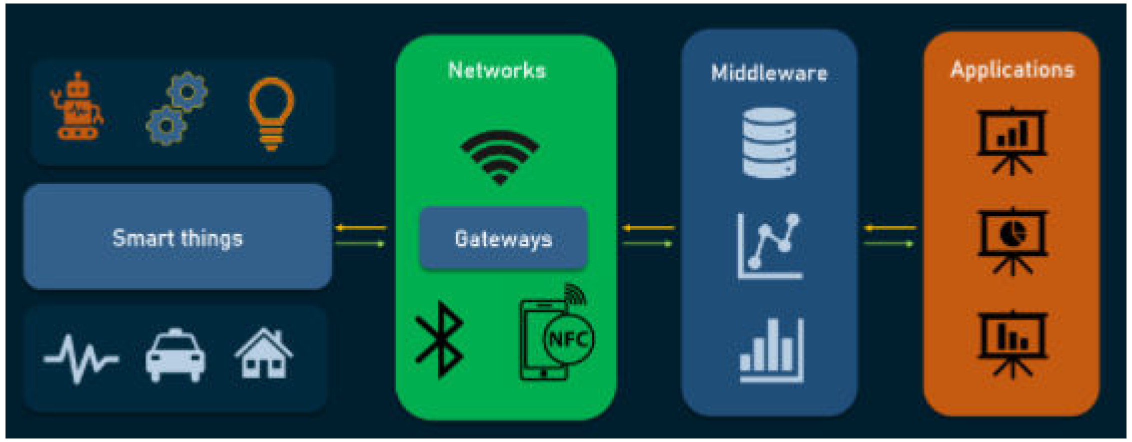
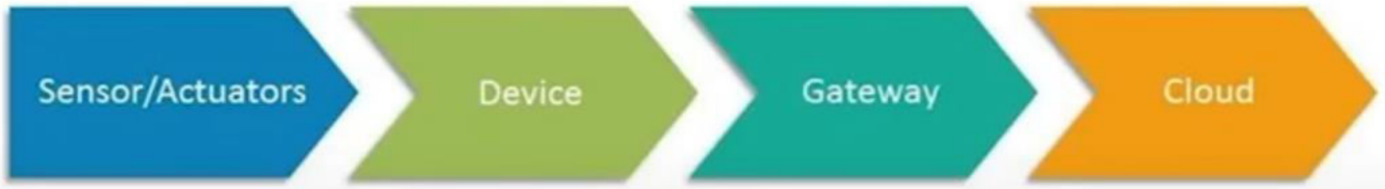
Chapter 2
Architectural Overview of IOT



State of the art – IoT Architecture

Having or using the latest techniques or equipment.

- IoT architecture varies from solution to solution, based on the type of solution which we intend to build. IoT as a technology majorly consists of four main components, over which an architecture is framed.
  - Sensors
  - Devices
  - Gateway
  - Cloud



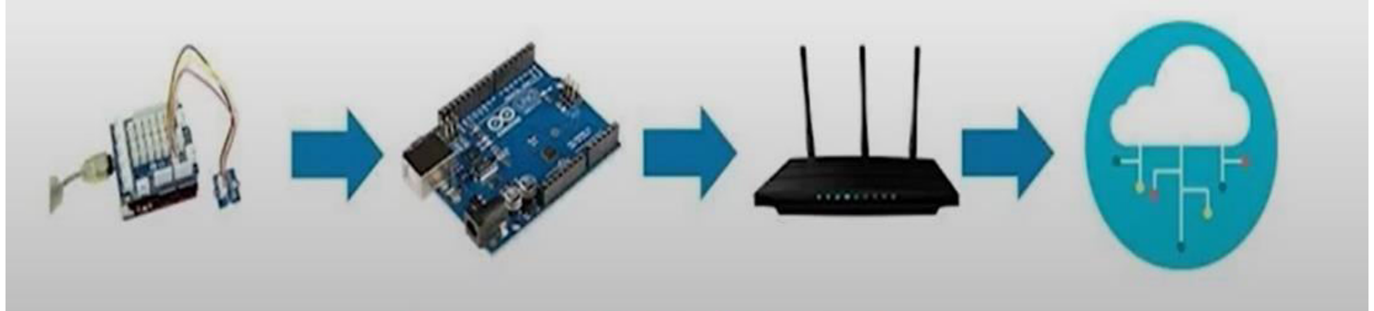


Figure: State of the art

Reference Model and Architecture (ARM)

