MODULE 1

* IOT Definition
* ~~Vision of~~ IOT [~~Smart~~, ~~Hyperconnectivity~~]
* GENERAL Framework [~~Hub~~, Edge, Things]
* CONCEPTUAL Framework [Oracle, IBM]
* REFERENCE Model
* ARCHITECTURAL Framework [Cisco]
* Major Components of IOT
* M2M Communication
* Modified OSI Model for IOT/M2M

IOT

__/__/____

* INTERNET of THINGS

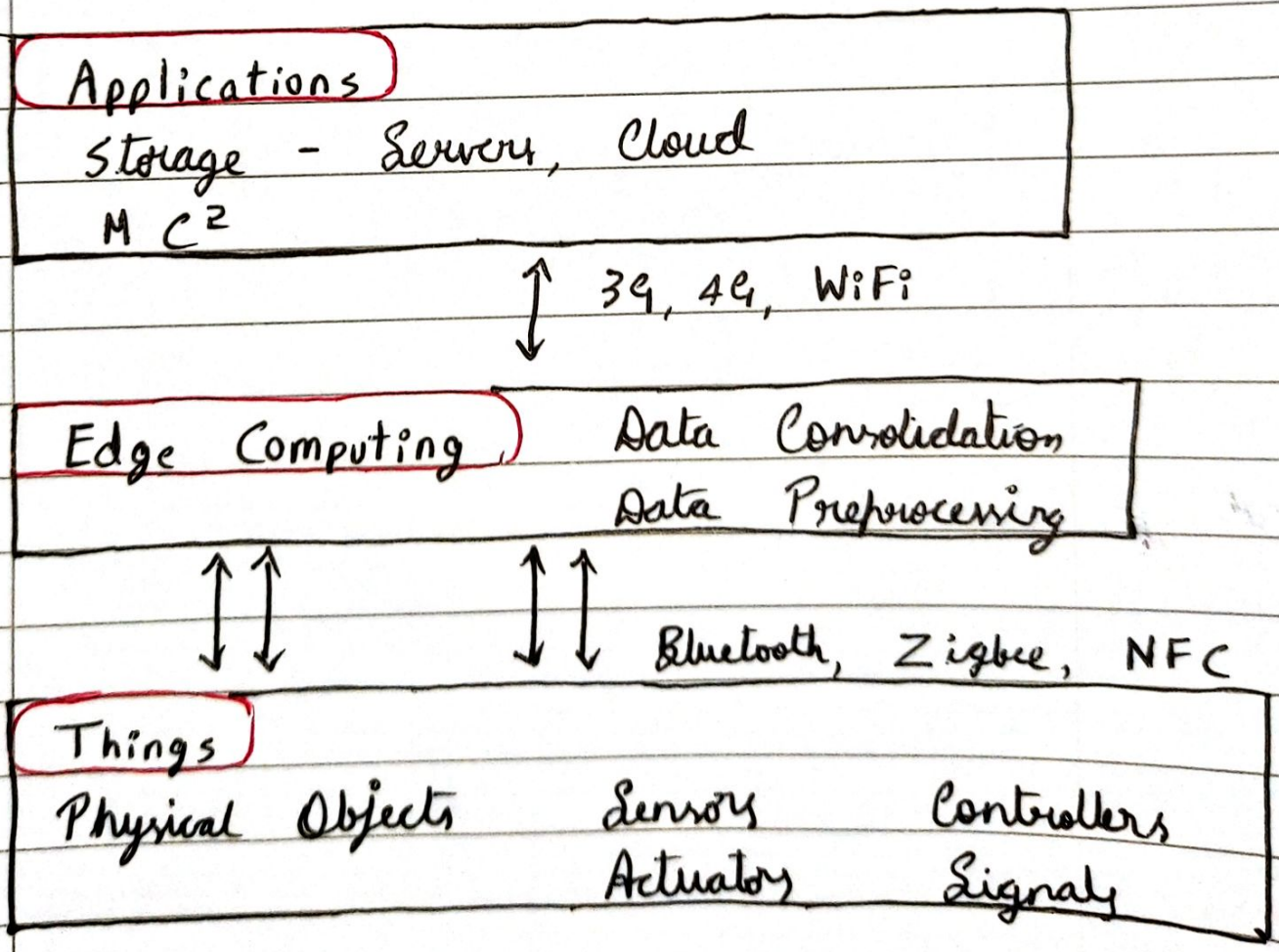
Internetwork of things, that enables things to communicate as if they are computing unit, so that they can be monitored, controlled, coordinated with other devices

* Vision of IoT

Smart

Hyperconnectivity

Things becoming smart / intelligent and behaving alive

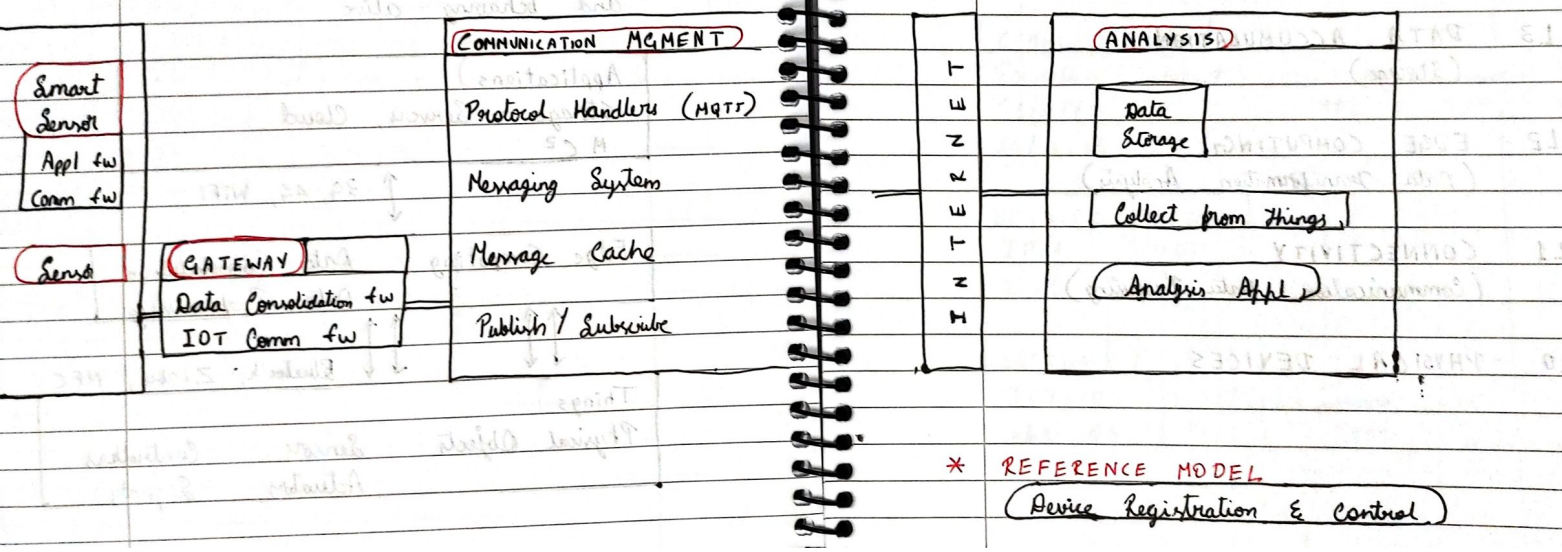


* CONCEPTUAL FRAMEWORK

– Physical Objects + S and A + Internet

(Oracle) – G E S M A OA

(IBM) – G Cs Cn C1 A MA



* REFERENCE MODEL

Device Registration & Control

__/__/____

* IOT Architecture View

L6 COLLABORATION & PROCESSES
(People and Business Process)

