Module 1:

Overview of Internet of Things:

Syllabus:

IoT Conceptual Framework, IoT Architectural View, Technology Behind IoT, Sources of IoT,M2M communication, Examples of IoT. Modified OSI Model for the IoT/M2M Systems, data enrichment, data consolidation and device management at IoT/M2M Gateway, web communication protocols used by connected IoT/M2M devices, Message communication protocols (CoAP-SMS, CoAPMQ, MQTT,XMPP) for IoT/M2M devices.

Definition:

The **Internet of Things (IoT)** is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

OR

The **Internet of things** refers to a type of network to connect anything with the Internet based on stipulated protocols through information sensing equipments to conduct information exchange and communications in order to achieve smart recognitions, positioning, tracing, monitoring, and administration.

Characteristics of IoT:

The fundamental characteristics of IoT are as follows:

- **Interconnectivity:** In IoT, anything can be interconnected with the global information and communication infrastructure.
- **Things-related services:** The IoT is capable of providing thing-related services within the constraints, such as privacy protection and semantic consistency between physical things and their associated virtual things. In order to provide thing-related services within the constraints of things, both the technologies in physical world and information world will change.
- **Heterogeneity:** The devices in the IoT are heterogeneous as based on different hardware platforms and networks. They can interact with other devices or service platforms through different networks.
- **Dynamic changes:** The state of devices change dynamically, e.g., sleeping and waking up, connected and/or disconnected as well as the

context of devices including location and speed. Moreover, the number of devices can change dynamically.

- **Enormous scale:** The number of devices that need to be managed and that communicate with each other will be at least an order of magnitude larger than the devices connected to the current Internet
- **Safety:** As benefits are more with the usage, safety becomes an ad joint issue. This includes the safety of personal data and the safety of physical well-being. Securing the endpoints, the networks, and the data moving across all of it means creating a security paradigm that will scale.
- **Connectivity:** Connectivity enables network accessibility and compatibility. Accessibility is getting on a network while compatibility provides the common ability to consume and produce data.

Basic IoT Architectural View [only for understanding purpose but forms the base for the next topic]:

IOT architecture consists of different layers of technologies supporting IOT.

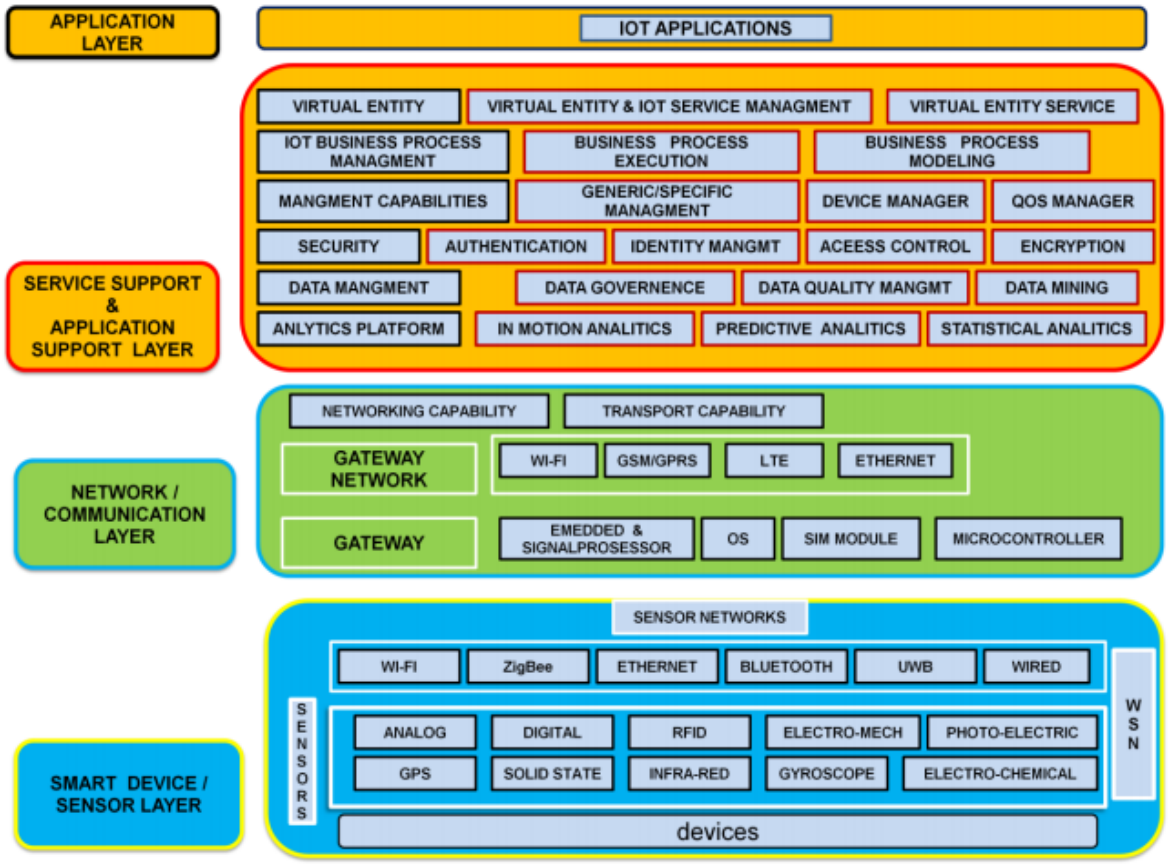


Figure1: Architectural view of IoT

1. **Smart device / sensor layer:**
   - ❖ The lowest layer is made up of smart objects integrated with sensors.

- ❖ The sensors enable the interconnection of the physical and digital worlds allowing real-time information to be collected and processed.
- ❖ There are various types of sensors for different purposes.
- ❖ The sensors have the capacity to take measurements such as temperature, air quality, speed, humidity, pressure, flow, movement and electricity etc. In some cases, they may also have a degree of memory, enabling them to record a certain number of measurements.
- ❖ A sensor can measure the physical property and convert it into signal that can be understood by an instrument.
- ❖ Sensors are grouped according to their unique purpose such as environmental sensors, body sensors, home appliance sensors and vehicle telemetric sensors, etc.
- ❖ Most sensors require connectivity to the sensor gateways. This can be in the form of a Local Area Network (LAN) such as Ethernet and Wi-Fi connections or Personal Area Network (PAN) such as Zigbee, Bluetooth and Ultra Wideband (UWB).
- ❖ Sensors that use low power and low data rate connectivity, they typically form networks commonly known as wireless sensor networks (WSNs).

2. **Gateways and Networks:**
   - ❖ Massive volume of data will be produced by tiny sensors.
   - ❖ It requires a robust and high performance wired or wireless network infrastructure as a transport medium.
   - ❖ Current networks, often tied with very different protocols, have been used to support machine-to-machine (M2M) networks and their applications.
   - ❖ With demand needed to serve a wider range of IOT services and applications such as high speed transactional services, context-aware applications, etc, multiple networks with various technologies and access protocols are needed to work with each other in a heterogeneous configuration.
   - ❖ These networks can be in the form of a private, public or hybrid models and are built to support the communication requirements for latency, bandwidth or security.
   - ❖ Various gateways (microcontroller, microprocessor…) & gateway networks (WI-FI, GSM, GPRS…) are shown in figure 1
3. **Management Service Layer**:
   - ❖ The management service renders the processing of information possible through analytics, security controls, process modeling and management of devices.
   - ❖ One of the important features of the management service layer is the business and process rule engines.

- ❖ IOT brings connection and interaction of objects and systems together providing information in the form of events or contextual data such as temperature of goods, current location and traffic data.
- ❖ Some of these events require filtering or routing to post processing systems such as capturing of periodic sensory data, while others require response to the immediate situations such as reacting to emergencies on patient's health conditions.
- ❖ The rule engines support the formulation of decision logics and trigger interactive and automated processes to enable a more responsive IOT system. In the area of analytics, various analytics tools are used to extract relevant information from massive amount of raw data and to be processed at a much faster rate.