IOT Reference Model

Architecture of
IOT Reference Model

IOT Reference Model

Fundamentals of IoT Reference model

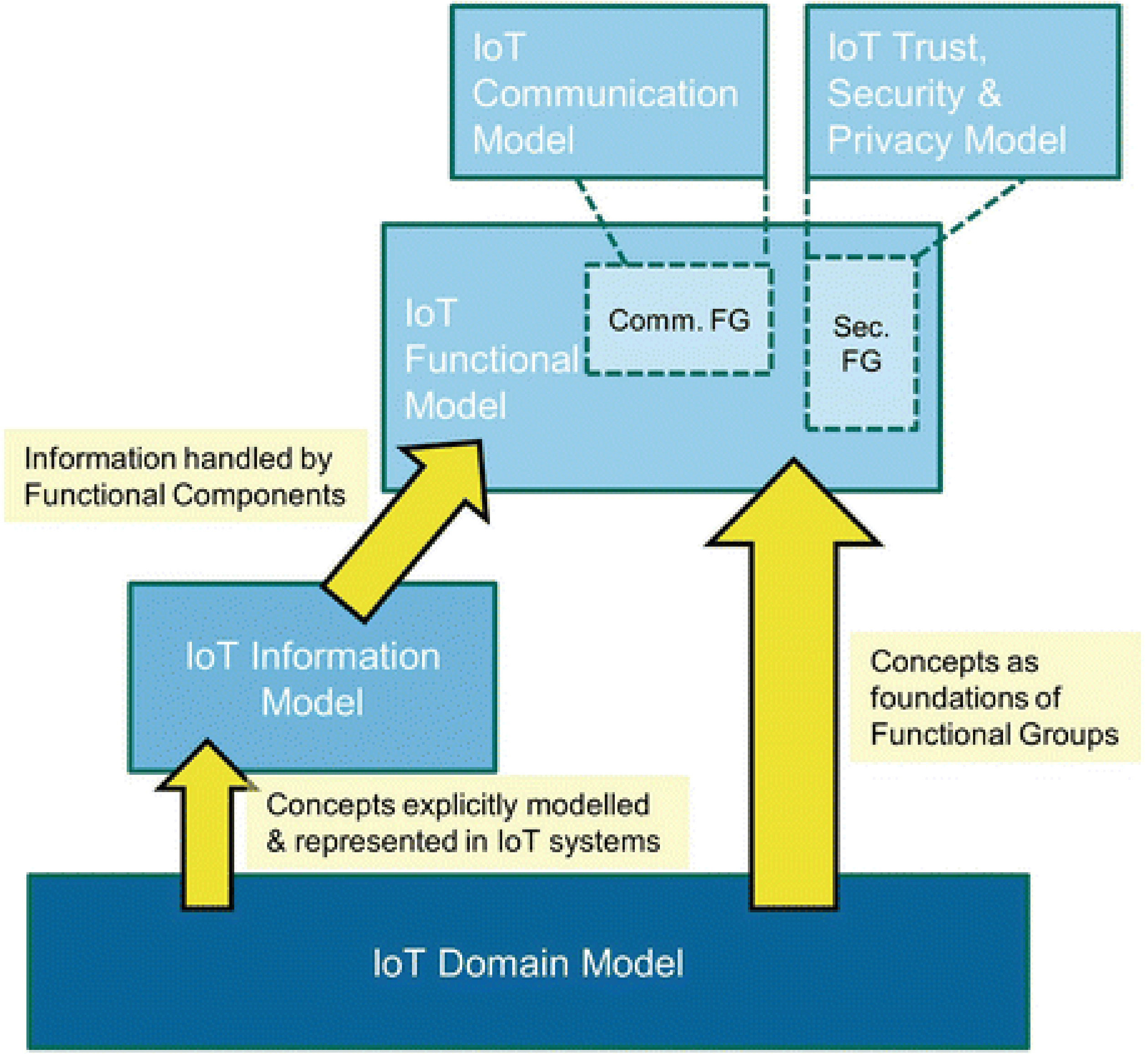
- An IoT system has multiple levels, each level explains distinct functionalities and Components of the IoT network. These levels include different physical devices, connectivity protocols, data processing, applications, etc.
- These levels are also known as tiers. Tiers mean the hierarchical structure of the IoT network. Each tier has specific roles and responsibilities.
- Several reference architectures are expected to co-exist in the IoT domain. There can be many reference architectures of IoT systems for different applications. These reference architectures provide guidelines, best practices, and common patterns for designing and implementing IoT solutions.
- A model enables the conceptualization of a framework.
- A reference model can be used to depict building blocks, successive interactions, and integration of various IoT components.

Define the role of IoT reference model in IoT system (03 marks) (any 3 point each for 1 mark)

IOT Reference Model

Draw IoT reference model and explain (06 marks)

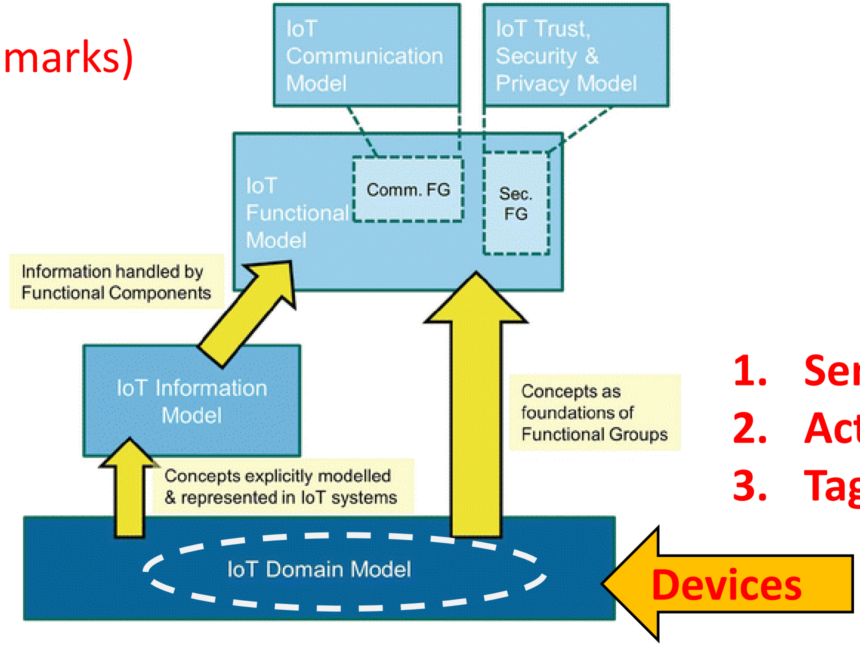
- A reference model describe the domain of IoT system using a number of **sub models.**
- The IoT Reference Model provides the highest abstraction level for the definition of the IoT-A Architectural Reference Model.
- It promotes a common understanding of the IoT domain.
- The description of the IoT Reference Model includes a general discourse on the IoT domain, an IoT Domain Model as a top-level description, an IoT Information Model explaining how IoT information is going to be modelled, and an IoT Communication Model in order to understand specifics about communication between many heterogeneous IoT devices and the Internet as a whole.


IoT Communication Model
IoT Trust, Security & Privacy Model
IoT Functional Model
Comm. FG
Sec. FG
Information handled by Functional Components
IoT Information Model
Concepts as foundations of Functional Groups
Concepts explicitly modelled & represented in IoT systems
IoT Domain Model

IoT domain model

Explain IoT domain model of IoT reference model (04 marks)

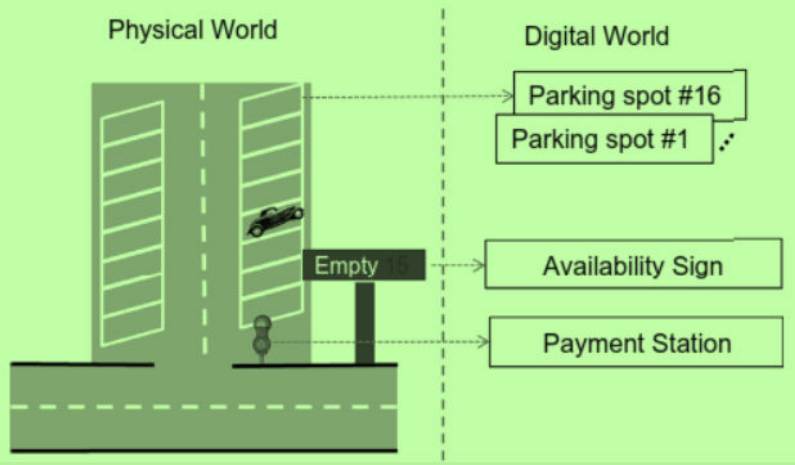
- The **base** of the reference model is IoT Domain model.
- The domain model captures the basic attributes of the **main concepts** and the **relationship** between these concepts.
- A domain model also serves as a tool for human communication between people working in the domain in question and between people who work across different domains.


