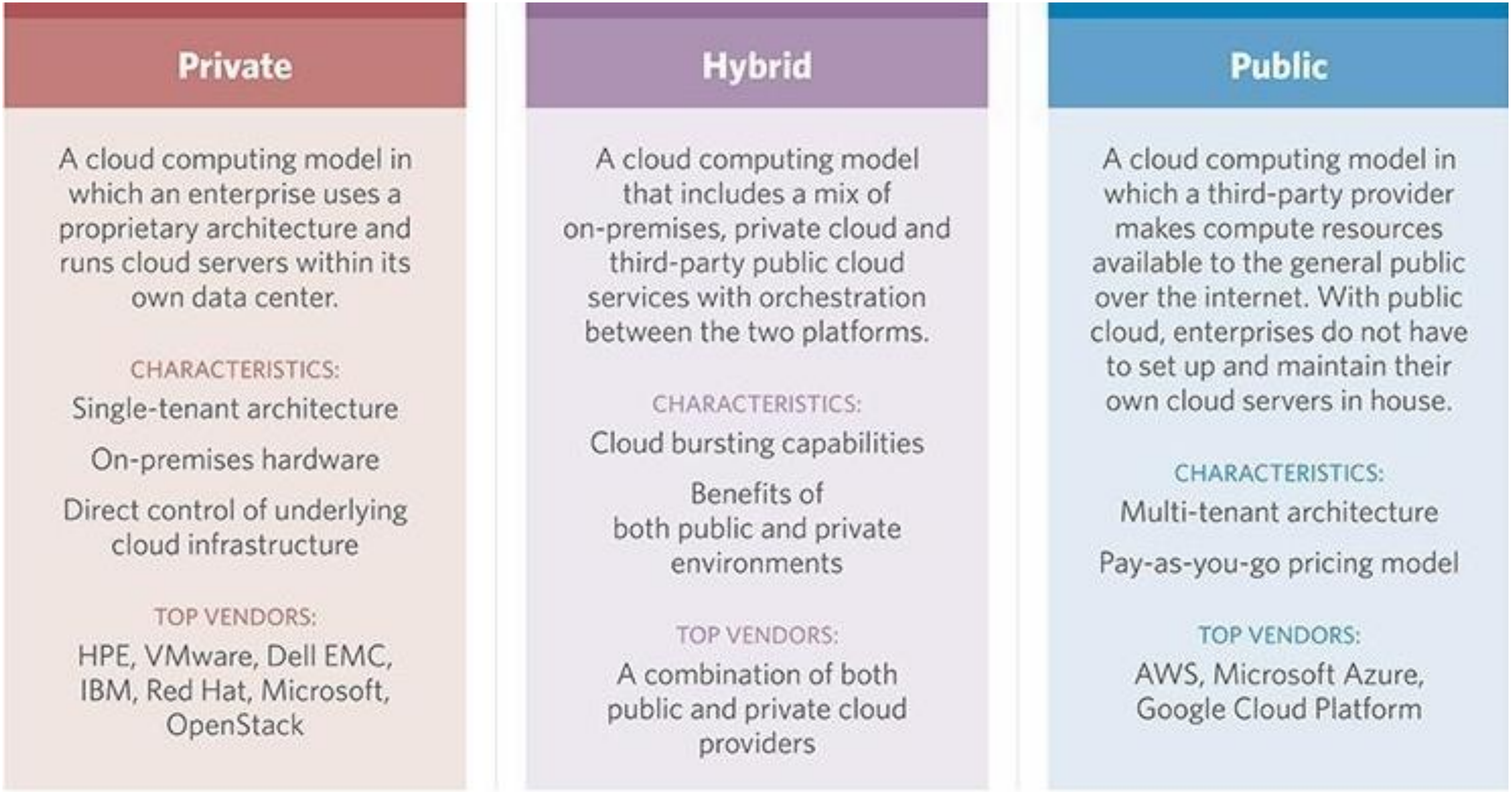
The Private Cloud **allows systems and services to be accessible within an organization**. It offers increased security because of its private nature for example Oracle, IBM, Redhat