L5 APPLICATION
(Reporting, Analysis, Control)

L4 DATA ABSTRACTION
(Access) (raw data → useful info)

L3 DATA ACCUMULATION
(Storage)

L2 EDGE COMPUTING.
(Data Transformation, Analysis)

L1 CONNECTIVITY
(Communication, Data Processing)

L0 PHYSICAL DEVICES

__/__/____

* TECHNOLOGY behind IOT

HARDWARE

Arduino, Raspberry Pi, Intel Galileo, Intel Edison, ARM mbed

IDE for developing software, firmware, API

PROTOCOLS

CoAP, MQTT, RESTful

COMMUNICATION

Powerline Ethernet RFID Bluetooth WiFi
6 Low PAN NFC Zigbee WiMax
2g/3g/4g

NETWORK BACKBONE

IPv4 UDP
IPv6 6 low PAN

SOFTWARE

RIOT OS Things Square Mist
Contiki OS Eclipse IOT

CLOUD / Data Center

ML MODELS

__/__/____

* Major Components of IOT

- Physical Object
- Hardware [S, A, C]
- Comm Module
- Software

* SOURCES of IOT

__/__/____

* M2M COMMUNICATION
- M2M between different machines
- 3 domain architecture (device, network, appl)

Application Domain
RAC
M2M applications

Network Domain
M2M Server Device Identity Mgmt
Data - Abstraction, Accumulation, Analysis
Messaging System (Push Pull, Pub/Sub)
Unicast, Multicast communication
Connectivity (Comm and Processing units)

↕

Communication GATEWAY

↕

Device Domain

Connectivity Interface (Comm & PU)
Edge Computing (Data PP, T, A)
Physical Devices (SAC)

MODULE 2

* Constrained Application Protocol (COAP)
  [CoRE using ROLL]
* Lightweight M2M Communication Protocol
* MQTT Protocol [COAP Message Queue, Broker]
* Terminologies [Message Communication Protocols]
* Terminologies – Communication Protocol
  – API – URI – URL
  – Communication Gateway

IOT UNIT 2

1.Terms in web connectivity

Application or App refers to a software for applications such as creating and sending an SMS, measuring and sending the measured data, receiving a message from a specific sender etc. **Application Programming Interface (API)** refers to a software component, which receives messages from one end; for example, from an application or client or input. An API may consist of GUIs (button, check box, text box, dialog box).

Web service refers to a servicing software which uses web protocols, web objects or WebSockets; for example, weather reports service, traffic density reports, streetlights monitoring and controlling service.

Object refers to a collection of resources; for example, collection of data and methods (or functions or procedures) to operate on that data. . An object instance can be just one or more than one for an object. An example of an object instance is birth_date.

Communication gateway is one that functions as communication protocol translator (convertor) for provisioning communication capabilities. For example, the gateway for communication between ZigBee and IP networks.

Client refers to a software object which makes request (or an API associated with it makes request) for data, messages, resources or objects. A client can have one or more object instances. Server is defined as a software which sends a response on a request. The server also sends messages, alerts or notifications.

Broker denotes an object, which arranges the communication between two ends; for example, between the message publisher and subscriber or for example taking the request from a source and sending the response received back for that source after arranging the response from another object, such as a server.

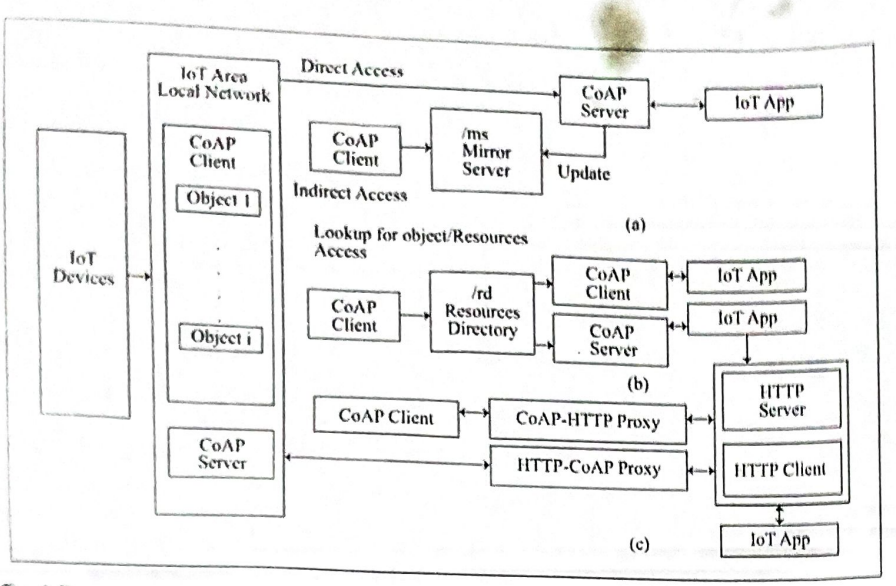
Proxy refers to an application which receives a response from the server for usage of a client or application and which also receives requests from the client for the responses retrieved or saved at proxy.

Communication protocol defines the rules and conventions for communication between networked devices and between systems. The protocol includes mechanisms for devices or systems to identify and make connections with each other.

Web protocol is a protocol that defines the rules and conventions for communication between the web server and web clients. It is a protocol for web connectivity of web objects, clients, servers and intermediate servers or firewalls.

2. COAP

Features of CoAP are: An IETF defined application-support layer protocol .CoAP web-objects communicate using request/response interaction model. A specialised web-transfer protocol which is used for CoRE using ROLL network. It uses object-model for the resources and each object can have single or multiple instances. Each resource can have single or multiple instances. An object or resource use CoAP, DTLS (security binding with PSK, RPK and Certificate) and UDP protocols for

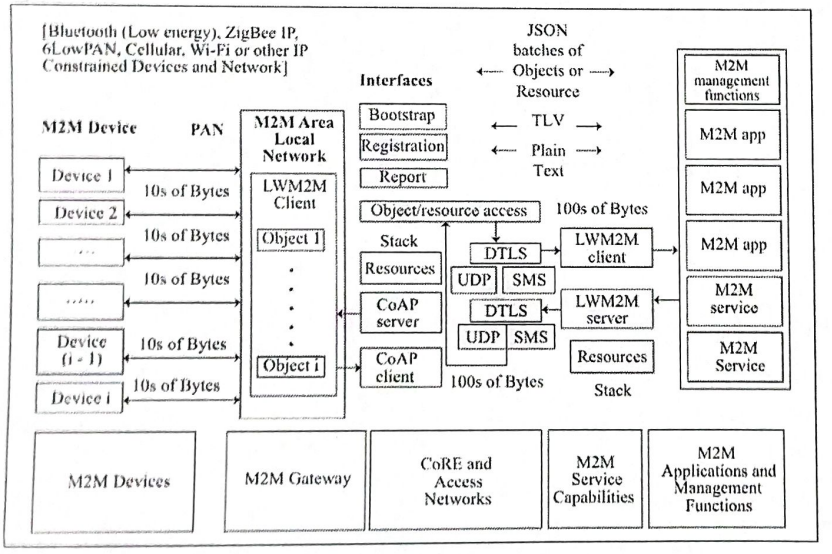


sending a request or response. Supports the resource directory and resource-discovery functions.CoRE communication is asynchronous communication over the ROLL. Integrates easily with the web using CoAP application cross-protocol proxies.