IoT Communication Model
IoT Trust, Security & Privacy Model
IoT Functional Model
Comm. FG
Sec. FG
Information handled by Functional Components
IoT Information Model
Concepts as foundations of Functional Groups
Concepts explicitly modelled & represented in IoT systems
IoT Domain Model
Devices

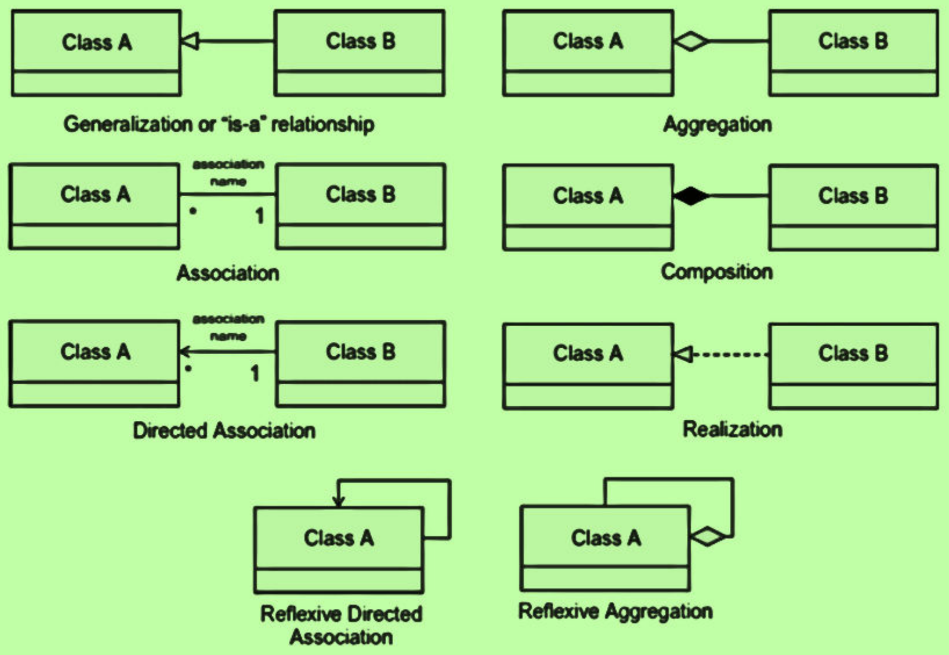
1. **Sensors**
2. **Actuators**
3. **Tags**

Main concepts

The IoT is a support infrastructure for enabling objects and places in the physical world to have a corresponding representation in the digital world.



Model notation and semantics



UML Class diagram main modelling concepts

Define the role of devices associated with IoT domain model of IoT reference model. (each for 02 marks)

The Devices are physical artefacts with which the physical and virtual worlds interact. Devices as mentioned before can also be Physical Entities for certain types of applications, such as management applications when the interesting entities of a system are the Devices themselves and not the surrounding environment. For the IoT Domain Model, three kinds of Device types are the most important:

1. **Sensors:**

- ✓ These are simple or complex Devices that typically involve a transducer that converts physical properties such as temperature into electrical signals.
- ✓ These Devices include the necessary conversion of analog electrical signals into digital signals, e.g. a voltage level to a 16-bit number, processing for simple calculations, potential storage for intermediate results, and potentially communication capabilities to transmit the digital representation of the physical property as well receive commands.
- ✓ A video camera can be another example of a complex sensor that could detect and recognise people.

2. **Actuators :**

- ✓ These are also simple or complex Devices that involve a transducer that converts electrical signals to a change in a physical property (e.g. turn on a switch or move a motor).
- ✓ These Devices also include potential communication capabilities, storage of intermediate commands, processing, and conversion of digital signals to analog electrical signals.

3. Tags:

- ✓ Tags in general identify the Physical Entity that they are attached to. In reality, tags can be **Devices** or **Physical Entities** but not both, as the domain model shows.

- ✓ An example of a Tag as a **Device is a Radio Frequency Identification (RFID)** tag, while a tag as a **Physical Entity is a paper-printed immutable barcode or Quick Response (QR) code.**

- ✓ Either electronic Devices or a paper-printed entity tag contains a unique identification that can be read by optical means (bar codes or QR codes) or radio signals (RFID tags).

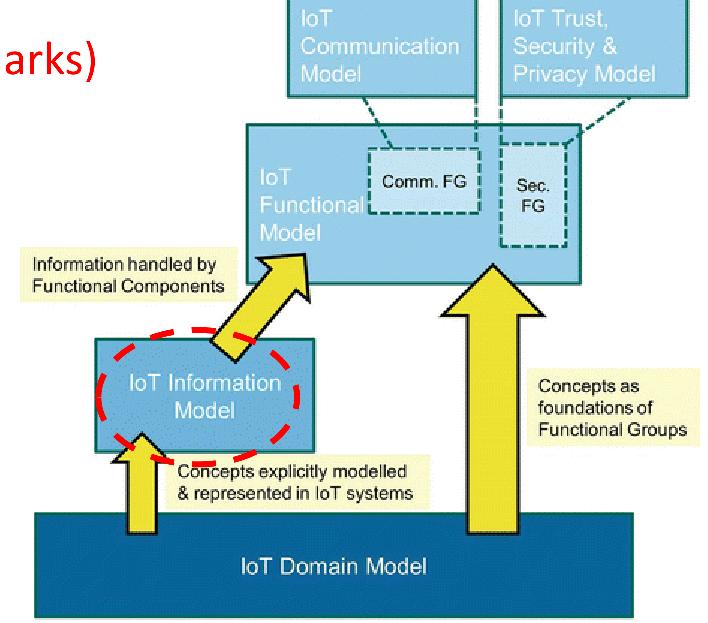
- ✓ The reader Device operating on a tag is typically a sensor, and sometimes a sensor and an actuator combined in the case of writable RFID tags.