COMMUNIY CLOUD :

The **Community Cloud** allows **systems and services to be accessible by group of organizations.**

HYBRID CLOUD :

The **Hybrid Cloud** is **mixture of public and private cloud.**



Private	Hybrid	Public
A cloud computing model in which an enterprise uses a proprietary architecture and runs cloud servers within its own data center.	A cloud computing model that includes a mix of on-premises, private cloud and third-party public cloud services with orchestration between the two platforms.	A cloud computing model in which a third-party provider makes compute resources available to the general public over the internet. With public cloud, enterprises do not have to set up and maintain their own cloud servers in house.
CHARACTERISTICS:	CHARACTERISTICS:	CHARACTERISTICS:
Single-tenant architecture On-premises hardware Direct control of underlying cloud infrastructure	Cloud bursting capabilities Benefits of both public and private environments	Multi-tenant architecture Pay-as-you-go pricing model
TOP VENDORS:	TOP VENDORS:	TOP VENDORS:
HPE, VMware, Dell EMC, IBM, Red Hat, Microsoft, OpenStack	A combination of both public and private cloud providers	AWS, Microsoft Azure, Google Cloud Platform

Service Models

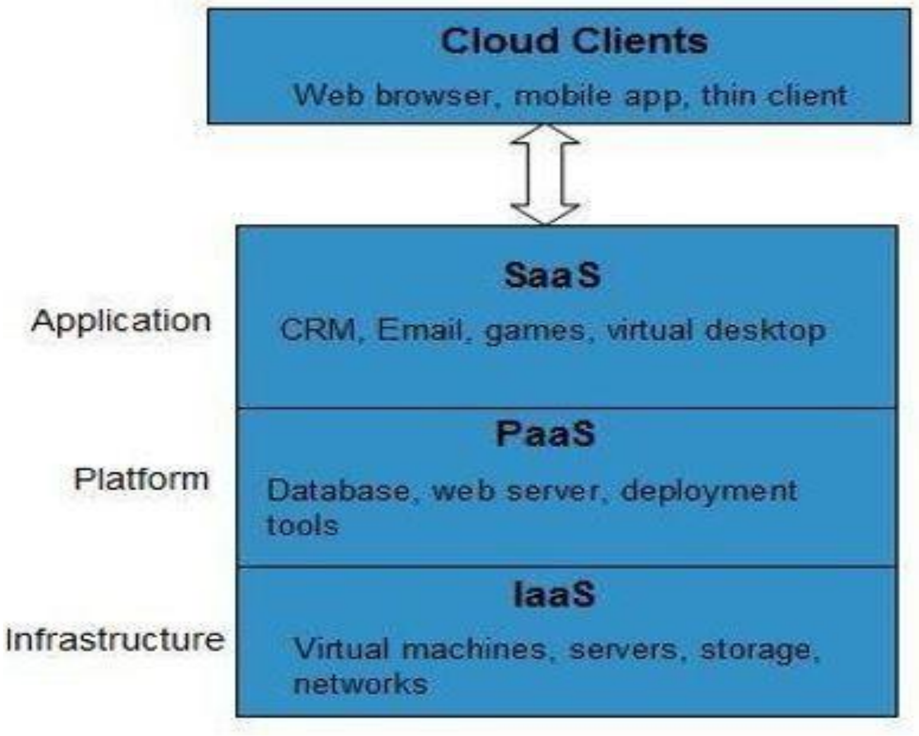
Cloud computing is not a single piece of technology, like a microchip or a cell phone. Rather, it's a system, primarily comprised of three services:

Service Models are the reference models on which the Cloud Computing is based. These can be categorized into three basic service models as listed below:

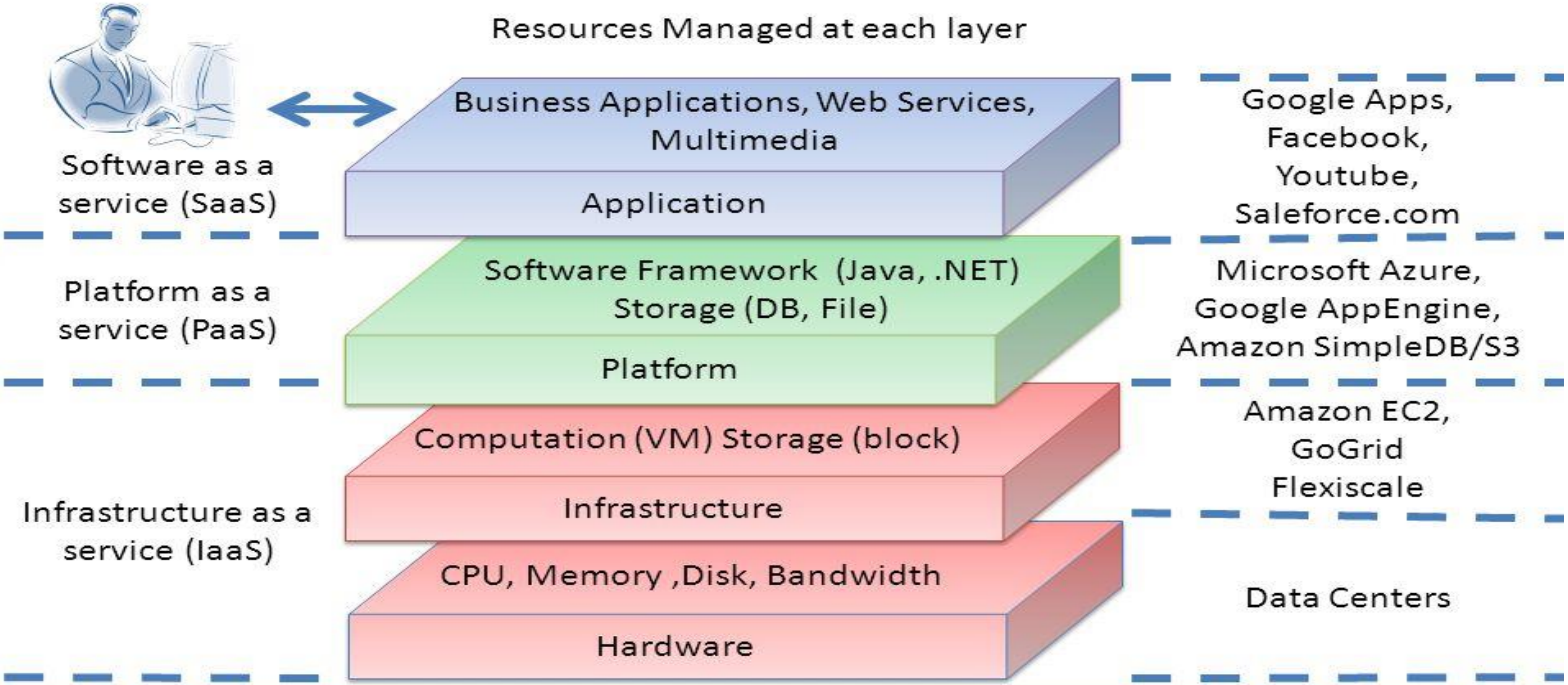
1. **Infrastructure as a Service (IaaS)**

2. **Platform as a Service (PaaS)**

3. **Software as a Service (SaaS)**


Cloud Clients
Web browser, mobile app, thin client
SaaS
CRM, Email, games, virtual desktop
Application
PaaS
Database, web server, deployment tools
Platform
IaaS
Virtual machines, servers, storage, networks
Infrastructure

Cloud Computing Layers



Resources Managed at each layer

Software as a service (SaaS)

Business Applications, Web Services, Multimedia

Application

Google Apps, Facebook, Youtube, Saleforce.com

Platform as a service (PaaS)

Software Framework (Java, .NET) Storage (DB, File)

Platform

Microsoft Azure, Google AppEngine, Amazon SimpleDB/S3

Infrastructure as a service (IaaS)

Computation (VM) Storage (block)

Infrastructure

Amazon EC2, GoGrid Flexiscale

CPU, Memory ,Disk, Bandwidth

Hardware

Data Centers

Software as a Service (SaaS)

SaaS is a method for delivering software applications over the Internet, on rent (i.e. on demand) to the end users typically on a subscription basis.