A proxy is an intermediate server, which accepts a request from a client and sends the response to the client using a protocol. HTTP-CoAP proxy accepts requests from HTTP client using HTTP protocol and sends the request to the server using CoAP protocol. CoAP-HTTP proxy accepts requests from CoAP client using CoAP protocol and sends the request to the server using HTTP protocol. Transport Layer Security (TLS), earlier known as Secure Socket Layer (SSL) is the protocol used for securing the TCP-based Internet data interchanges. DTLS is the TLS for datagram. The features of DTLS are: DTLS provisions for three types of security services—integrity, authentication and confidentiality. DTLS protocol derives from TLS protocol and binds UDP for secured datagram transport.

Secured Use of a Key for Client Authentication : PSK stands for Pre-Shared Key and is a method of securing using a key to authenticate a client. The key contains up to 133 characters in English. PSK method generates a unique encryption key for each client. A PSK is a symmetric key without forward secrecy (sender key not secret from receiver). RPK stands for Random Pair-wise Keys, which also stands for Raw Public Key meaning only the private/public key [means end 1 using RPK uses (K1 and Kp) and the other end using RPK uses (K2 and Kp)] are asymmetric.

3. Lightweight M2M communication protocol

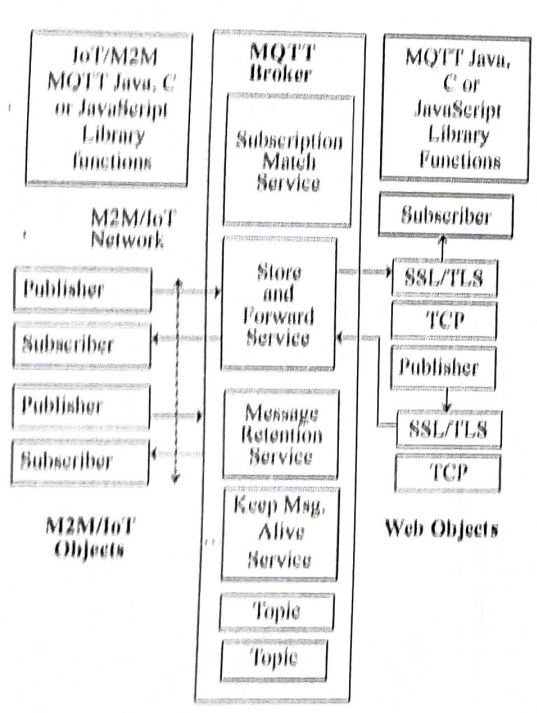


Lightweight Machine-to-Machine Communication (LWM2M) protocol is an application layer protocol specified by Open Mobile Alliance (OMA) for transfer of service data/messages. It finds applications in M2M. It enables functionalities for device management in cellular or sensor networks. Lightweight presently means data transfer formats between client and server are binary and has Tag Length Value (TLV) or Java Script Object Notation (JSON) batches of object arrays or resource arrays and transfers up to 100s of bytes unlike the webpages of 1000s of bytes. The protocol enables communication between LWM2M client at IoT device and an LWM2M server at the M2M application and service capability layer. The protocol is a compact one, meaning small header.

Assume I number of M2M devices. Figure 3.3 shows the following: Local M2M constrained devices use, for example, the Bluetooth low energy, 6LowPAN (IPv6 over low power WPAN), CoRE, ROLL, NFC, ZigBee PAN, cellular, Wi-Fi or ZigBee IP (left-hand side) network technologies. M2M area network functions as PAN for connectivity between devices and the M2M gateway. 10s of bytes communicate between a device and the PAN. Communication between LWM2M objects (right-hand side). LWM2M client refers to object instances as per the OMA standard LWM2M protocol. A client object sends a request or receives a response of the LWM2M server over the access and CoRE networks. CoRE network, for example, 3GPP or other networks for the IP connectivity Communication from an object instance using interface functions. The functions are bootstrapping; registration, deregister or update a client and its objects; reporting the notifications with new resource values; and service and management access through the server. Use of the CoAP, DTLS, and UDP protocols by the object or resource. 100s of bytes communicate between objects at the client or server for plain text or JSON or binary TLV format data transfer.

LWM2M specifications and features are as follows: • 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 or binary TLV format data transfer for a package for a batch of resource representations in a single data transfer. An object or its resource access using an URI.

4. MQTT protocol



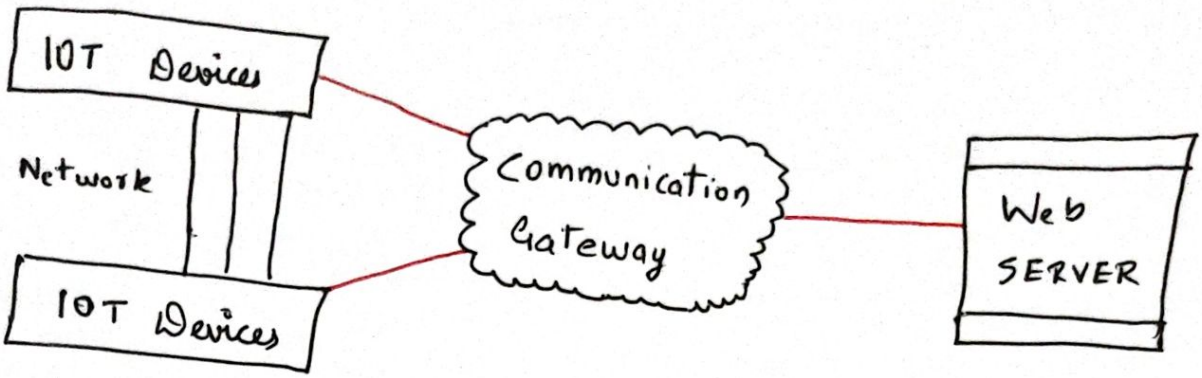
Message Queuing Telemetry Transport (MQTT) is an open-source protocol for machine-to machine (M2M)/IoT connectivity.

. A version is MQTT v3.1.1. MQTT has been accepted (2014) as OASIS (Organization for the Advancement of Structured Information Standards) standard 6 MQTT protocol is used for connectivity in M2M/IoT communication. A version is MQTT-SN v1.2. Sensor networks and non-TCP/IP networks, such as ZigBee can use the MQTT-SN. MQTT-SN is also a publish/subscribe messaging protocol. It enables extension of the MQTT protocol for WSNs, the sensor and actuator devices and their networks.

Figure shows MQTT-broker subscription, subscription match, store and forward, last good message retention and keep message alive services. The figure also shows that device objects use MQTT Java, C or JavaScript library functions. The objects communicate using the connected devices network protocols such as ZigBee.