Differentiate between RFID and Barcode (each for 01 marks) 3/4/6

S.No.	RFID	Barcode
1.	It is based totally on radio-frequency.	It is primarily based on optical technology.
2.	It does no longer require Line of sight.	It requires Line of sight because scanner must have an unobstructed view and must be oriented properly.
3.	It has greater data storage as in contrast to barcodes.	It has much less data storage up to solely 24 characters.
4.	Memory storage is possible in RFID with assist of tags.	Memory storage is not possible in barcodes.
5.	It is more resistant or durable than a barcode.	It is much less resistant than RFID.
6.	Several RFID tags can be examined simultaneously i.e. multiple read is allowed in RFID.	Only a single barcode can be scanned at a time i.e one card can be read at a time.
7.	Read/write abilities using RFID tags.	Barcode has totally reading capabilities and can’t write anything.
8.	It processes faster than Barcode.	It is slower than RFID.

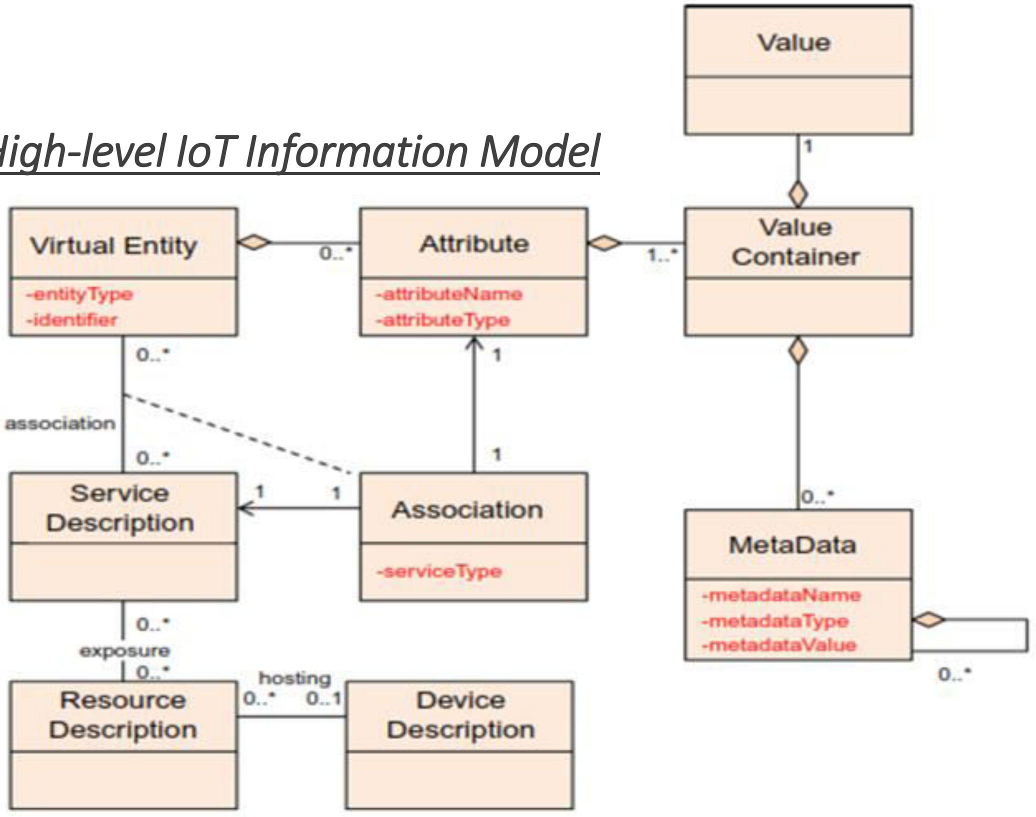
IOT Information Model:

Explain IoT information model of IoT reference model (04 marks)