There are several SaaS applications, some of them are listed below:

- Billing and Invoicing System
- Help Desk Applications
- Human Resource (HR) Solutions
- Customer Relationship Management (CRM) applications

Benefits of Software as a service (SaaS):

➢SaaS supports **multitenant environment** i.e. it allows access to programs and recent software to large number of users through browser.

➢A SaaS provider **gives access to applications to multiple clients and users over web** by hosting and managing the given application in their or leased datacenters.

➢Provides **centralized management of data** like software control, maintenance, updation to new version, infrastructure, platform and resource requirements etc.

Issues of Software as a Service (SaaS)

- Browser based risks
- Network dependence
- Lack of portability between SaaS clouds

SaaS Examples

Microsoft Online Services: Business Productivity Online Suite

Microsoft SharePoint Online

Microsoft Office Communications Online

Microsoft Exchange Online

Microsoft Office Live Meeting

Platform as a Service (PaaS)

- Here instead of delivering software online, it supplies or rent a platform over some time for creating developing, testing, delivering and managing software applications like web or mobile apps, without worrying about setting up or managing the underlying infrastructure of servers, storage, network and databases needed for development

Google's App Engine, Force.com are examples of PaaS offering vendors.

Platform as a Service (PaaS)

Benefits :

- LOWER **ADMINISTRATIVE OVERHEAD**
- LOWER **COST OF OWNERSHIP**
- **SCALABLE SOLUTIONS**
- **MORE CURRENT SYSTEM SOFTWARE**

Issues :

- LACK OF PORTABILITY BETWEEN PAAS CLOUDS
- EVENT BASED PROCESSOR SCHEDULING
- SECURITY ENGINEERING OF PAAS APPLICATIONS

PaaS Examples

Nimbits

Xively

Infrastructure as a Service (IaaS)

- It rent complete IT infrastructure to the user like—servers and virtual machines (VMs), storage, data center, networks, operating systems through IP-based connectivity as part of an on-demand service.

- Cloud paradigm also serves as a business model apart from technology.

Infrastructure as a Service (IaaS)

IaaS provides access to fundamental resources such as physical machines, virtual machines, virtual storage, etc.

Apart from these resources, the IaaS also offers:

- Virtual machine disk storage
- Virtual local area network (VLANs)
- Load balancers
- IP addresses
- Software bundles

Infrastructure as a Service (IaaS)

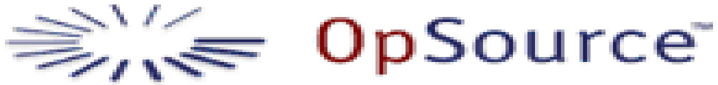
Benefits :

- **Full Control through Administrative Access to VMs** on the computing resource.
- **Flexible and Efficient renting** of Computer Hardware.
- **Portability, Interoperability** with Legacy Applications.

Issues :