MQTT Broker (also referred simply as MQ) does the following: Functions as a server node capable of storing messages from publishers and forwarding them to the subscribing clients. Receives topics from the publishers. Examples of topics are measured information of ambient light conditions, traffic density, nearby parking space availability and waste container status. Performs a store-and-forward function, stores topics from publishers and forwards to subscribers. Receives subscriptions from clients on the topics, matches subscriptions and publications in order to route messages to right

KEY TERMS - Web Connectivity



---

* **Application** : Software
* **API** : Software Component

* sends/receives data
* Form, Button

* **Web Service** : Software that uses web protocols
* **Object** : Collection of Resources (↔ R, W, E)
* **Class** : Blueprint for Object
* **Communication Gateway**

Communication Protocol Translator

* **Communication Protocol**

  Defines Rules for communication b/w two networked devices

URI	URL
Universal Resource Identifier	Universal Resource Locator
It is a string	Type of URI
/Contacts/School/	http://www.wikipedia.com/RAM
Identifies a resource	Identifies a resource
document, service	and
Doesn't provide means to access it	Provides means to access it (protocol: HTTP, FTP, ...)

* CoRE

- Constrained RESTful Environment
- It is an environment that supports communication b/w IOT devices within LAN

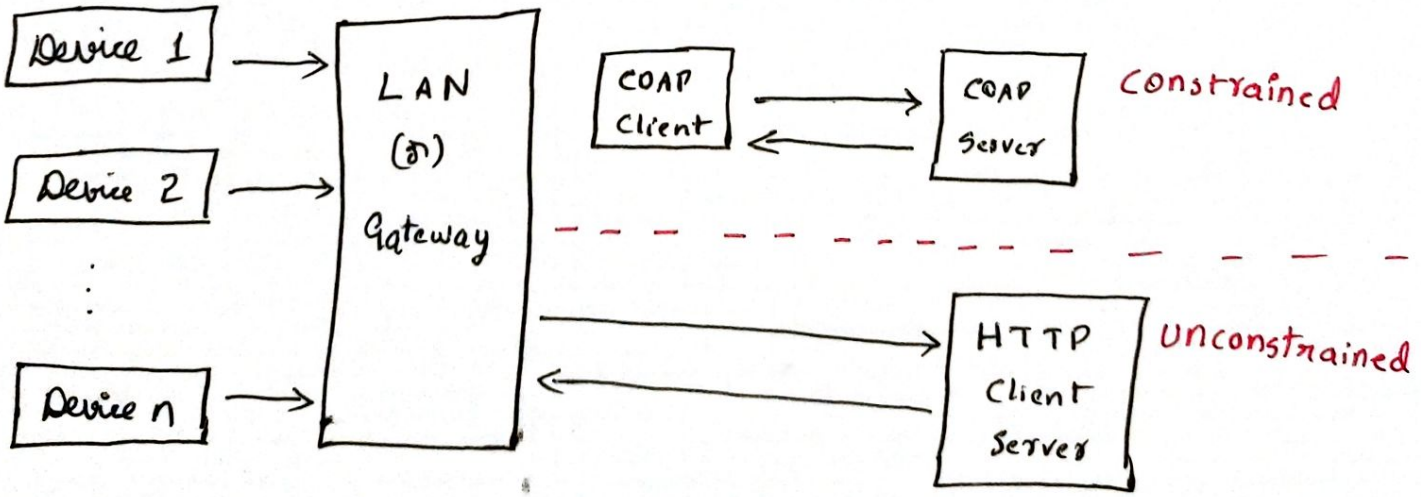
* Characteristics

- Low Data Volume
- Uses REST architecture (Representational State Transfer)
- Low Power Devices
- Uses ROLL (Routing Over low power & lossy Networks)

---

* Unconstrained Environment

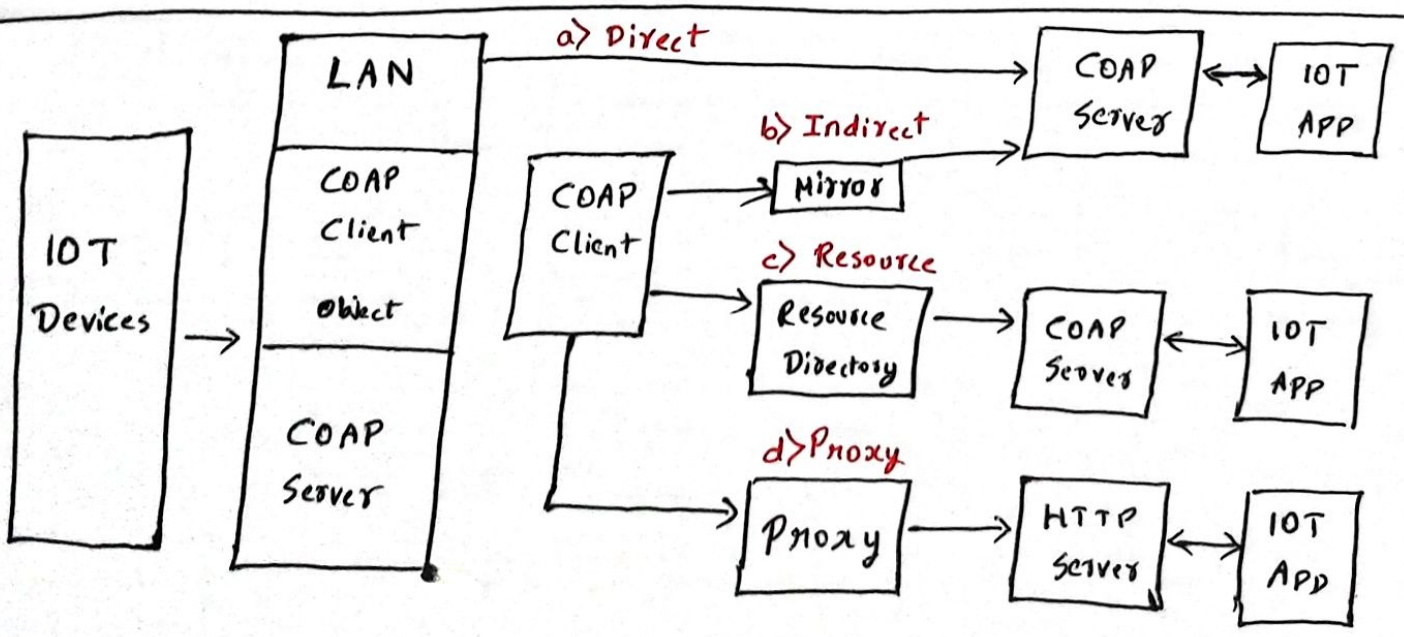
- Web Apps use HTTP b/w Client and Server



* COAP [Constrained Application Protocol]

- Application - Support Layer Protocol
- Defined by IETF
- Specialized Web Transfer Protocol for constrained env

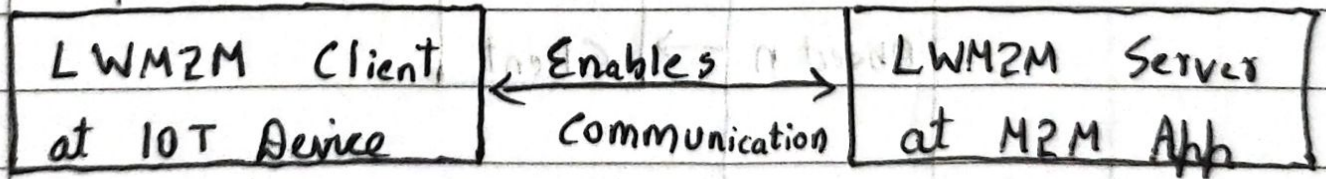
* Request / Response model : Compatible with REST architecture
* Optimized for CoRE (particularly ROLL network)
* Object - Model for Resources (coap://sensor/rain)
* Security → DTLS, PSK (Pre-Shared Keys)
  RPK (Raw Public Keys)
  Certificate based Security
* Small Message Header (4 bytes)
* Asynchronous Communication
* Resource Discovery and Directory