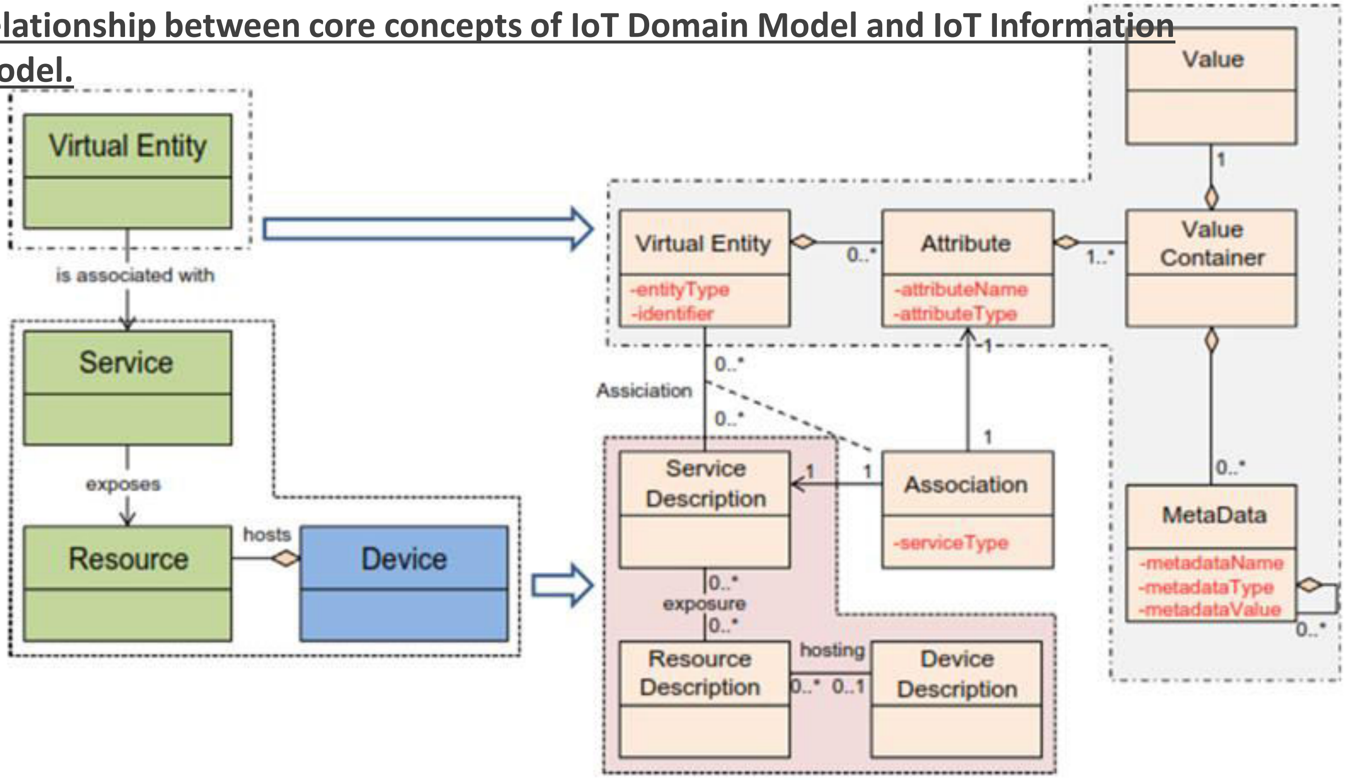

IoT Communication Model
IoT Trust, Security & Privacy Model
IoT Functional Model
Comm. FG
Sec. FG
Information handled by Functional Components
IoT Information Model
Concepts as foundations of Functional Groups
Concepts explicitly modelled & represented in IoT systems
IoT Domain Model

- Virtual Entity in the IoT Domain Model is the "Thing" in the Internet of Things, the IoT information model captures the details of a Virtual Entity- centric model.
- Similar to the IoT Domain Model, the IoT Information Model is presented using **Unified Modelling Language (UML) diagrams**.

High-level IoT Information Model



Relationship between core concepts of IoT Domain Model and IoT Information Model.

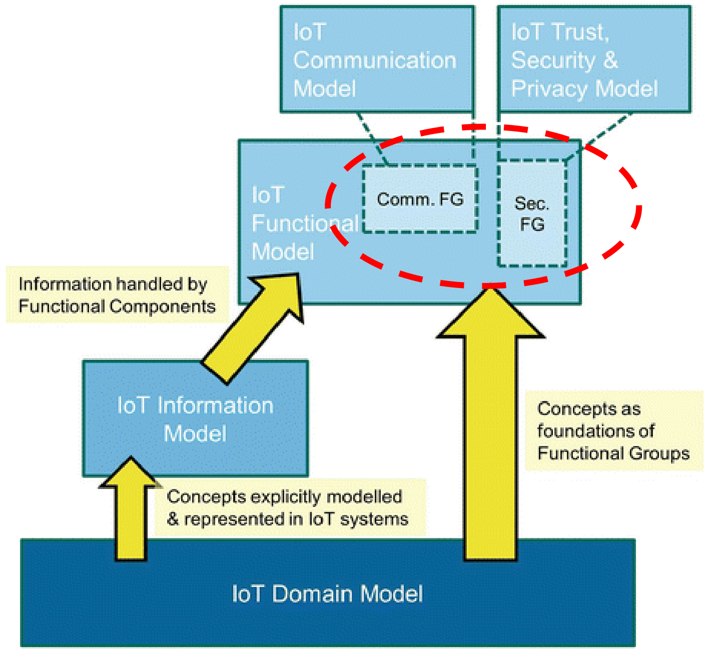


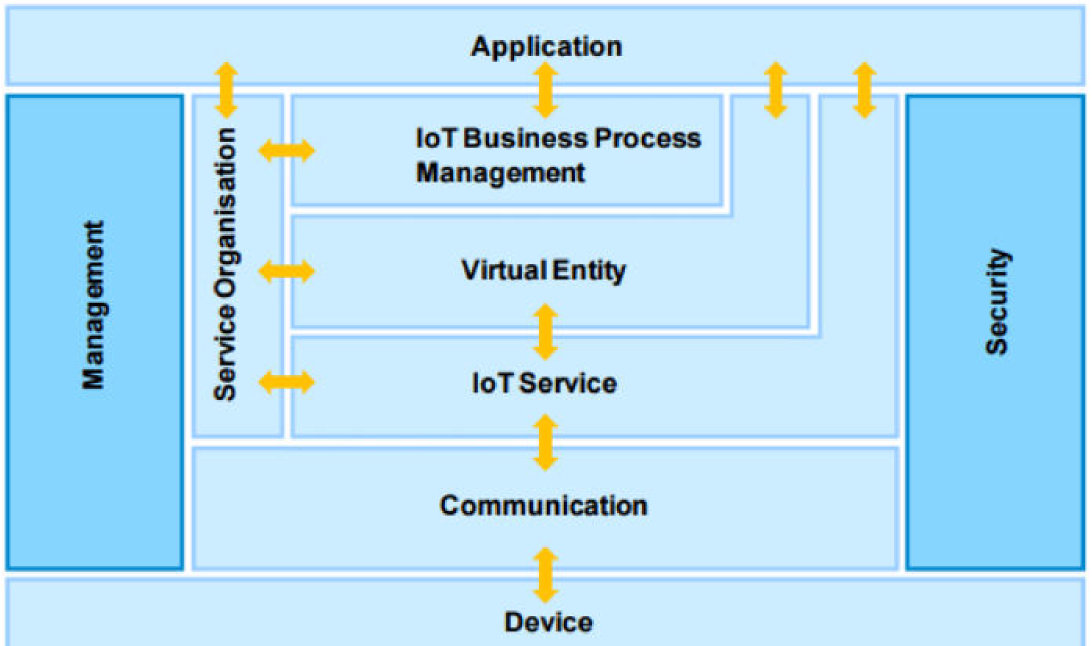
IoT Functional Model:

Draw and explain IoT functional model of IoT reference model 06 marks)

- The IoT Functional Model aims at describing mainly the **Functional Groups (FG)** and their **interaction** with the **ARM**, while the Functional View of a Reference Architecture describes the functional components of an FG, interfaces, and interactions between the components.
- The Functional View is typically derived from the Functional Model in conjunction with **high-level requirements.**
- Functional View describes the system's **runtime Functional Components**, their **responsibilities**, default functions, interfaces and primary interactions.
- The Functional View derives from the Functional Model and reflects the **developer's perspectives on the system**.

  • It will need to be extended with all identified and recommended) new profile-specific Functional Components including their interfaces and a list of Sequence Charts illustrating recommended usage of those components.