4. **Application Layer**:

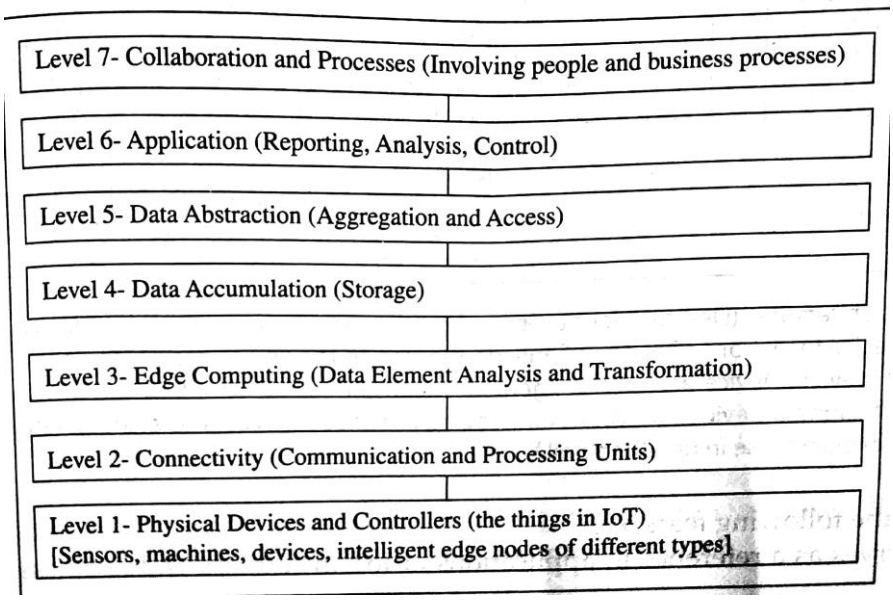
   The IoT application covers "smart" environments/spaces in domains such as: Transportation, Building, City, Lifestyle, Retail, Agriculture, Factory, Supply chain, Emergency, Healthcare, User interaction, Culture and tourism, Environment and Energy.

IoT Architectural View: [As per VTU Syllabus]

The IoT system is defined in different levels called as tiers. A model enables the conceptualisation of the framework.

A reference model can be used to depict the building blocks, successive interactions and integration.

The diagram below depicts the CISCO presentation of a reference model comprising of 7 levels and the functions of each level.



Features of the architecture:

- The architecture serves as a reference in the applications of IoT in services and business processes.
- A set of sensors which are smart, capture the data, perform necessary data element analysis and transformation as per device application framework and connect directly to a communication manager.
- The communication management subsystem consists of protocol handlers, message routers and access management.
- Data routes from gateway through the Internet and data centre to the application server or enterprise server which acquires that data.
- Organisation and analysis subsystems enable the services, business processes, enterprise integration and complex processes.

IEEE P2413

IEEE suggested P2413 standard for architecture of IoT. It is a reference architecture which builds upon the reference models. This reference model defines the relationship between various IoT Applications like Transportation and Health Care.
The characteristics of this IEEE standard are as follows:

- ❖ Follows top- down approach.
- ❖ Does not define a new architecture but reinvent existing architectures congruent with it
- ❖ Gives a blue print for data abstraction.
- ❖ Specifies abstract IoT domain for various IoT domains.
- ❖ Recommends quality 'quadruple' trust that includes protection, security, privacy and safety.
- ❖ Addresses the documentation of data.
- ❖ Strives for mitigating architecture divergence.

IoT Conceptual Frame Work:

Explain the concept of operation in an IoT System.

Explain the Oracle Conceptual Frame work of IoT

Explain the IBM Conceptual Frame work of IoT

An IoT System has multiple levels as seen in the basic architecture. It can be explained using the equations given below:

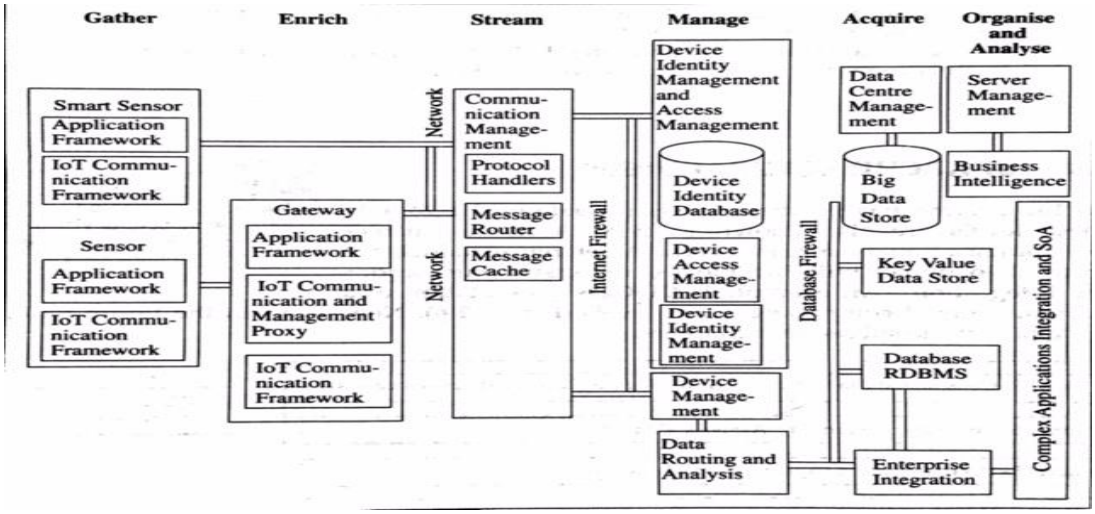
Physical Object+ Controller, Sensor & Actuators+ Internet= IoT--------- (1)

IoT is an internetwork of devices and physical objects. The operation of these devices is could be to gather the information or acquire the parameter through a sensor or a controller and an actuator to serve the application.

Ex: A series of street lights communication data to the group controller which connects t the central server using the **Internet.**

Gather + Enrich + Stream + Manage + Acquire + Organise& Analyse =IoT with Connectivity to Data enter, Enterprise or Cloud ---------- (2)

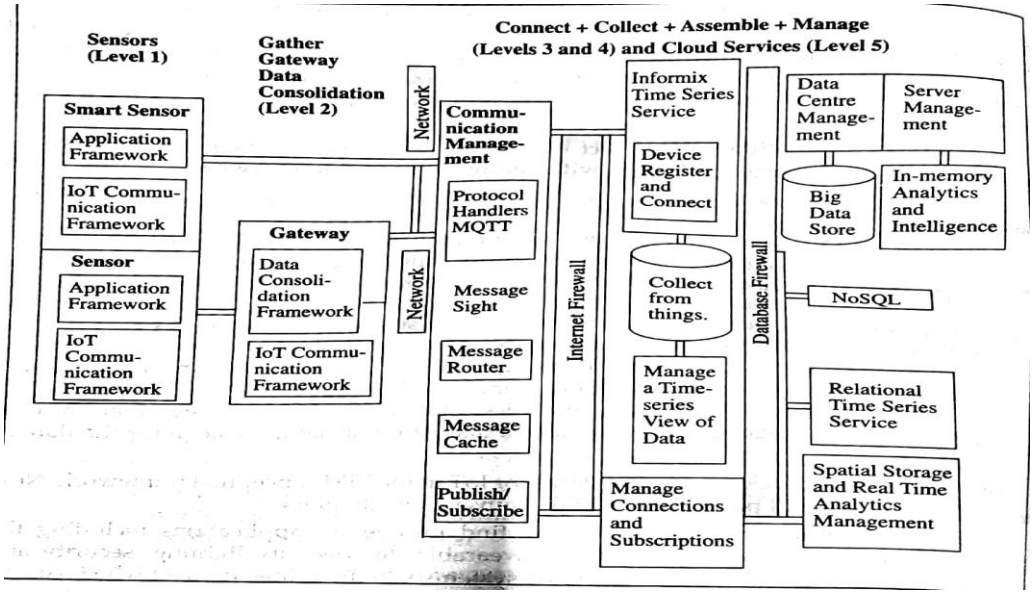
The equation (2) represents the **conceptual frame work and architecture** presented by **Oracle** as in the figure below.
The steps and processes that this architecture follows to communicate the data at different levels in IoT are:



1. Level 1: The data of the devices (things) using sensors are gathered from Internet.
2. Level 2: A sensor connected to the Gateway functions as a smart sensor. The data is then enriched-transcoding at the gateway.
3. Level 3: A communication management subsystem sends and receives the data streams.
4. Level 4: The device management, identity management and access management subsystems receive the device's data.
5. Level 5: The data store or database acquires the data.
6. Level 6: Data routed from the devices and things is organised and analysed.

Gather + Consolidate + Connect + Assemble + Manage& Analyse = IoT With Connectivity to Cloud Services ---------- (3)

The equation (3) presents an alternate conceptual approach for a complex system proposed by ***IBM***. The framework is as shown below.



The steps for the actions and communication of data at the successive levels of IoT are as given below:

1. Levels 1 and 2 consist of a sensor network to gather and consolidate the data.
2. The gateway at level2 communicates the data streams between level 2 and 3. The system uses a communication management subsystem at level 3.
3. An information service consists of connect, collect, assemble and manage subsystems at levels 3 and 4.
4. Real time series analysis, data analytics and intelligence subsystems are at level 4 & 5. A cloud infrastructure, a data store or database acquires the data at level 5.