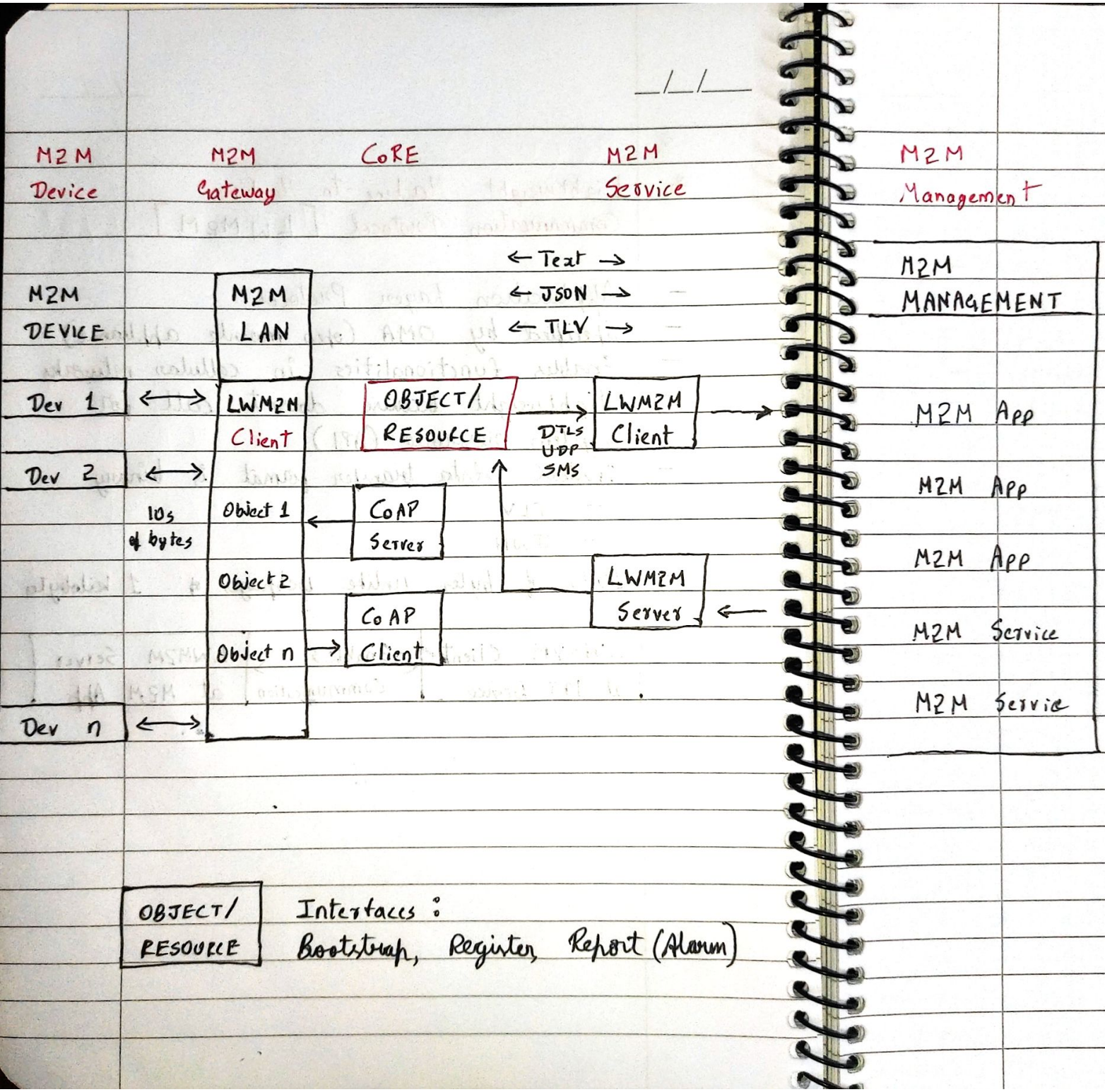
__/__/____

* **Lightweight Machine-to-Machine Communication Protocol [LWM2M]**

- Application layer Protocol.
- Specified by OMA (open mobile appliance)
- Enables functionalities in cellular networks
- Lightweight because doesn't call for system resource (API)
- Instead, data transfer format is binary
  - TLV
  - JSON
- 100s of bytes unlike webpages of 1 kilobytes



__/__/____


M2M Device
M2M Gateway
CoRE
M2M Service
M2M Management
← Text →
← JSON →
← TLV →
M2M DEVICE
M2M LAN
M2M MANAGEMENT
Dev 1
Dev 2
Dev n
LWM2M Client
OBJECT/ RESOURCE
DTLS UDP SMS
LWM2M Client
M2M App
M2M App
M2M App
M2M Service
M2M Service
10s of bytes
Object 1
Object 2
Object n
CoAP Server
CoAP Client
LWM2M Server
OBJECT/ RESOURCE
Interfaces :
Bootstrap, Register, Report (Alarm)

__/__/____

* Formats

1) JSON (JavaScript Object Notation)

- Open standard format
- Used to transmit data between server and web application
- Human readable
- Attribute - Value pairs

```
{
"0" : { " :id : 0,
        "name" : "myname",
        "mandatory" : true,
        "description" : "..... \n"
      },
"3" : {  -
         -
      },
}
```