IoT Communication Model
IoT Trust, Security & Privacy Model
IoT Functional Model
Comm. FG
Sec. FG
Information handled by Functional Components
IoT Information Model
Concepts as foundations of Functional Groups
Concepts explicitly modelled & represented in IoT systems
IoT Domain Model



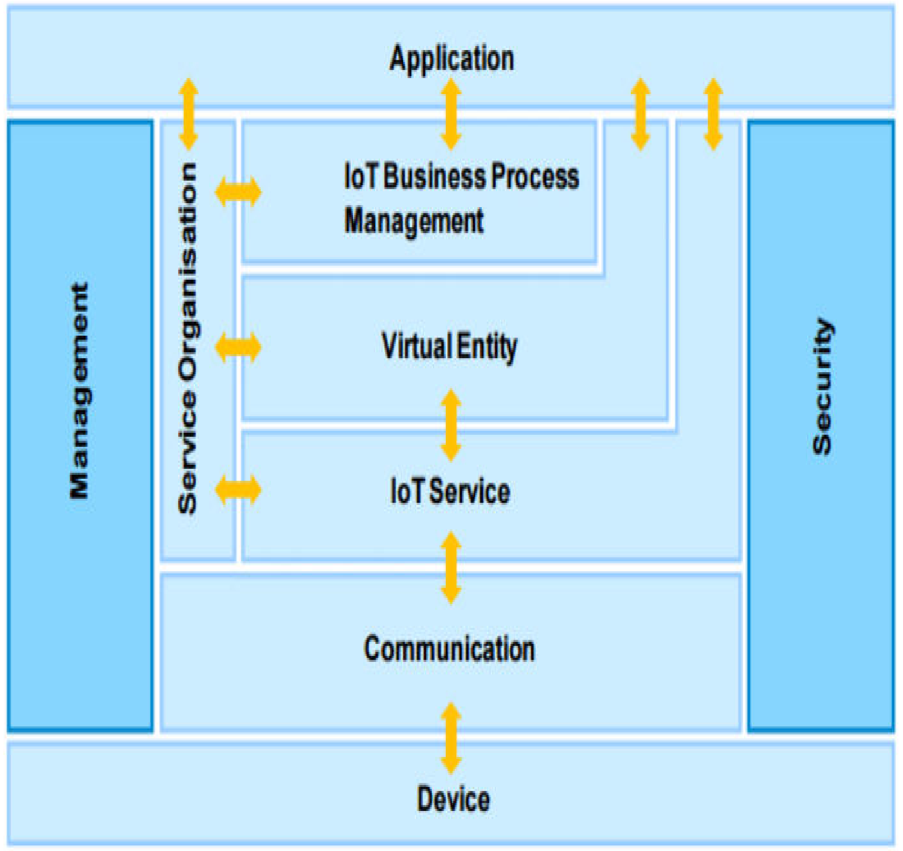
1. Device and Application functional group : Device functional components contains the sensing, actuation tag, processing and storage components. Application functional group contains standalone application.

2. Communication functional group:
Abstract all possible communication mechanisms used by relevant devices in an actual systems in order to transfer information to the digital world components or other devices.(hide all the socket or API)

3. IoT Service functional group

•The IoT Service FC is a collection of service implementations, which interface the related and associated Resources.

•For a Sensor type of a Resource, the IoT Service FC includes Services that receive requests from a User and returns the Sensor Resource value in synchronous or asynchronous (e.g. subscription/notification) fashion.
•The IoT Service Resolution FC contains the necessary functions to realize a directory of IoT Services that allows dynamic management of IoT Service descriptions and discovery/lookup/resolution of IoT Services by other Active Digital Artifacts.



<u>IoT Functional Model</u>

4. Virtual Entity functional group

- The Virtual Entity FG contains functions that support the interactions between Users and Physical Things through Virtual Entity services.
- An example of such an interaction is the query to an IoT system of the form, "**What is the temperature in the conference room Titan?"**

5. Process Management functional group

The IoT Process Management FG, integration of business processes with IoT-related services.
It consists of two FCs:

- The **Process Modelling FC** provides that right tools for modelling a business process that utilises IoT-related services.
- The **Process Execution FC** contains the execution environment of the process models created by the Process Modelling FC and executes the created processes by utilising the Service Organisation FG in order to resolve high-level application requirements to specific IoT services

6.Service Organization functional group

•The Service Composition FC manages the descriptions and execution environment of **complex services** consisting of simpler dependent services.
•An **example** of a complex composed service is a service offering **the average of the values coming from a number of simple Sensor Services**.
•The Service Orchestration FC resolves the requests coming from IoT Process Execution FC or User into the concrete IoT services that fulfill the requirements.

7.Security functional group

The Security FG contains the necessary functions for ensuring the security and privacy of an IoT system.

- The **Identity Management FC** manages the different identities of the involved Services or Users in an IoT system
- The **The Authentication FC** verifies the identity of a User and creates an assertion upon successful verification. verifies the identity of a User and creates an assertion upon successful verification.
- The **Authorization FC** manages and enforces access control policies. It provides services to manage policies (CUD), as well as taking decisions and enforcing them regarding access rights of restricted resources.
- The **Key Exchange & Management** is used for setting up the necessary security keys between two communicating entities in an IoT system.
- The **Trust & Reputation FC** manages reputation scores of different interacting entities in an IoT system and calculates the service trust levels.

8. Management functional group

- The Configuration FC maintains the configuration of the FCs and the Devices in an IoT system. The component collects the current configuration of all the FCs and devices, stores it in a historical database, and compares current and historical configurations.

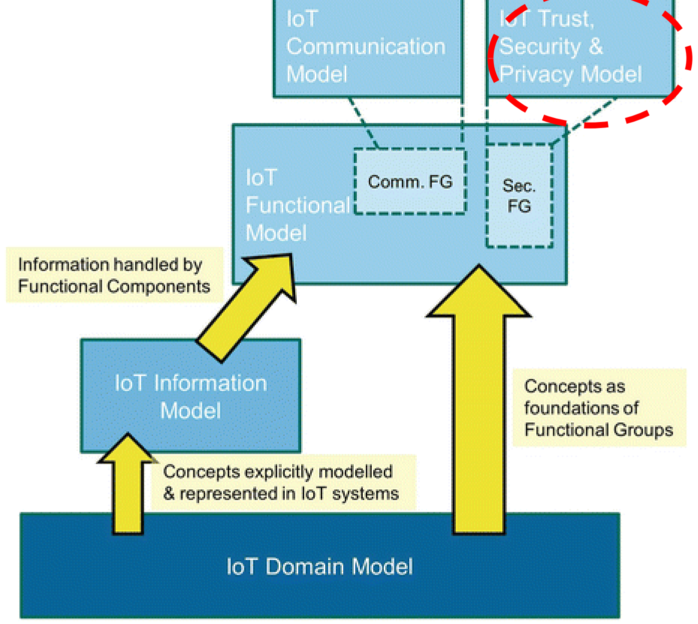
- The Fault FC detects, logs, isolates, and corrects system-wide faults if possible. This means that individual component fault reporting triggers fault diagnosis and fault recovery procedures in the Fault FC.The component collects the current configuration of all the FCs and devices, stores it in a historical database, and compares current and historical configurations.