Various conceptual frameworks for IoT find number of applications. Ex: M2M communications, wearable devices, smart objects, smart automation of the house etc…

Smart systems use the user interfaces (UIs), Application Programming Interfaces(APIs), identification data, sensor data and communication ports to process the data and communicate it to the next level.

Technology behind IoT:

The following entities provide a diverse technology environment and are examples of technologies involved in IoT:

- Hardware: A variety of Hardware play a vital role in communicating the parameters from the IoT to the Publisher or Subscriber.
- The hardware to communicate requires an Integrated Development Environment (IDE) for developing device software, firm ware and APIs.

- Protocols are a means to effectively put the data into format. Ex: RPL, CoAP, Restful, HTTP, MQTT, XMPP.
- Communication: Media of information transfer- Power line Ethernet,RFID,NFC,6LowPAN,UWB,ZigBee,Bluetooth,WiFi,WiMAX,2G3G /4G.
- Network Backbone: IPV4, IPV6, UDP and 6LowPAN.
- Software: RIOT OS, Contiki OS, Thingsquare, Mist Firm ware, Eclipse IoT
- Internetwork Cloud platforms/ Data Centre: Sense, ThingsWorx, Nimbits
- Machine Learning Algorithm and Software

Server End Technology:

IoT Servers are application servers, enterprise servers, cloud servers, data centres and databases.

Servers offer the following components:

1. Online Platforms
2. Devices identification, identity management and their access management.
3. Data accruing aggregation, integration, organising and analysing
4. Use of web applications, services and business process.

Major Components of IoT Systems:

Major Components of IoT devices are as follows:

1. **Physical Object with embedded software into hardware** - Sensors and control units. Sensors are electronic devices that sense the physical environment. Control units commonly are the microcontroller units or a custom chip that can comprise of a processor, memory and several units which are interfaced together.
2. **Hardware** consisting of a microcontroller, firmware, sensors, control unit, actuators and communication module.
3. **Communication module**: Software consisting of device APIs and device interface for middleware for creating communication stacks using CoAP, LWM2M, IPV4, IPV6 and other protocols.
4. **Software** for actions on messages, information and commands which the devices receive and then output to the actuators, which enable actions such as glowing LEDs, robotic hand movement.

Sources of IoT:

Arduino Boards

- E.g. Arduino Yún
- Using Microcontroller ATmega32u4

- Includes Wi-Fi, Ethernet, USB port, micro-SD card slot and three reset buttons
- Runs Linux

Intel Galileo board

- A line of Arduino-certified development boards.
- Intel x86, Intel SOC X1000 Quark based System-On-Chip
- Power over Ethernet (PoE) and 6 Analog Inputs

Beagle Board

- Very low power requirement
- Card like computer, Can run Android and Linux
- Open source Hardware designs and the software for the IoT devices are

Raspberry Pi

- Wi-Fi-connected device
- Included code open source RasWIK

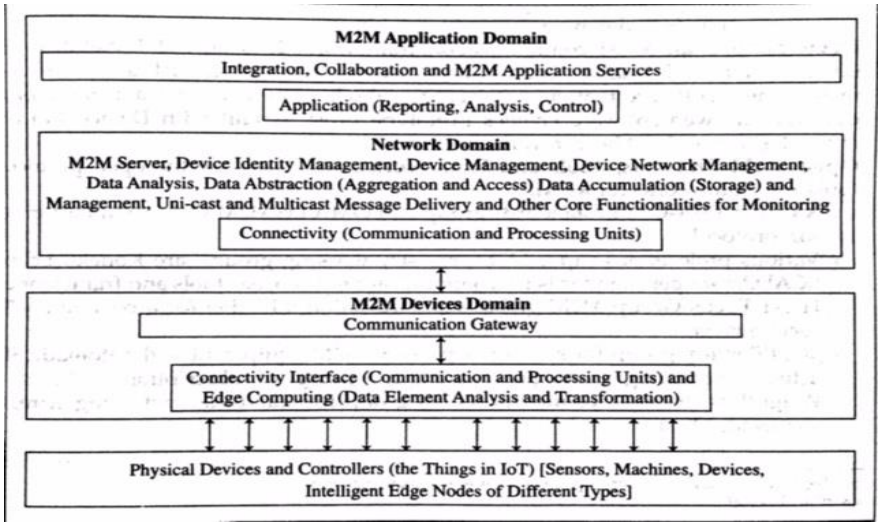
M2M Communication:

M2M refers to a process of communication of a physical object or device at machine with others of same type, mostly for monitoring and control purposes.

M2M to IoT:

- Technology closely relates to IoT which use smart devices to collect data that is transmitted via the Internet to other devices.
- Close differences lies in M2M uses for device to device communication also for coordinated monitoring and control purposes.

M2M Architecture:



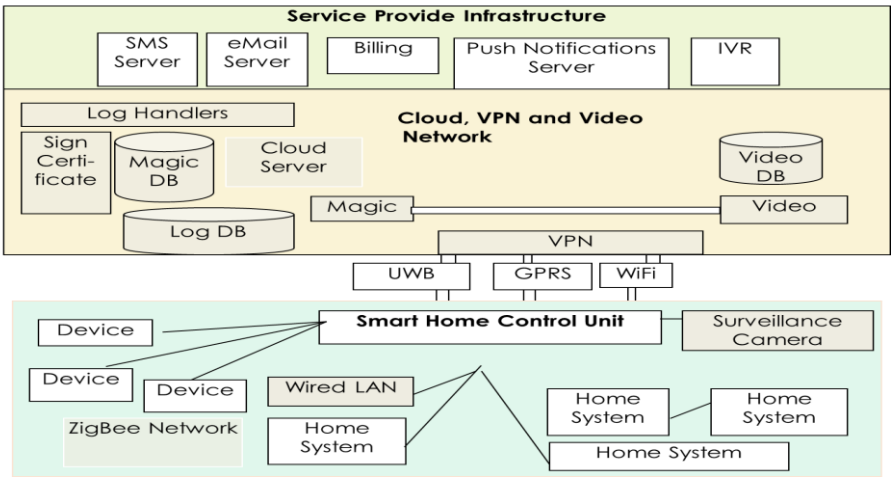
The architecture consists of three domains:

- M2M device domain
- M2M network domain
- M2M application domain

- The device management domain consists of three entities: physical devices, communication interface and gateway.
- Communication interface is a port or a subsystem which receives the input from one end sends the data received to another.
- M2M Network domain consists of M2M server, device identity management, data analytics and data & device management similar to IoT architecture level.

Examples of IoT:

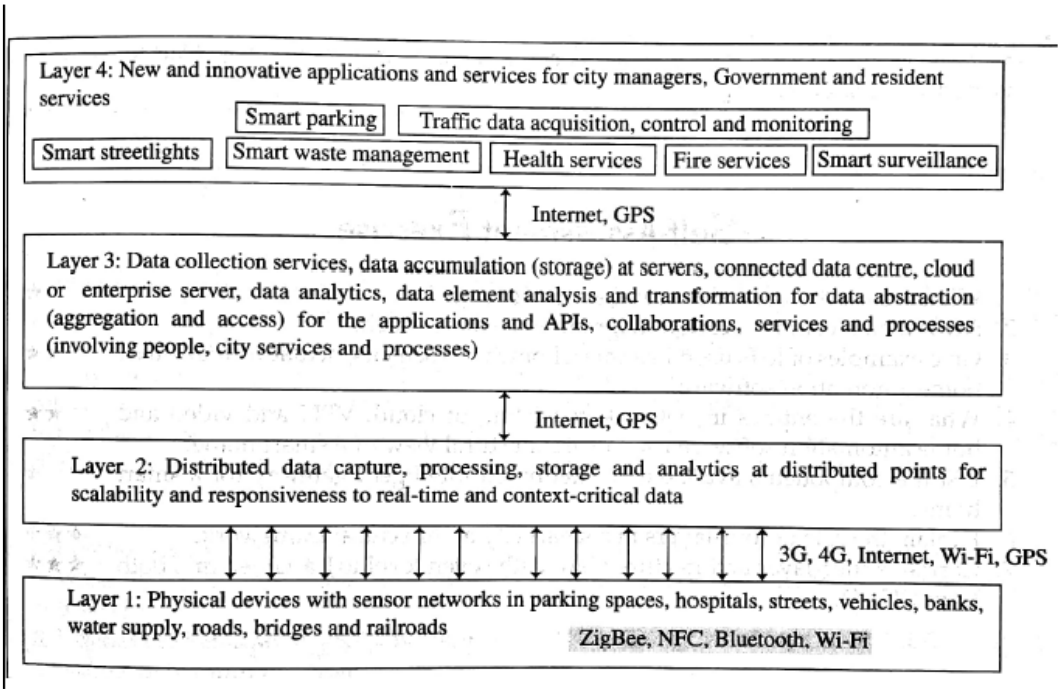
An example for Smart home automation/ Smart Home application:



- Sensors and actuators manage a smart home with an Internet connection. Wired and Wireless sensors are incorporated into the security sensors, thermostats and many more.
- In the device layer, the devices that are monitored like the temperature, lighting, power meter and so on are connected to a sensor.
- The sensor records any change in the operation of the device and communicates to the intermediate layer via UWB, GPRS or Wifi.
- Using the data is uploaded into cloud through Internet. With proper authentication the user can observe the changes at home.