- COMPATIBILITY WITH LEGACY SECURITY VULNERABILITIES
- VIRTUAL MACHINE SPRAWL
- DATA ERASE PRACTICES

IaaS Examples



NTT Communications

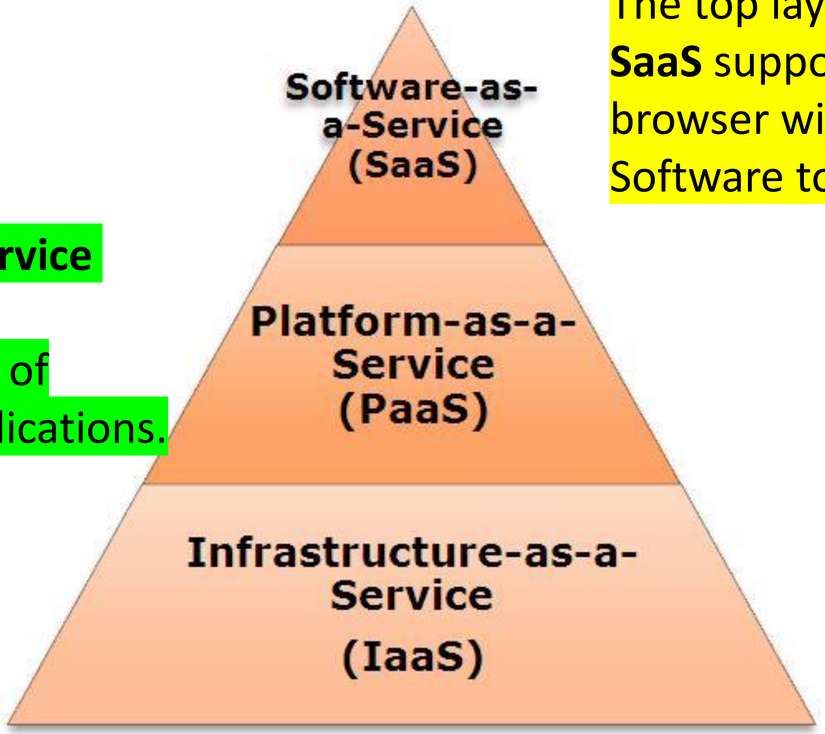
BLIZZARD
ENTERTAINMENT

CLOUD SERVICE MODEL:

Layers in Cloud:

Different layers are outlined based on the kind of services provided by the Cloud

The top layer is **Software-as-a-Service (SaaS)**. **SaaS** supports accessing user's applications through a browser without the knowledge of Hardware or Software to be installed.


Software-as-a-Service (SaaS)
Platform-as-a-Service (PaaS)
Infrastructure-as-a-Service (IaaS)

Middle layer is **Platform-as-a-Service (PaaS)** which mainly supports deployment and dynamic scaling of .NET, Python and Java based applications. One such an example of **PaaS is Google App Engine**.

Bottom layer contains basic hardware resources like Memory, Storage Servers. Hence it is denoted as **Infrastructure-as-a-Service (IaaS). For example Amazon easy Storage Service (S3) and Amazon Elastic Compute Cloud (EC2).**

- **Cloud Computing can be considered as = SaaS+ PaaS+IaaS**

- Therefore Cloud connects the devices, data, applications, service, persons and business.

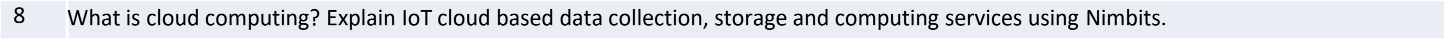
- Cloud services are considered as **distribution service -a service** for linking the resources (computing functions, data store, processing functions, networks, servers and applications) and provisions of coordinating between the resources.

8 What is cloud computing? Explain IoT cloud based data collection, storage and computing services using Nimbits.

Cloud Computing Paradigm

Cloud computing is the delivery of computing services —servers, storage, databases, networking, software, hardware, analytics and more—over the Internet (“the cloud”). Cloud delivers the computational functionality.

Companies offering these computing services are called cloud service providers and typically charge for cloud computing services based on usage, similar to how you are billed for water or electricity at home.



8	What is cloud computing? Explain IoT cloud based data collection, storage and computing services using Nimbits.

IoT Cloud based data collection, storage and Computing Services using Nimbits

- Nimbits is a platform as a service (PaaS) used to develop software and hardware solutions that seamlessly connect to the cloud and each other.
- Nimbits server runs on powerful cloud platforms like Google App Engine to the smallest Raspberry Pi device.
- Nimbits server is a web portal and API designed to

- Provides time-stamping or geo-stamping on incoming data.
- Store and process that time and location stamped data over cloud (pushing the data over cloud and store them in a data point)
- Provide filtering to incoming data from noise, add important changes to it and then generate trigger events and alerts based on rules and then sending them in real time over internet.
- It provides rule engine for connecting sensors, persons and software to cloud.

The basic services offered by Nimbits are

Nimbits clients can plot charts and graphs of real time collected data over the internet.