__/__/____

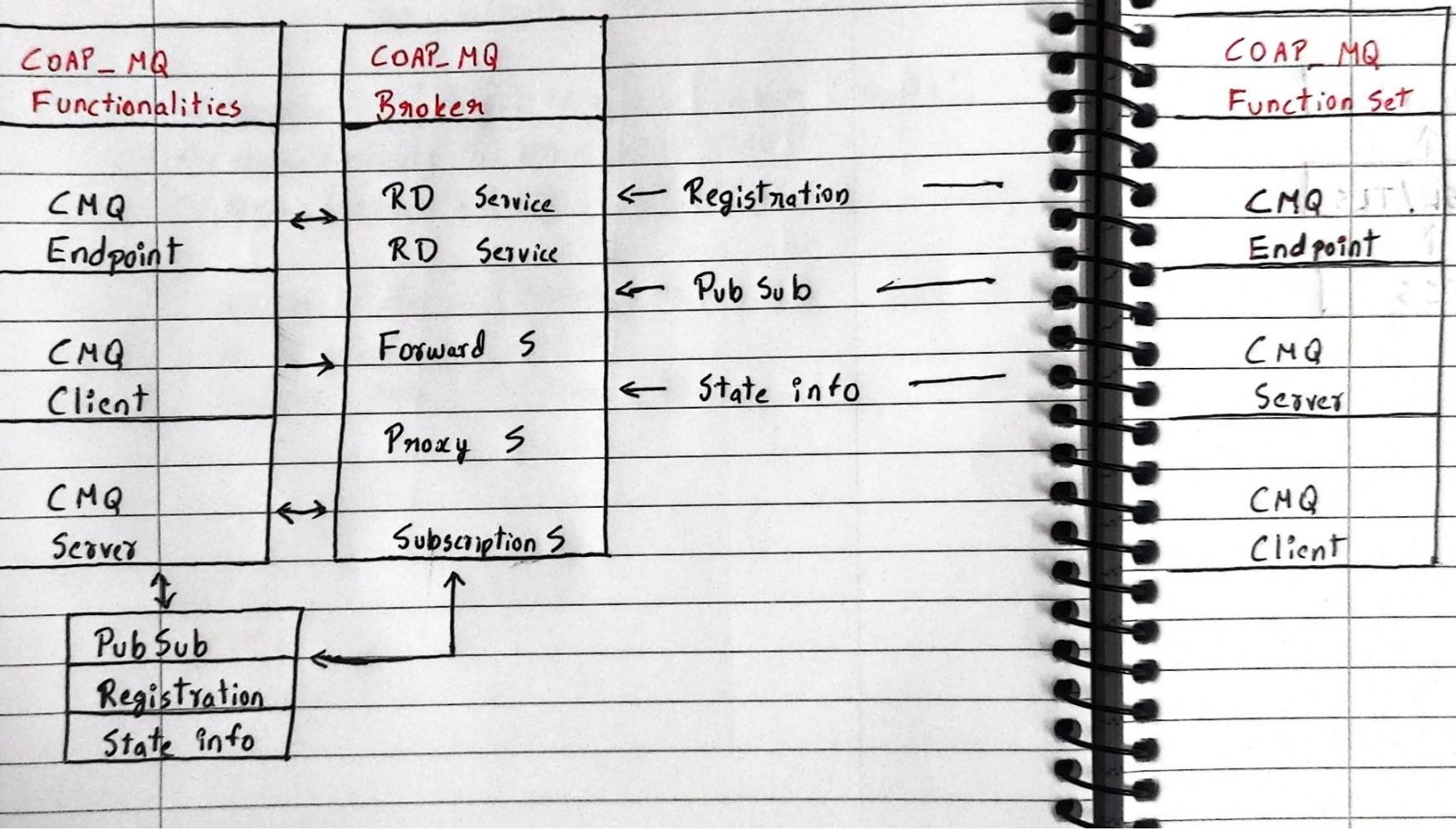
2> TLV (Tag Length Value)

</engine> ; Obj ; id = "1" ; name = "Toyota" ;
</engine/velocity> ; Obj ; id = "260" ,
</engine/rpm> ; Obj ; id = "70"

Tag	Type of Data (Name, Age, ...)
Length	No. of Bytes (4)
Value	Actual Data (John, 30 year)

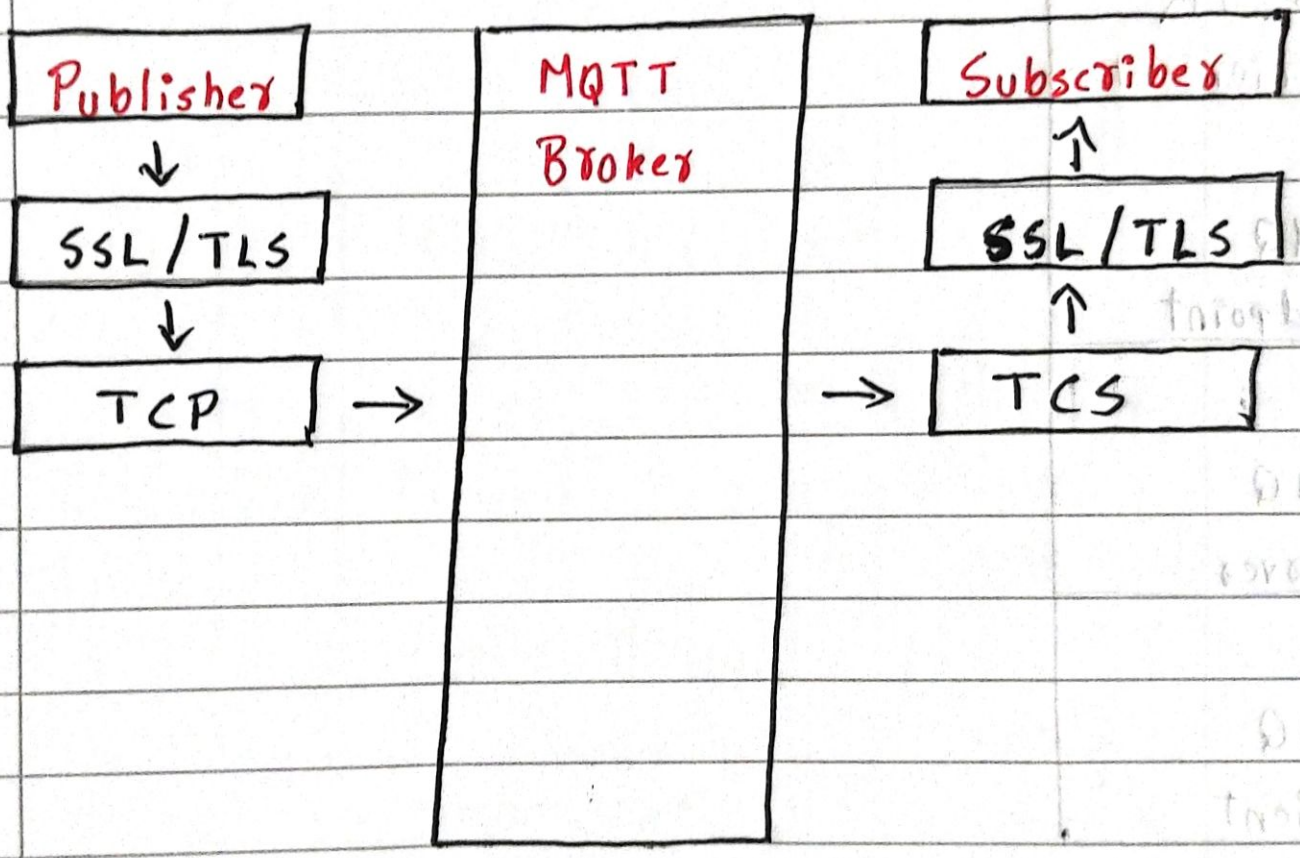
__/__/____

* COAP_MQ


COAP_MQ Functionalities
CMQ Endpoint
CMQ Client
CMQ Server
COAP_MQ Broker
RD Service
RD Service
Forward S
Proxy S
Subscription S
Registration
Pub Sub
State info
COAP_MQ Function Set
CMQ Endpoint
CMQ Server
CMQ Client
Pub Sub
Registration
State info