- The Member FC manages membership information about the relevant entities in an IoT system.

- The State FC is similar to the Configuration FC, and collects and logs state information from the current FCs, which can be used for fault diagnosis, performance analysis and prediction, as well as billing purposes.

- The Reporting FC is responsible for producing compressed reports about the system state based on input from FCs. ii-font-family

IOT Trust,Security & privacy model:


IoT Communication Model
IoT Trust, Security & Privacy Model
IoT Functional Model
Comm. FG
Sec. FG
Information handled by Functional Components
IoT Information Model
Concepts as foundations of Functional Groups
Concepts explicitly modelled & represented in IoT systems
IoT Domain Model

Safety

the IoT Reference Model can only provide IoT-related guidelines for ensuring a safe system to the extent possible and controllable by a sys- tem designer.
Eg: smart grid.

Privacy

Because interactions with the physical world may often include humans, protecting the User privacy is of utmost importance for an IoT system. The IoT-A Privacy Model depends on the following functional components: Identity Management, Authentication, Authorisation, and Trust & Reputation

Trust

Generally, an entity is said to 'trust' a second entity when the first entity makes the assumption that the second entity will behave exactly as the first entity expects."

Security

The Security Model for IoT consists of communication security that focuses mostly on the confidentiality and integrity protection of interacting entities and functional components such as Identity Management, Authentication, Authorisation, and Trust & Reputation.

Reference architecture

Purpose of Using a Reference Architecture: 04 marks

- An IoT reference architecture serves as a **foundational blueprint** that **outlines** the **essential components** and **interactions** within an IoT system.
- It provides a **solid starting point** for **designing** and **implementing** IoT solutions.
- A reference architecture serves as a standardized blueprint that provides a clear structure and guidelines for designing and implementing an IoT system.
- It enables **consistency**, promotes **best practices**, and facilitates **communication** and **collaboration** among **stakeholders**.
- By leveraging a reference architecture, developers can **reduce design complexity**, ensure **interoperability**, and **accelerate** the **development process**, ultimately leading to more **efficient** and **reliable** IoT solutions.

Benefits of Using IoT Reference Architecture

04 marks

1. **Common Framework:** IoT reference architecture provides a standardized framework for designing and implementing IoT solutions, ensuring consistency and interoperability across systems.

2. **Security and Scalability:** The architecture serves as a foundation for implementing robust security and scalability measures, safeguarding IoT systems against threats and enabling future growth.

3. **Cost Efficiency:** By leveraging a reference architecture, organizations can avoid reinventing the wheel and utilize existing technologies and expertise, reducing the cost of development and deployment.

4. **Faster Time to Market:** Utilizing a reference architecture accelerates the implementation of IoT solutions, enabling organizations to get their systems up and running more quickly and efficiently.

The Reference Architecture

Draw and explain reference architecture - 06 marks

- IoT **reference architectures** typically consist of **multiple layers** that work together to enable the functioning of an IoT system.
- While the specific layering may vary based on different frameworks or standards, a commonly used layered structure includes the following:

1. **Perception Layer:** This layer comprises the physical devices or sensors that collect data from the environment or interact with the physical world. These devices can include temperature sensors, motion detectors, cameras, and other IoT–enabled devices.
2. **Network Layer:** The network layer facilitates the connectivity and communication between the IoT devices and the cloud or other data processing components. It includes protocols, gateways, routers, and other networking infrastructure to ensure seamless data transfer and reliable connections.
3. **Data Processing Layer:** This layer involves processing and analyzing the data collected from IoT devices. It may include edge computing devices or cloud–based platforms where data is aggregated, filtered, transformed, and analyzed to derive valuable insights.
4. **Application Layer:** The application layer encompasses the software applications or services that utilize the processed IoT data to provide specific functionalities or address specific use cases. These applications can range from real–time monitoring and control systems to predictive analytics, machine learning algorithms, and automation