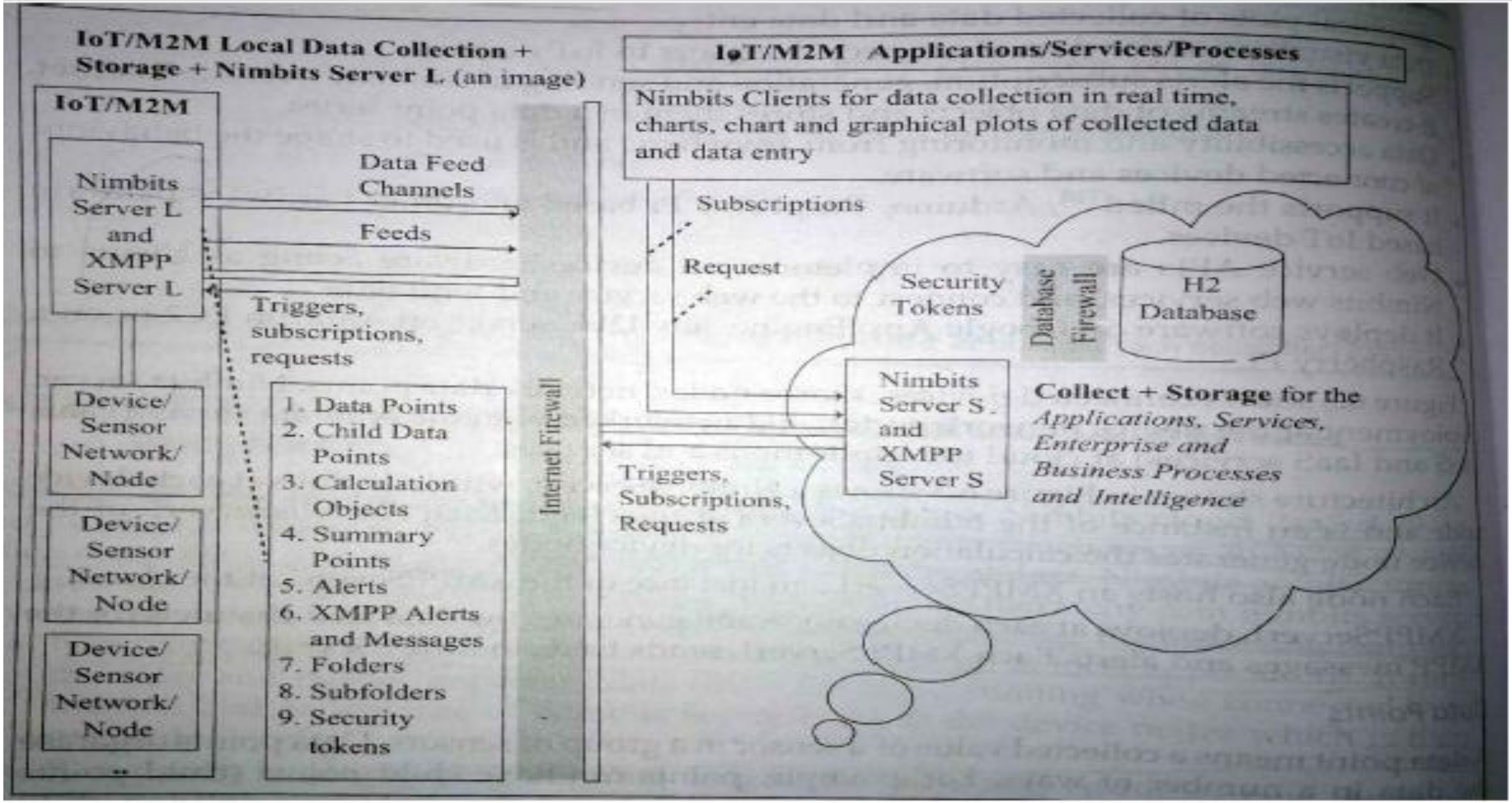
It supports many format like text, JSON or XML values into the cloud.

It provides edge computing locally on devices and nodes**.**

It supports multiprogramming languages, M2M communication and hardware platform of IoT devices like mbed, Arduino, raspberry Pi based etc.

Nimbits data points can relay data between the software systems or hardware devices like Arduino, using cloud as backend.