__/__/__

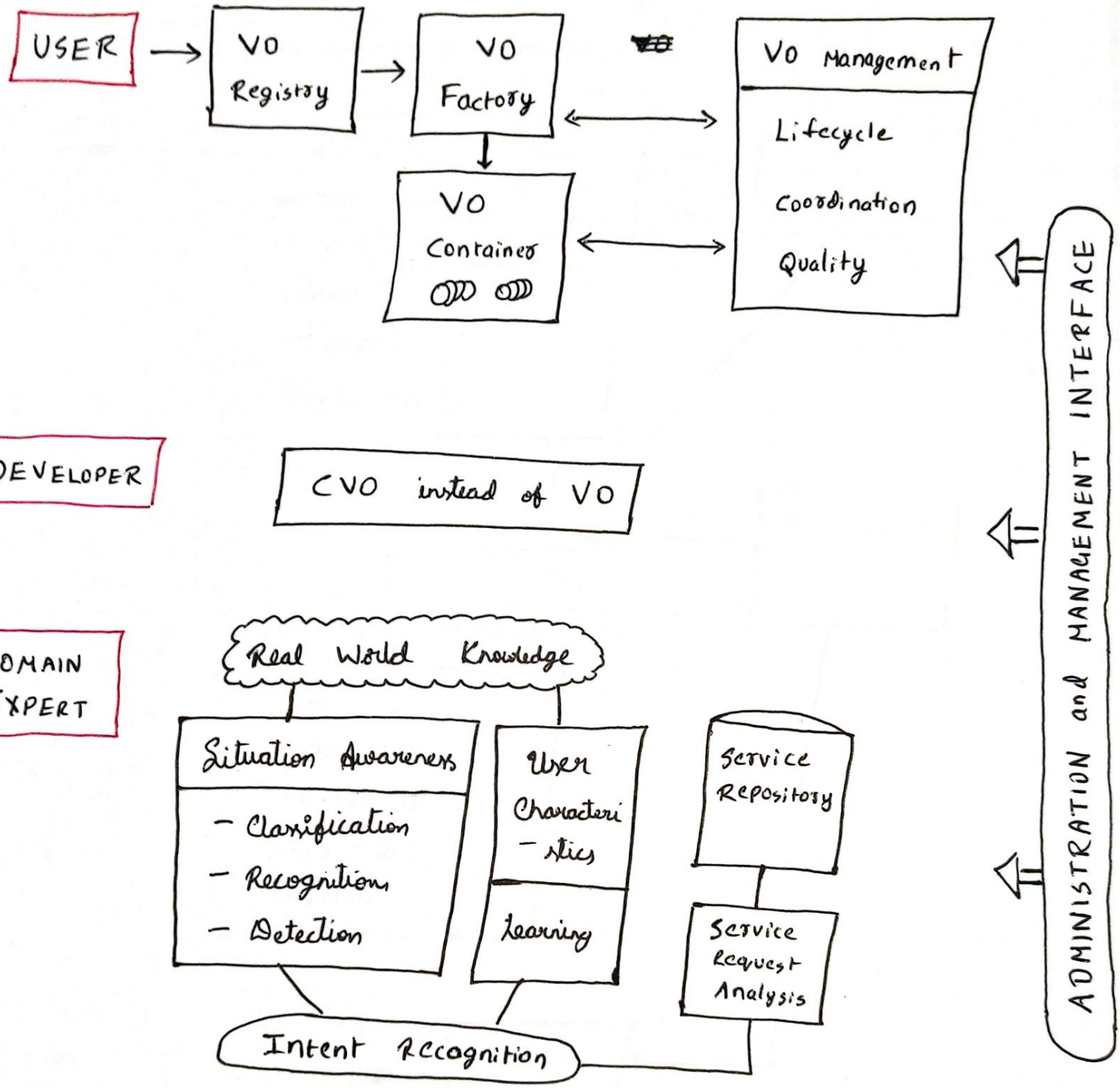
* MQTT



UNIT 3

* FP7 iCore Access Framework
* IOT @ Work Capability based Access Control System
* SMARTIE → Dependability, Security, Privacy features
* Smart City → Security, Privacy, Trust
* - Smart Transportation
  - Smart Campus

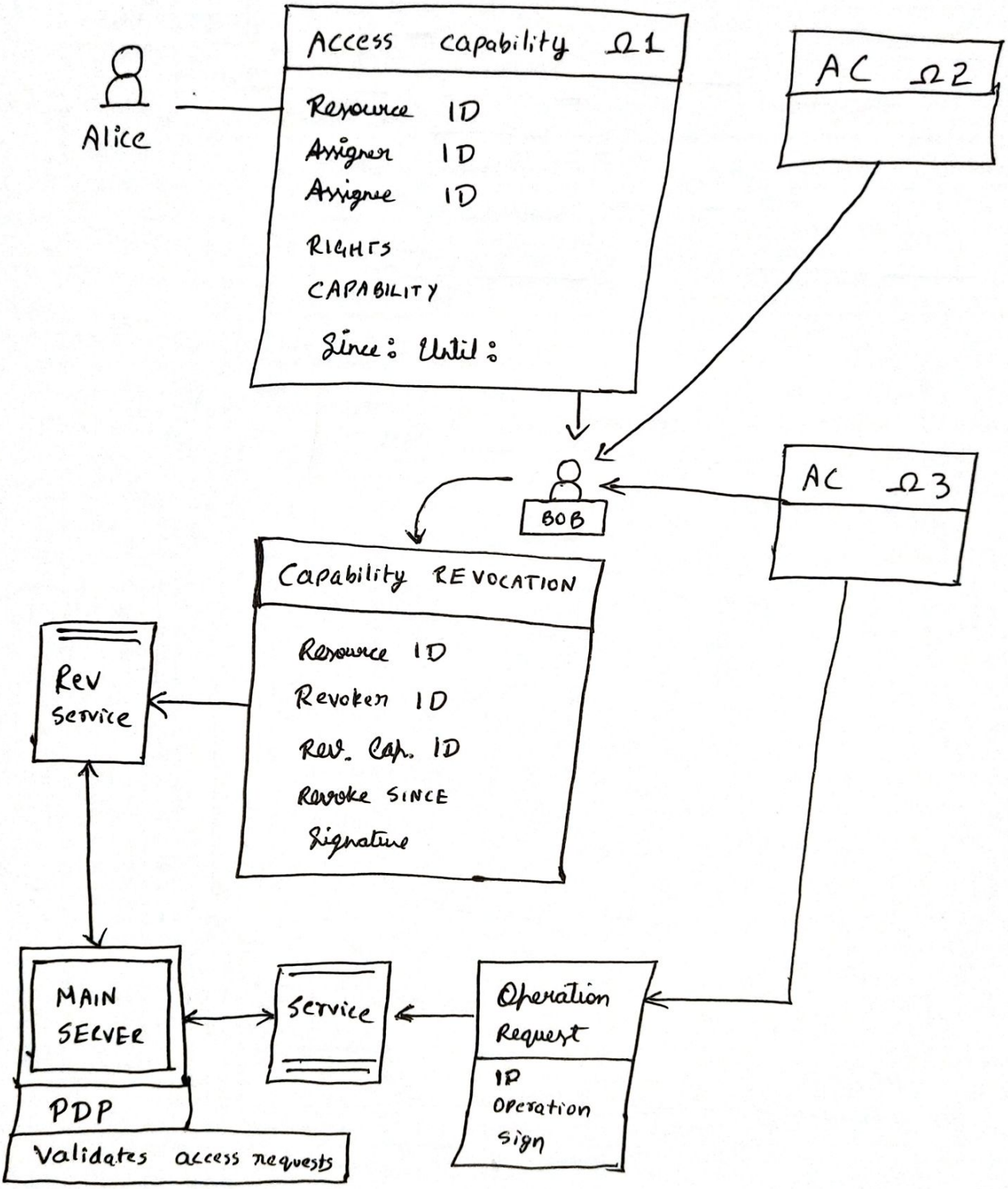
* FP7 iCore Access Framework



---

* Cognitive Approach to IOT
* Thing → VO → CVO

* IOT @ Work Capability based Access Control System



Capability :→ Communicable, Unforgeable } token of authority
- rights
- subjects
- objects