- The cloud provides the information to the user by sending an email, and SMS, or Push Notifications for which the user could pay the electricity bill, telephone bill, switch off the lights or On the lights accordingly.
- This is an example of smart home automation using IoT.

An example for Smart City application:



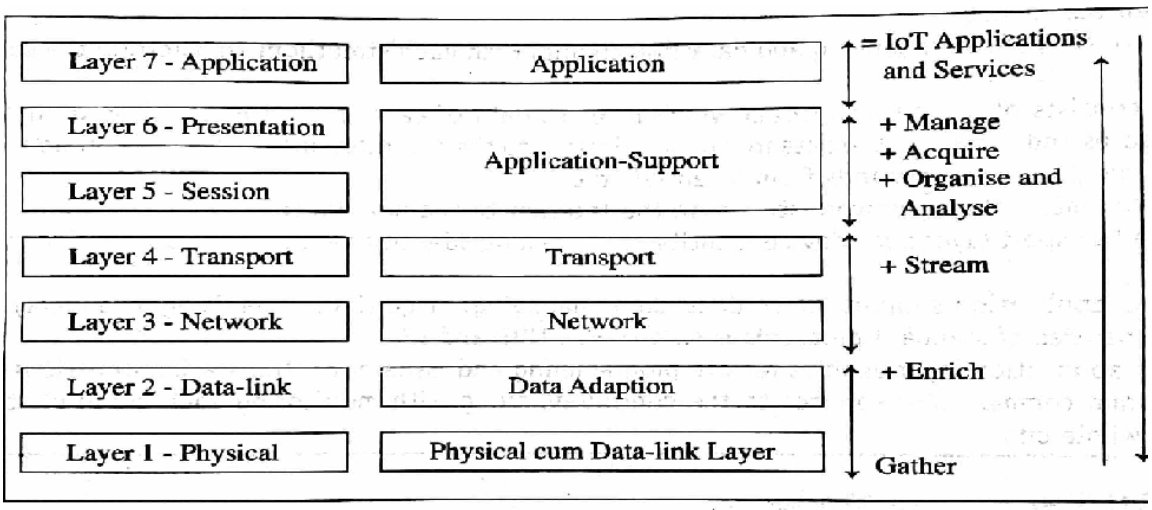
- The IoT technology can be expanded to construct a smart city.
- The feature of this application is that it connects traffic in the city to the hospitals to the schools via Internet.
- Layer 1 describes the physical device level. Here sensors are deployed in the parking space, hospitals, streets, vehicles, banks, water supply, roads, bridges and railroads.
- Layer 2: The data captured from the sensors is integrated and processed with the requirement.
- Layer 3: It is meant for central collection services, connected data centres and cloud.
- Layer 4: Consists of new innovative applications such as waste containers monitoring, WSN for power loss monitoring and to inform the concerned organisation.

Differences between IoT and M2M:

Parameters	M2M	IoT
Definition	M2M solutions contain a linear communication channel between various machines that enables them to form a work cycle. It's more of a cause and effect relation where one action triggers the other machinery into activity.	IoT can be defined as a system where multiple devices communicate with each other through sensors and digital connectivity. They talk to each other, work in tandem, and form a combined network of services.
Interactions	M2M refers to communication and interaction between machines & devices Such Interaction can occur via a cloud computing Infrastructure e.g. devices exchanging information through cloud infrastructure	IoT has broader scope than M2M, since it comprises broader range of interactions, including interactions between devices /things , things and people, things with applications and people with applications. It also enables composition of workflows comprising all of the above interactions
Interactivity	Machine to machine solutions operate by triggering responses based on an action. It's mainly a one-way communication.	The key advantage IoT has over M2M solutions is the ability to add interactivity amongst devices. In this system to and fro communication flows freely. There can be countless scenarios and combinations.
Connectivity Scope	M2M solutions rely primarily on conventional connection tools like wired connection , in wireless wifi , cellular , etc	IoT adds more sophisticated sensors into the mix. its result, Internet of Things based systems have much more flexible and varied connectivity options.
Solutions	M2M solutions, because of their limited scope, are confined to creating a network of machines that work in synchronization.	On the other hand, IoT creates 360° solutions that allow for flexible responses and multi-level communication.
Communications	Point to point communication usually embedded within hardware at customer site	Devices communicate using IP networks, incorporating with varying communication protocols
Integration	Limited integration option , as devices must have corresponding communication standards	Unlimited integration options, but requires a solution that can manage all the communications

IoT/M2M Systems, Layers and Design Standardisation:

Modified OSI Model for the IoT/ M2M Systems:



- The above diagram refers to the modified 7 layer OSI model for IoT/ M2M Systems.
- The modifications are proposed by IETF.
- Each layer proposes the received data and creates a new data stack which transfers it to the next layer.
- The processing takes place at the intermediate layers between the functional layer to the top layer.
- Device end also receives the data from the application/ service after processing.
- This shows a similarity to the operation of the equation 2 w.r.t conceptual framework as given below:
  Gather + Enrich + Stream + Manage + Acquire + Organise& Analyse =IoT with Connectivity to Data enter, Enterprise or Cloud

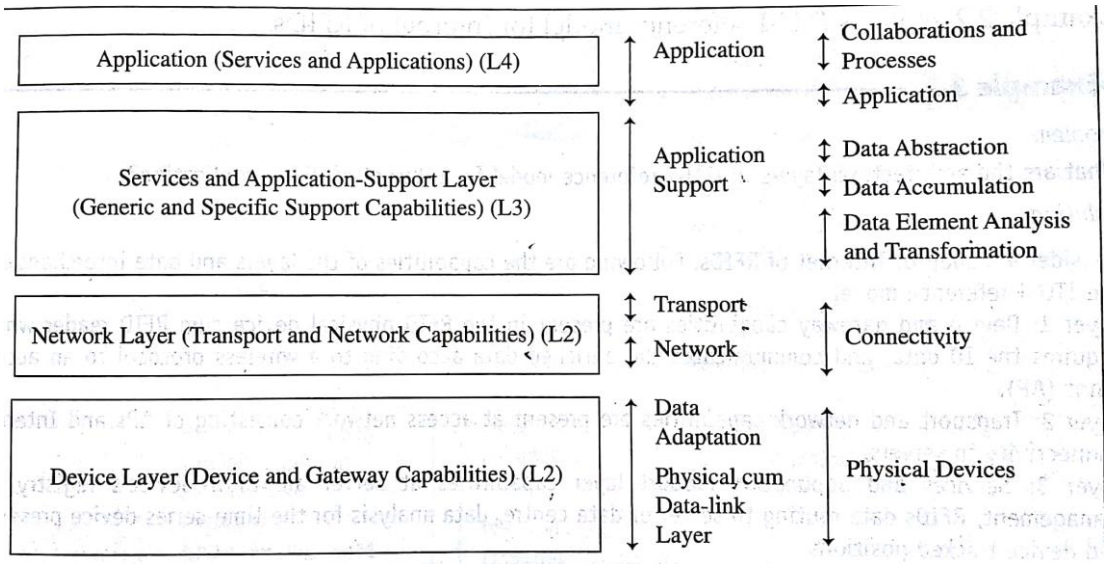
ITU-T Reference Model:

- The diagram below shows the ITU-T Reference Model called as RM1.
- This corresponds to the model with the six layers modified OSI model.
- Layer1: L1 is the device layer and has device and gateway capabilities.
- Layer2:L2 has transport and network capabilities.
- Layer3:L3 is the services and application-support layer. The support layer has two types f capabilities- Generic and specific service or application support capabilities.
- Top Layer: L4 is for applications and services.