It provides data logging services, access and data monitoring from anywhere.


IoT/M2M Local Data Collection + Storage + Nimbits Server L (an image)
IoT/M2M
Nimbits Server L and XMPP Server L
Device/ Sensor Network/ Node
Device/ Sensor Network/ Node
Device/ Sensor Network/ Node
..
Data Feed Channels
Feeds
Triggers, subscriptions, requests
1. Data Points
2. Child Data Points
3. Calculation Objects
4. Summary Points
5. Alerts
6. XMPP Alerts and Messages
7. Folders
8. Subfolders
9. Security tokens
Internet Firewall
IoT/M2M Applications/Services/Processes
Nimbits Clients for data collection in real time, charts, chart and graphical plots of collected data and data entry
Subscriptions
Request
Security Tokens
Database Firewall
H2 Database
Nimbits Server S and XMPP Server S
Collect + Storage for the Applications, Services, Enterprise and Business Processes and Intelligence
Triggers, Subscriptions, Requests

The Figure shows the connected devices, sensor nodes, network data points, nimbits server, deployment at the device network nodes and networked with the nimbits server (PaaS, SaaS and IaaS services) at cloud for applications and services.

Working

The Nimbits serverL : It is an instance of Nimbits serverS at the cloud

- The Nimbits serverL is deployed at each device node.

- The Nimbits servers first store, then filter and clean the data from noise, add some important changes, provides time-stamping or geo-stamping to it and send events, alerts (like email alerts or push notifications) by using rules an calculations called as **data feeds channels** (A data feed contains latest updates of current information like events, alerts etc) using XMPP messages.
- These data feeds (notifications) are sent over data feed channel using XMPP (Extensible Messaging and Presence Protocol) messages

- This pushing of the **alerts and messages down quickly or repeatedly is called as "Jabbing".**

- Hence server relay that data up to a website or to a some mobile device

Data Points

Data point means a collected value of a sensor in a group of sensors.
Data points organise the data in a number of ways.
For example points can have child points (mean subpoints), points can be in folders, the folders can go as deep as like a tree (Tree means a folder having a subfolders and so on).
Any type of document can be organised with the points.

Child Data Points

It means the **subpoints stored in the folders** (like subfolder)

Data Feed

A data feed is a special point (an ongoing stream of structured data) that **provides users with updates of current information (like events, alerts etc) from one or more sources**.
Feed consists of time/ geo-stamped data points, streams and alerts.
Feed for messages and alerts generated on application of rules for filtering and calculation.

Data feed channels

User create data feed channels which shows system events, messages, data alerts also called as feed.

The user can subscribe to the data point of other users also and configure the subscriptions to send messages to the feed.

Calculations objects

A user create calculations objects for a point.
The user can apply many formulas for a single data point e.g. for temperature sensor on formula is for increase in last value other formula is for increase over a normal value and then organise the objects in a tree.

Summary points

A user can create **summary point which can compute averages, minimum, maximum. Standard deviations, variance and sums** of another point on a specific time interval basis.

Folders, subfolders

- Data points can be stored in folders and subfolder

Security tokens

- Used for authentication and authorization.

Alerts and XMPP alerts and messages

The **Nimbits servers filter and clean the data from noise,** get important data and then relay that data up to a website or to a some mobile device

Hence **when the data is passed form sensors to Nimbits server, it adds rules and the formulas calculations to the incoming data and also execute triggers** like email alerts or push notifications **which are sent or broadcasted over XMPP (**Extensible Messaging and Presence Protocol**)** and those triggers generated data i.e. **the data feeds that further cascade with other data feeds** and this cascading result could be an alert which indicates that some relevant changes are happening within this massive system.

The filter item is called as "ah" for XMPP alerts and use custom Jabber IDs.

Jabbing

Jabbing means pushing the alerts and messages down quickly or repeatedly
And each alert and message is assigned a jabber ID called JID.

JIDs consist of three main parts:

- The **node identifier** (optional)
- The **domain identifier** (required)
- The **resource identifier** (optional)

Folders, subfolders

Data points can be stored in folders and subfolder

Security tokens

- Used for authentication and authorization.