User → tokens → Service Provider (Delegation) (Revocation)

* Overview of Activity Chain 05

- IERC has established no. of activity chains

i) - To foster collaboration

ii) - Platform to exchange ideas on IOT-related challenges

Activity Chain 05

Significant chains

Focussed on → Privacy, Security, Governance

Privacy: users

Security: systems

Governance: structures to ensure trust

* CIOT Challenge

BLURS

1) Bandwidth

2) Latency

3) Uninterrupted

4) Resource - constrained devices

5) Security

UNIT 4

* FEC - Advantages & Features

[Complement to Cloud in IoT]

SCALE

1) Security : IOT devices (FEC node) cloud

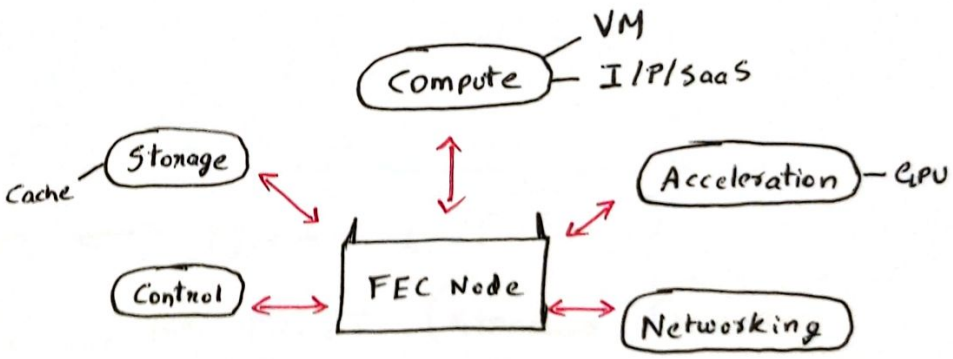
2) Cognition : decision-making, (S-O, S-A, S-H)

3) Agility : SDKs & OSS interfaces, Small buriness can contribute

4) Latency : real time response

5) Efficiency : Optimizes performances, reduce cost

HOW? SCANC



1) Storage - Cache

2) Compute - VMs, Containers, I/P/SaaS

3) Acceleration - GPU, FPGA

4) Networking - IP based, Zigbee/Bluetooth

5) Control - D, A, M, S

* ACCELERATION

1) Networking Acceleration

→ Network Virtualization : operates multiple routes

→ SDN (software defined network)

Custom routing path for application

2) Computing Acceleration

→ GPU, ~~EPGPA~~ FPGA

↓ faster processing

↳ customize hardware

* NETWORKING

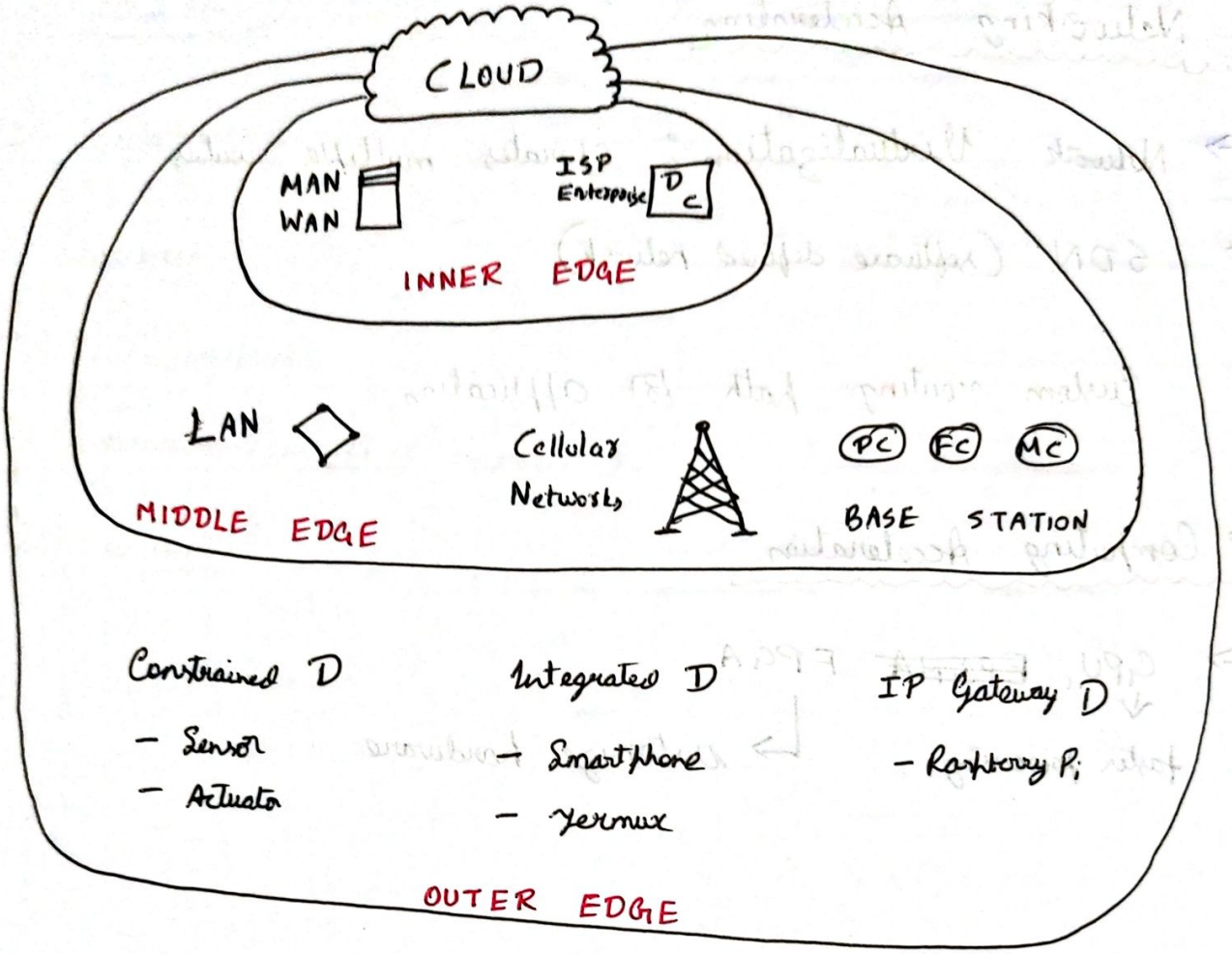
1) Vertical

- Direct IOT Device ⟷ Cloud (IP based)
- TCP/UDP, HTTP, ... CoAP, MQTT (Standard Protocols)
- FEC nodes are Proxies

2) Horizontal

- Device to Device comm
- Heterogeneous Protocols
- Bluetooth, Zigbee

* Hierarchy of Fog and Edge



* Business Model of FEC

1) XaaS
2) Support Service (H, M, TS, C)
3) Application Service

* Opportunities

1) OOBE
   $7000m
   equipment, software
2) Open Platforms
3) System Mgmt

* Challenges

→ Standardization
→ Compatibility issues
→ Complexity