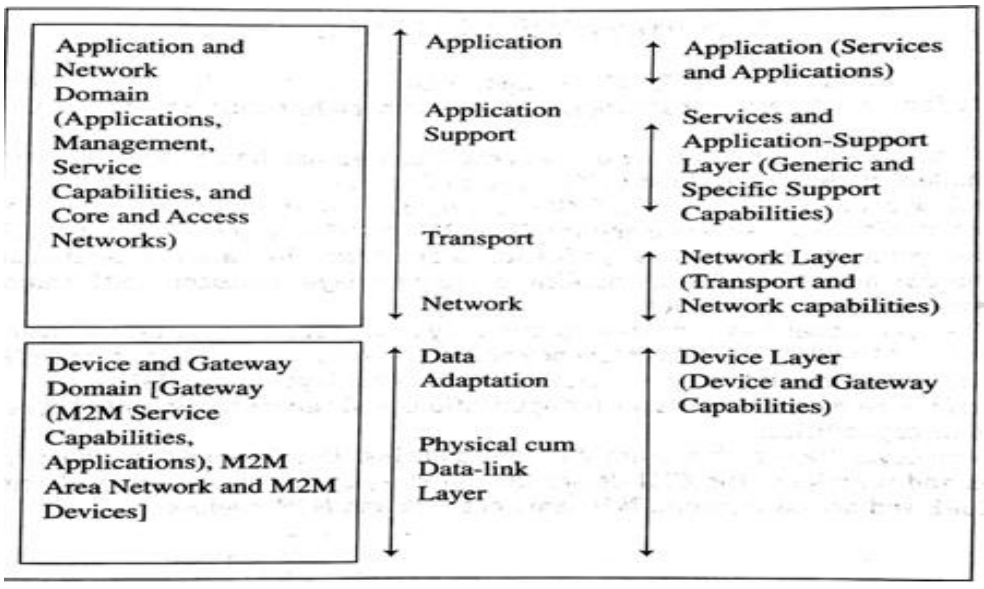
Comparison with the CISCI IoT reference model:

- L4 capabilities are similar to the Cisco Reference Model and processes and applications are of top two levels.

- L3 functions are similar to that of the middle level function of data abstraction, accumulation, analysis and transformation.
- L2 layer capabilities are similar to the connectivity in the Cisco Model.
- L1 device layer capabilities are similar to the physical devices level.



ETSI M2M Domain Architecture



- Like ITU-T , ETSI specifies the functional areas, a high level architecture and reference model for communicating the data from and to the IoT/ M2M devices
- The above diagram shows the ETSI M2M domains and architecture and the high level capabilities of each domain.
- It also depicts the architectural correspondence with the 6 layer modified OSI model and 4 layer of the ITU-T Reference model.
- The ETSI Network Domain has 6 capabilities and functions:
  - M2M Applications

- ❖ M2M Service Capabilities
- ❖ M2M Management functions
- ❖ Network Management Functions
- ❖ CoRE network ex: 3G and IP Networks
- ❖ Access network ,WLAN and Wi Max.

➢ ETSI device and gateway have the following functional units:

- ❖ Gateway between M2M area network, CoRE and access network , processing M2M service capabilities.
- ❖ M2M area network(Bluetooth, ZigBee, NFC, PAN,LAN)
- ❖ M2M Devices

- **Explain M2M ETSI domains and high level architecture for applications and services ATMs to bank servers.**
- **What are the architecture layers in ITU-T reference model for Internet of RFIDs application?**
- **What are the architectural layers in IoT? List the applications and advantages of IoT.**

Data Enrichment, Data Consolidation and Device Management at Gateway:

❖ A gateway at the data adaptation layer has several functions.
❖ These are data privacy, data security, data enrichment, data consolidation, transformation and device management.

Data Management and Consolidation Gateway:

➢ Gateway includes the following functions:

- ❖ Transcoding
- ❖ Privacy, Security
- ❖ Integration
- ❖ Compaction and fusion

➢ **Transcoding:** It means conversion and change of protocol, format or code using software.
➢ The gateway renders the web response the web and messages in formats and representations required and acceptable at an IoT device.
➢ IoT device requests are adapted, converted and changed into required formats acceptable at the server by the transcoding software.
➢ Ex. conversion from ASCII to Unicode at the server.
➢ A transcoding proxy can execute itself on the client system or the application server.
➢ It has conversional, computational and analysing capabilities while the gateway has conversion and computational capabilities only.

- **Privacy:** The data such as medical records, logistics, and inventories of a company may need privacy and protection.
- The following are the components of privacy model:
  - Devices and applications identity management
  - Authentication
  - Authorization
  - Trust
  - Reputation
- A suitable encryption method ensures data privacy.
- The data is decrypted and analysed and is an input to the application service or process.
- **Secure data access:** Access to data needs to be secured. The design needs to ensure the authentication of a request from a service or application.
- End to end security is a feature which uses a security protocol at each layer.
- **Data gathering and Enrichment:** IoT applications involve actions such as Data gathering(Acquisition), Validation, Storage ,Processing, Retention and analysis.
- Data gathering is to acquire the data from the devices or device networks. Four modes of Acquisition are:
  - **Polling:** Refers to the data sought by addressing the device[Its operated like the polling by a computer to access the control of a channel to transfer data or to check if there is a data addressed to it]
  - **Event Based:** The data acquired from the device on an event like a NFC or a card reader.
  - **Scheduled Interval:** The data acquired from the device at selected intervals. Ex: changes in the lighting condition of street lights.
  - **Continuous Monitoring:** Refers to the data sought from the device continuously. Ex: Data for traffic monitoring.
- **Data Dissemination: (Dissemination means to distribute, broadcast, diffuse or spread)**

  There are three steps in the data enrichment before data dissemination
  - **Aggregation:** Refers to the process of joining together present and previously received data frames after removing redundant or duplicate data.
  - **Compaction:** means making information short without changing the meaning or context; ex. transmitting only the incremental value of the data so that the information is short.
  - **Fusion:** formatting the information received in parts through various data frames and several types of data, removing redundancy in the received data.

- When the data transmission takes place in the wireless environment the energy dissipation or power consumption is a criteria. This is due to the battery life in the WSNs.
- Energy efficient computations can be made use of by using the concepts of data aggregation, compaction and fusion.
- **Data Source and Data Destination: ID:** Each device and resource is assigned an ID for specifying the data of source and a separate ID for data destination.
- **Address:** Header fields add the destination address.
- **Data Characteristics, Formats and structures:** Data characteristics can be in terms of temporal data i.e. dependent on time, Spatial Data i.e. dependent on location, real time data i.e. generated continuously and acquired continuously at the same pace, real world data i.e. ex: traffic or streetlight, Proprietary data i.e. data reserved with copy rights to authorised enterprises and Big Data i.e. unstructured voluminous data.
- Data received from the devices can be in different formats for further communication like: XML, JSON, TLV. The structure implies the ways for arranging the data bytes in sequences with size limit.
- **Device Management (DM) at gateway:** DM means provisioning the device ID or address which is distinct from other sources, device activating, configuring, registration, deregistering, attaching and detaching.
- DM also means accepting subscription for its resources.
- **Open Mobile alliance** (OMA)-DM and several standards for device management.
- OMA-DM model suggests the use of a DM server which interacts with devices through a gateway in case of IoT/M2M application.
- Gateway functions for device management are:
  - Forwarding the data/ request when the DM server and device interact without structuring.
  - Protocol conversion when the device and DM server use distinct protocols.
  - Proxy functions in case of intermediate pre fetch is required in a lossy environment or network environment needs.

Web communication protocols used by connected IoT/M2M devices

- An IoT/M2M device network gateway needs connectivity to web services.
- A communication gateway enables web connectivity, while IoT/M2M methods and protocols enable the web connectivity for a connected device network.

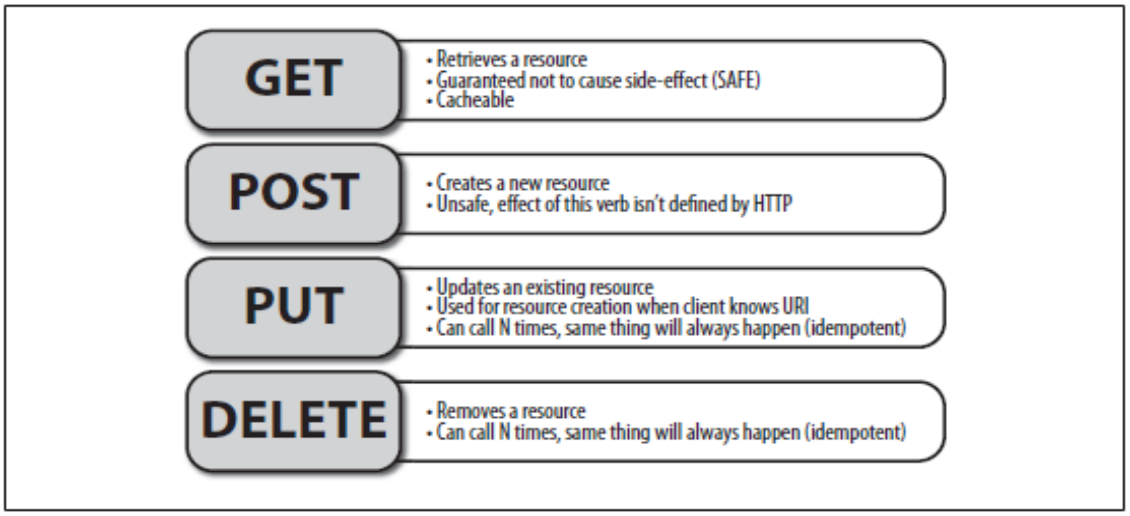
- Following are the key terms used in communication:
  - **Application or APP**: refers to software for applications for creating and sending an SMS, measuring and sending the measured data.
  - **Application Programming Interface(API):** refers to a software component, which receives messages from one end. It might consist of GUIs (Button, Checkbox and Dialog Box).
  - **Web service:** refers to a servicing software which web protocols, web objects utilize ex. Weather reports, traffic density.
  - **Object:** refers to collection of resources.
  - **Object Model:** defined as the usage of objects for values, messages, data or resource transfer, and creation of one or more object instances.
  - **Class:** It creates one or more instances.
  - **Communication Gateway:** functions as a communication protocol translator.
  - **Client:** refers to a software object which makes request for data, messages, resources or objects.
  - **Server:** is defining as software which sends a response on a request.
  - **Web Object:** That retrieves a resource from the web object at other end using a web protocol.
  - **Broker:** denotes an object which arranges the communication between two end point devices.
  - **Proxy:** an application which receives a response from the server for usage of the client or application and which also requests from the client for the responses retrieved or saved at proxy.
  - **Communication protocol:** defines the rules and conventions for communication between the web server and web clients.
  - **Web 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.
  - **Firewall:** is one that protects the server from unauthentic resources.
  - **Universal Resource Locator:** is generally used for retrieving resources by a client.
  - **Representational State Transfer (REST):** is a software architecture referring to ways of defining the identifiers for the resources, methods, access methods and data transfer during interactions.
  - REST also refers to usage of defined resource types when transferring the objects between two ends-URIs or URLs for representations of the resource.

- ❖ REST also refers to the usage of use verbs(commands), POST, GET, PUT and DELETE.
- ❖ RESTful refers to one which follows REST constraints and characteristics.

Web Communication Protocols for Connected Devices:

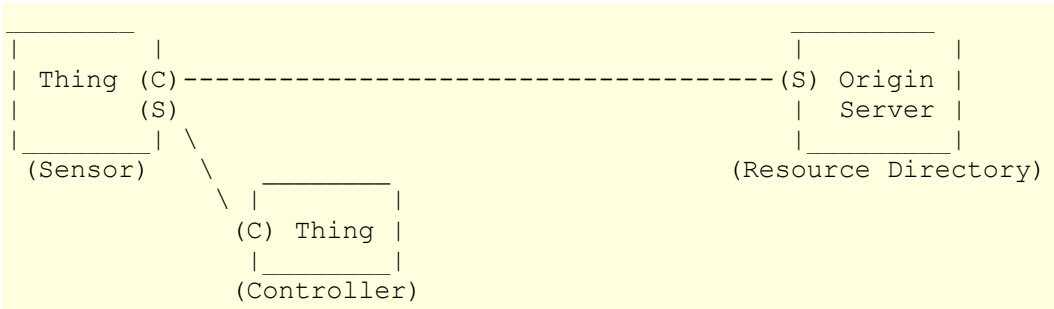
REST-Representation State Transfer

- ➢ REST (Representational State Transfer) is an architectural style for developing web services.
- ➢ REST is popular due to its simplicity and the fact that it builds upon existing systems and features of the internet's HTTP in order to achieve its objectives, as opposed to creating new standards, frameworks and technologies.
- ➢ It is implemented by using the following to fetch, maintain, enrich, update and append the data.



CoRE: Constrained RESTful Environment:

IoT devices or M2M devices communicate between themselves in a Local Area Network



- Nodes in IoT systems often implement both roles.
- Unlike intermediaries, however, they can take the initiative as a client (e.g., to register with a directory, such as CoRE Resource Directory or to interact with another thing) and act as origin server at the same time (e.g., to serve sensor values or provide an actuator interface).

Features:

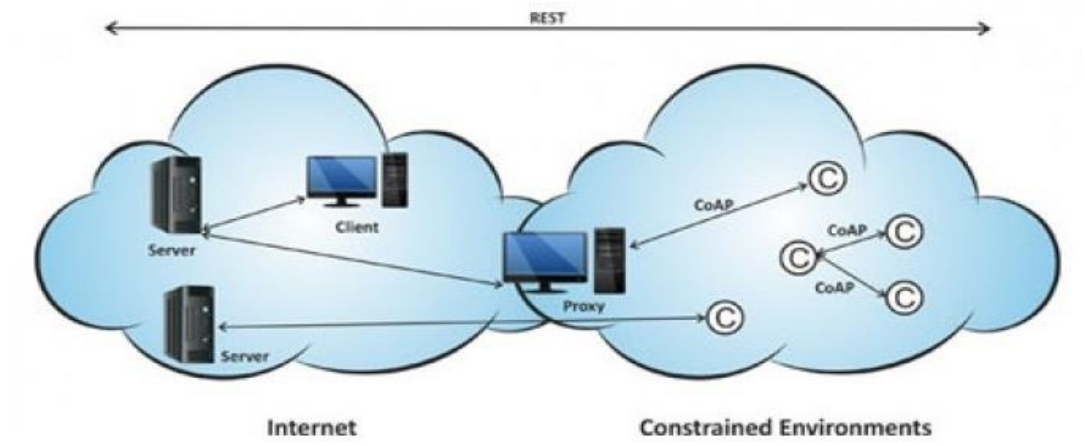
- Devices have a constraint in the sense that their data is limited in size compared to when data interchange between web clients and web servers takes place using HTTP, TCP and IP.
- Data Routing is another constraint when **Routing Over a Network of Low Power and (data) Loss- ROLL**.
- ROLL network is a low power wireless network.
- The devices may sleep most of the time in a low power environment and awaken on an event or when required.

Unconstrained Environment:

- Web applications use HTTP and RESTful HTTP for web client and web server communication.
- A web object consists of 1000s of bytes.
- Data routes over IP networks for the Internet.

Constrained Application Protocol

- Constrained Application Protocol (aka CoAP) is a specialized web transfer protocol for use with constrained nodes (low power sensors and actuators) and constrained networks (low power, lossy network).
- It enables those nodes to be able to talk with other constrained nodes over Internet.
- The protocol is specifically designed for M2M applications such as smart energy, home automation and many Industrial applications.



- CoAP protocol is necessary because traditional protocols such as TCP/IP are considered "too heavy" for IoT applications that involves constrained devices.
- CoAP protocol runs on devices that support UDP protocol. In UDP protocol, client and server communicate through connectionless datagrams.
- As it is a web transfer protocol, it is based on RESTful architecture which provides a request/response interaction model between application endpoints and supports built-in discovery of services and resources.
- Like HTTP, Servers make resources under URL and clients access those resources using methods such as GET, PUT, POST and DELETE.

The CoAP protocol has the following features

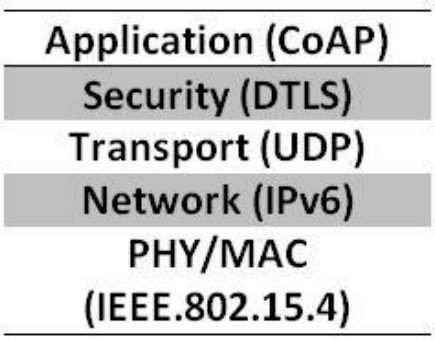
- It provides M2M communication in constrained environment.
- It provides security of data by datagram transportation layer security (DTLS).
- Asynchronous message exchange.
- Low header overhead and parsing complexity
- URI and content type support
- UDP binding with optional reliability supporting unicast and multicast requests.

- The CoAP is different from other protocols.
- When compared with HTTP, CoAP is implemented for IoT and M2M environment to send messages over UDP protocol.
- To compensate for the unreliability of UDP protocol, CoAP defines a retransmission mechanism and provides resource discovery mechanism with resource description.

CoAP should be on priority for the following three factors

- ❖ Quality of service with confirmable message
- ❖ When multicast support is needed
- ❖ Very low overhead and simplicity.

- ➢ CoAP follows a client-server communication model.
- ➢ Client makes request to the server and the server sends back the responses to the client.
- ➢ Client can GET, PUT, POST or DELETE the resources on network.
- ➢ CoAP improves the HTTP request model with the ability to observe a resource.
- ➢ In HTTP, the server needs to do polling again and again to check where there is any state changes to the client or not.
- ➢ Whereas in CoAP, the observe flag is set on the CoAP GET request, the server continues to reply after the initial document has been transferred.
- ➢ This allows servers to stream the state changes to clients as they occur. Any end can stop the observation.
- ➢ The CoAP defines a standard mechanism for resource discovery.
- ➢ Servers provide a list of their resources, along with metadata about them, at /.well-known/core. For Quality of Service (QoS), Requests and response messages may be marked as confirmable or non-confirmable.
- ➢ Confirmable messages must be acknowledged by the receiver. Non-confirmable messages are “fire and forget” type.



CoAP Protocol at Application Layer in Network Architecture

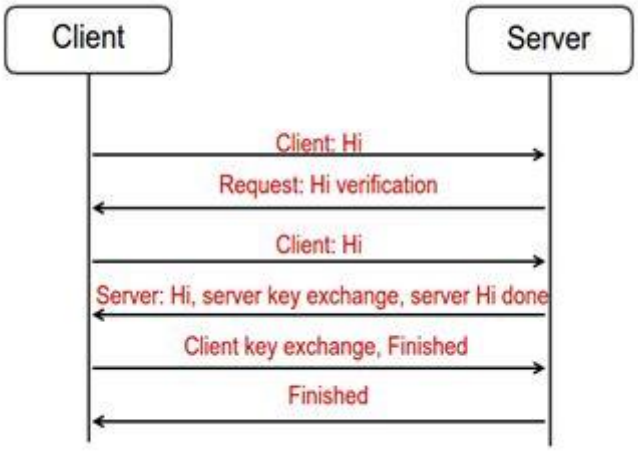
CoAP Protocol Security

- ➢ The main concern from security point of view is to provide Data Integrity, Data Authentication and Data Confidentiality.
- ➢ The CoAP provides security over Datagram Transportation Layer Security in Application layer.
- ➢ As CoAP runs over UDP protocol stack, there are chances of data loss or data disordering. But with DTLS security, these two problems can be solved.

DTLS security adds three implementations to CoAP

1) Packet retransmission
2) Assigning sequence number within handshake
3) Replay detection

- The security is designed to prevent eavesdropping, tampering or data forgery at any cost.
- Unlike network layer security protocols, DTLS in application layer protect end-to-end communication.
- DTLS also avoids cryptographic overhead problems that occur in lower layer security protocols.
- There is a Secured Handshake Mechanism in DTLS as shown in image below



DTLS Secured Handshake Mechanism for CoAP

- The CoAP can also be implemented over TCP and over TLS.
- Check out the following official documentation for CoAP implementation over TCP and TLS.
- An important part of RESTful API design is to model the system as a set of resources whose state can be retrieved and/or modified and where resources can be potentially also created and/or deleted.
- Uniform Resource Identifiers (URIs) are used to indicate a resource for interaction, to reference a resource from another resource, to advertise or bookmark a resource, or to index a resource by search engines.

```
  foo://example.com:8042/over/there?name=ferret#nose
  \_/   \______________/\_________/ \_________/ \__/
   |           |            |            |        |
scheme     authority       path        query   fragment
```

CoAP SMS

- Is a protocol when CoAP object uses IP with networks and uses SMS.
- SMS is used instead of UDP+DTLS by CoAP client server.
- Client communicates to a mobile terminal(MT) endpoint over GPRS,HSPA or LTE using CoAP-SMS protocol.

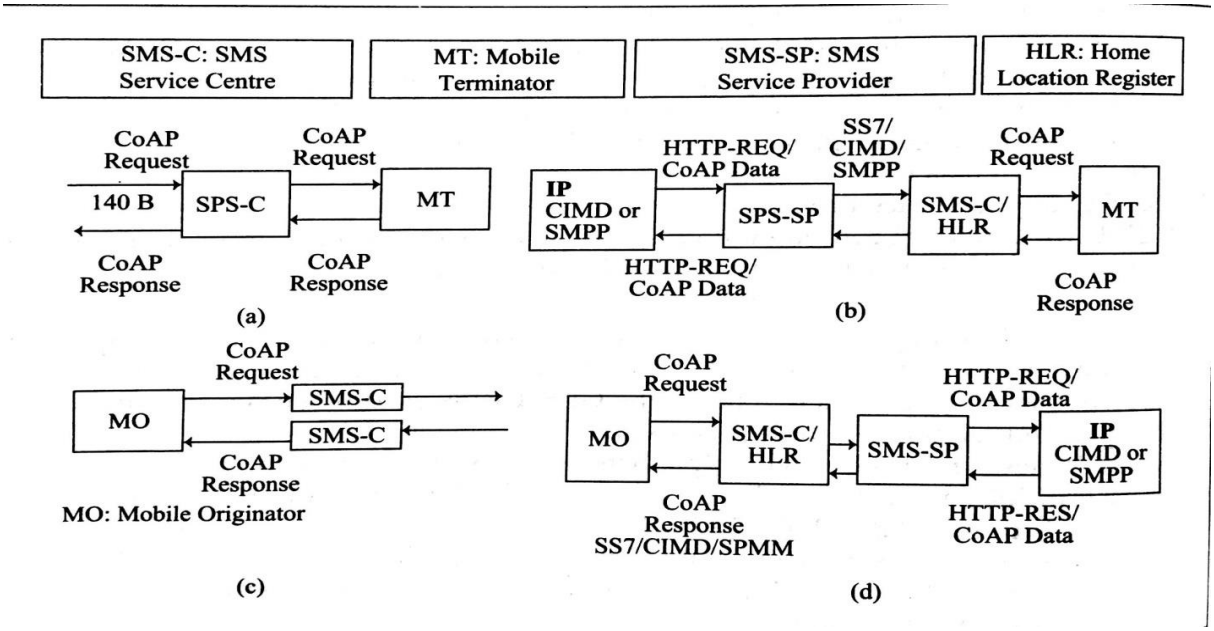
CoAP-SMS features:

- An URI(Universal Resource Identifier) is used to send specified telephone number.

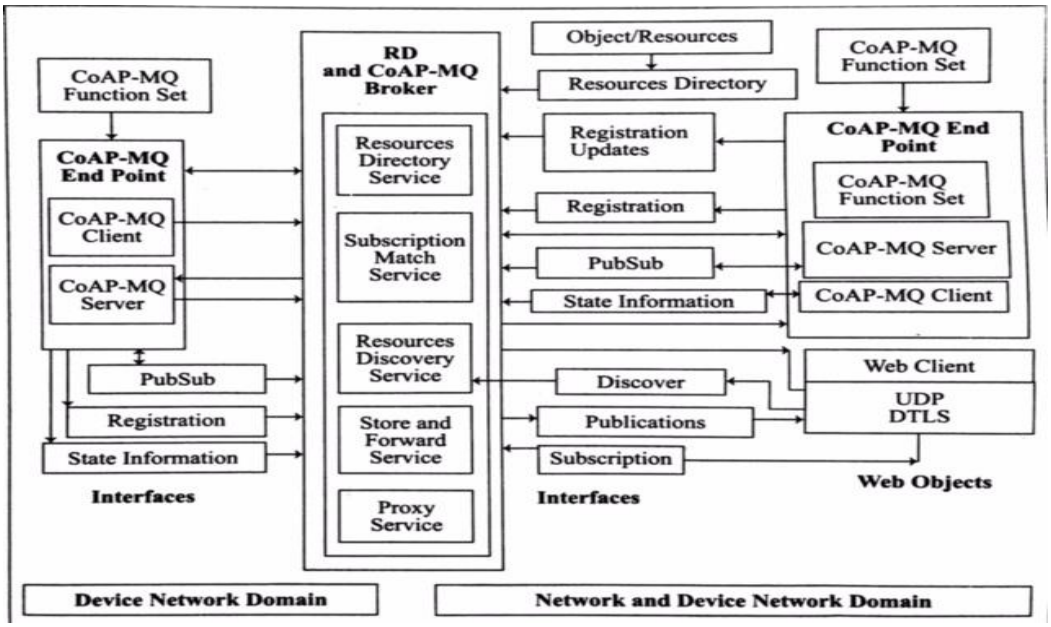
  Example: coap+sms://telNum/…….

- CoAP msg consists of 160 character in 7-bits/8 bits
- CoAP works with SIM(subscriber Identity Module) for SMS in cellular networks.
- Does not support multi-casting
- Two options are availble RUH(Response to URI-Host) and RUP(Response to URI-Port) for initiating CoAP client to know about the alternative interface are CIMP and SMPP
- MSISDN and SIM based security is used during SMs data exchange.

- CoAP request or response communication to a machine, IoT device or mobile terminal (MT) fig(a).
- A computer or machine interface using IP communication to a mobile service provider for data interchange with terminal fig(b)
- A machine or IoT device or mobile origin (MO) communication of CoAP request or response communication fig(c)
- An origin communication using SS7/CIMD/SMPP with a computer or machine interface using IP communication.



CoAP MQ:



CoAP-MQ feature

- It is message queue protocol.
- CoAP provides resource-subcription, from publishers.
- The device objects communicate using the CoAP client and server protocol and CoAP web object using DTLS as security protocol
- UDP for CoAP APIs.

Lightweight Machine to machine Communication Protocol:

- It is a Communication protocol at the application layer.
- Specified by OMA-Open Mobile Alliance for transfer of data/ message.
- Why Light Weight Management: It is widely used for mobile devices, low cost remote management and service enabled mechanism that works on wireless connection.
- It provides data management as well as application data handling.

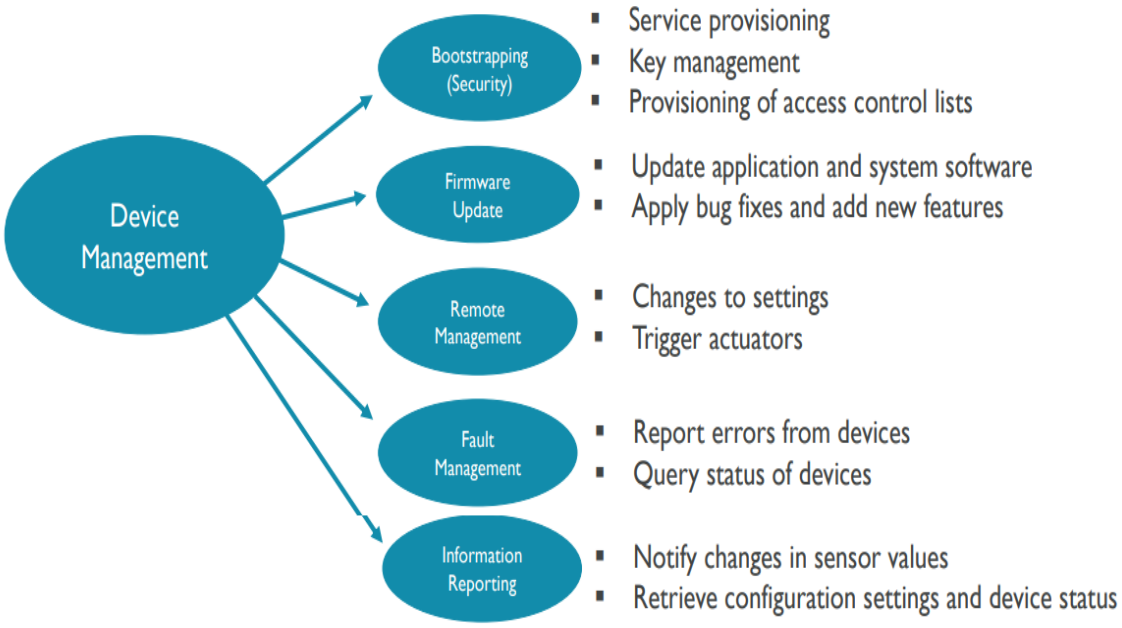
Features:

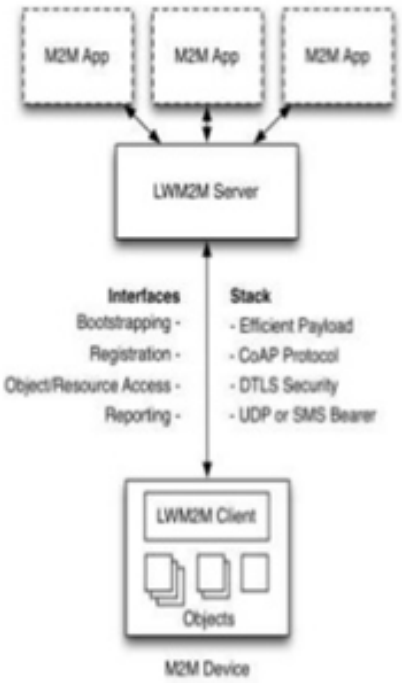
- An object or resource use CoAP, DTLS and UDP or SMS protocols for sending a request or response.
- Use of plain text for a resource or use of JSON during a single data transferor binary TLV format data transfer.
- An object or its resource access using an URI.
- It uses 3 types of Interface functions:
  - Bootstrapping

- Registration
- Report

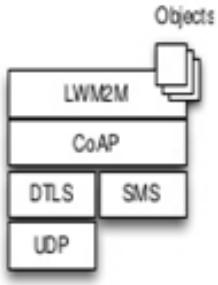
Advantages:

- Enables plug and play solution between an increasing variety of M2M
- Enables independent innovation of M2M applications and M2M platforms





- M2M Applications
  - Application abstraction through REST API
  - Resource Discovery and Linking
- LWM2M Server
  - Reuses IETF technologies, such as the CoAP protocol, DTLS, Resource Directory.
  - Deployable on gateways and in the cloud
- LWM2M Clients are Devices
  - Device abstraction through CoAP
  - LWM2M Clients are CoAP Servers
  - Any IP network connection



MQTT- Message Queue Telemetry Transport:

- An open source protocol for machine-to-machine (M2M)/"Internet of Things" connectivity
- Created by IBM
- The objects communicating using the Connected devices network protocols, such as ZigBee.
- Web objects also using MQTT library functions and communicate using IP network and SSL and TLS security protocols

MQTT Features

- Constrained environment protocol.
- PubSub messaging architecture in place of request-response client-server architecture
- publisher (message sender at the device domain or web object at network and application domain) sending the messages on a topic.
- Subscriber (message receiver at the device domain or web object at network and application domain) receiving the messages on a subscribed topic
- Lightweight, running on limited resources of processor and memory processor or memory resources
- Header of fixed-length header and two bytes only
- M2Mqtt library providing a set of functions for coding
- M2Mqtt library functions in Java needing just 100 kB and in C# is 30 kB,
- Minimum number of exchanges, and therefore lessening the network traffic
- Three Quality of Services
- MQTT TCP/IP Connectivity
- Broker-based publish/subscribe messaging protocol
- publish/subscribe functions enable one-to-many message distribution decoupled with the applications (unconcerned about the payload)
- Notifying on an abnormal disconnection of a client, notified all nodes subscribing to the message
- The last will specifying the final action to be taken on failure to send the messages.

MQTT Broker Functions

- Store and forward
- Clients publish topics and receives topics on subscription
- Recovers subscriptions on reconnect after a disconnection, unless client explicitly disconnected
- Acts as a broker between publisher of the topics and subscribers of the topics
- Finds client disconnection until DISCONNET message receives, keeps message alive till explicit disconnection
- retains the last received message from a publisher for a new connected subscriber on same topic, when retain field in the header is set.

XMPP (Extensible Messaging and Presence Protocol)

- 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.
- XMPP was originally developed in the Jabber open-source community to provide an **open, decentralized alternative to the closed instant messaging services** at that time.
- XMPP offers several key advantages over such services:
  - **Open** — the XMPP protocols are free, open, public, and easily understandable; in addition, multiple implementations exist in the form clients, servers, server components, and code libraries.
  - **Standard** — the Internet Engineering Task Force (IETF) has formalized the core XML streaming protocols as an approved instant messaging and presence technology.
  - **Decentralized** — the architecture of the XMPP network is similar to email; as a result, anyone can run their own XMPP server, enabling individuals and organizations to take control of their communications experience.
  - **Secure** — any XMPP server may be isolated from the public network (e.g., on a company intranet)
  - **Extensible** — using the power of XML, anyone can build custom functionality on top of the core protocols; to maintain interoperability, common extensions.
  - **Flexible** — XMPP applications beyond IM include network management, content syndication, collaboration tools, file sharing, gaming, remote systems monitoring, web services, lightweight middleware, cloud computing, and much more.
  - **Diverse** — a wide range of companies and open-source projects use XMPP to build and deploy real-time applications and services; you will never get "locked in" when you use XMPP technologies.