Example 1

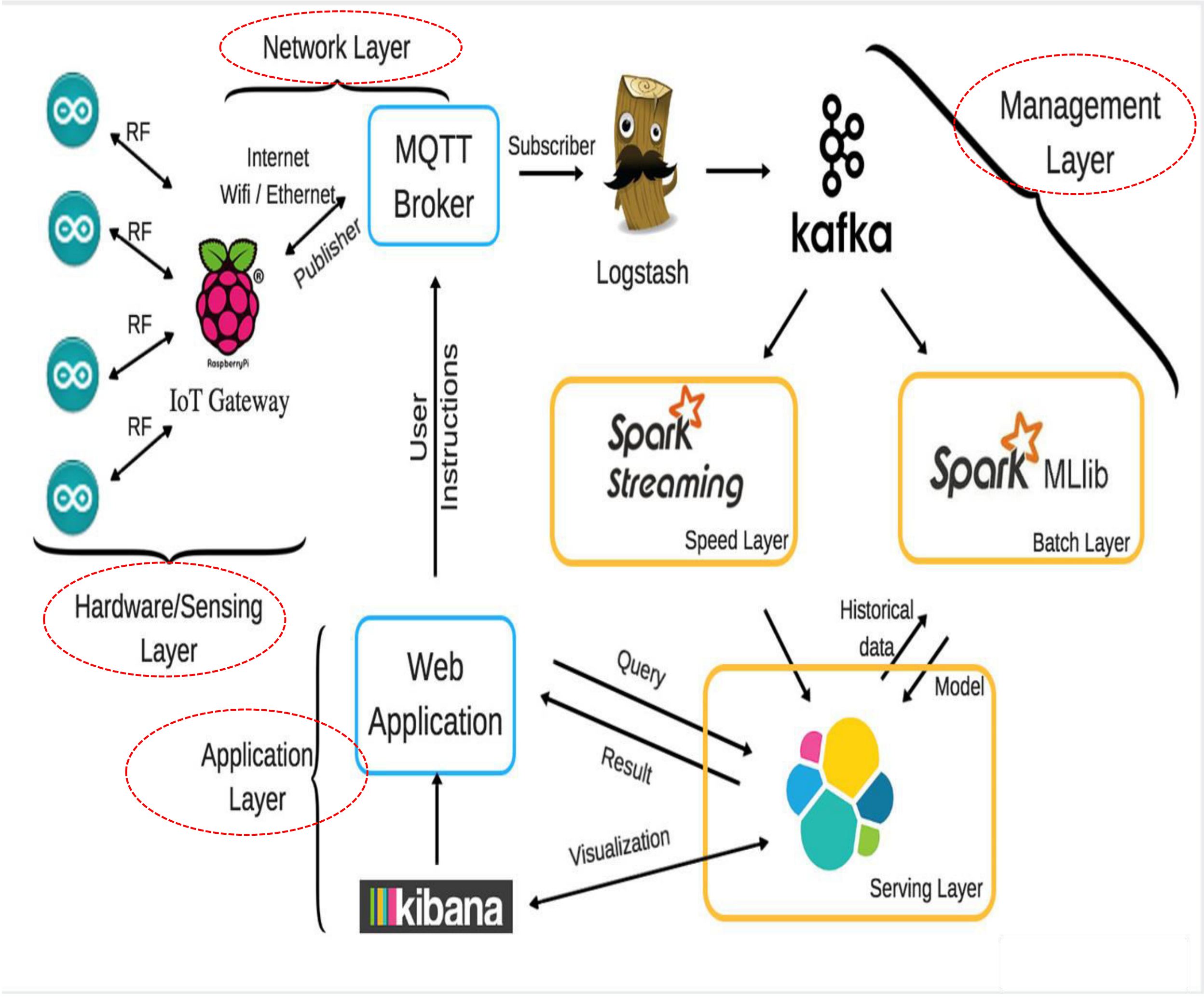


Fig 1 : Reference architecture for IoT

Example 2

The layers are :

- **Client/external communications -** Web/Portal, Dashboard, APIs
- **Event processing and analytics -** (including data storage)
- **Aggregation/bus layer** – ESB and message broker
- **Relevant transports** - MQTT/HTTP/XMPP/CoAP/AMQP, etc.
- **Devices**

The cross-cutting layers are

•**Device manager**
•**Identity and access management**

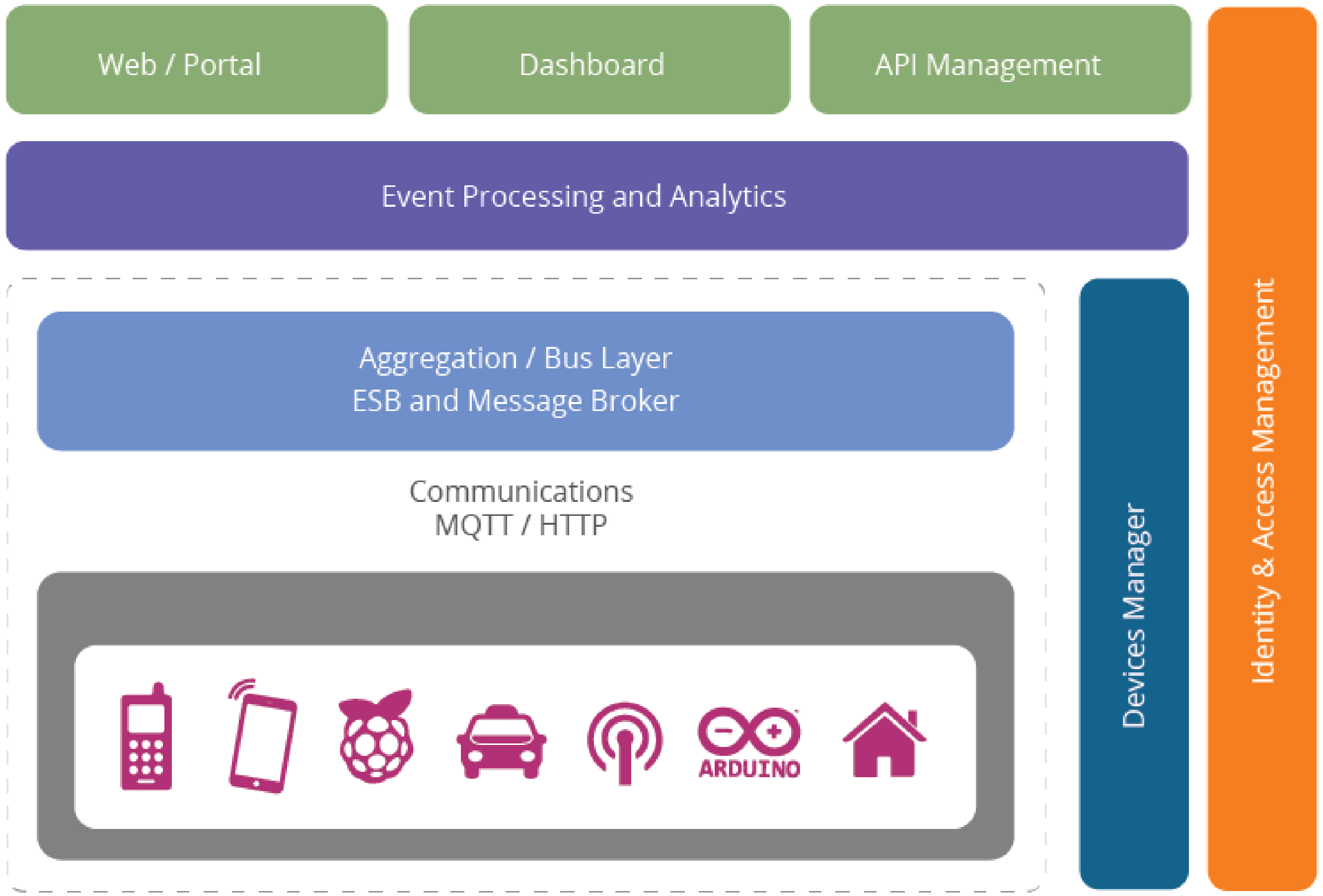


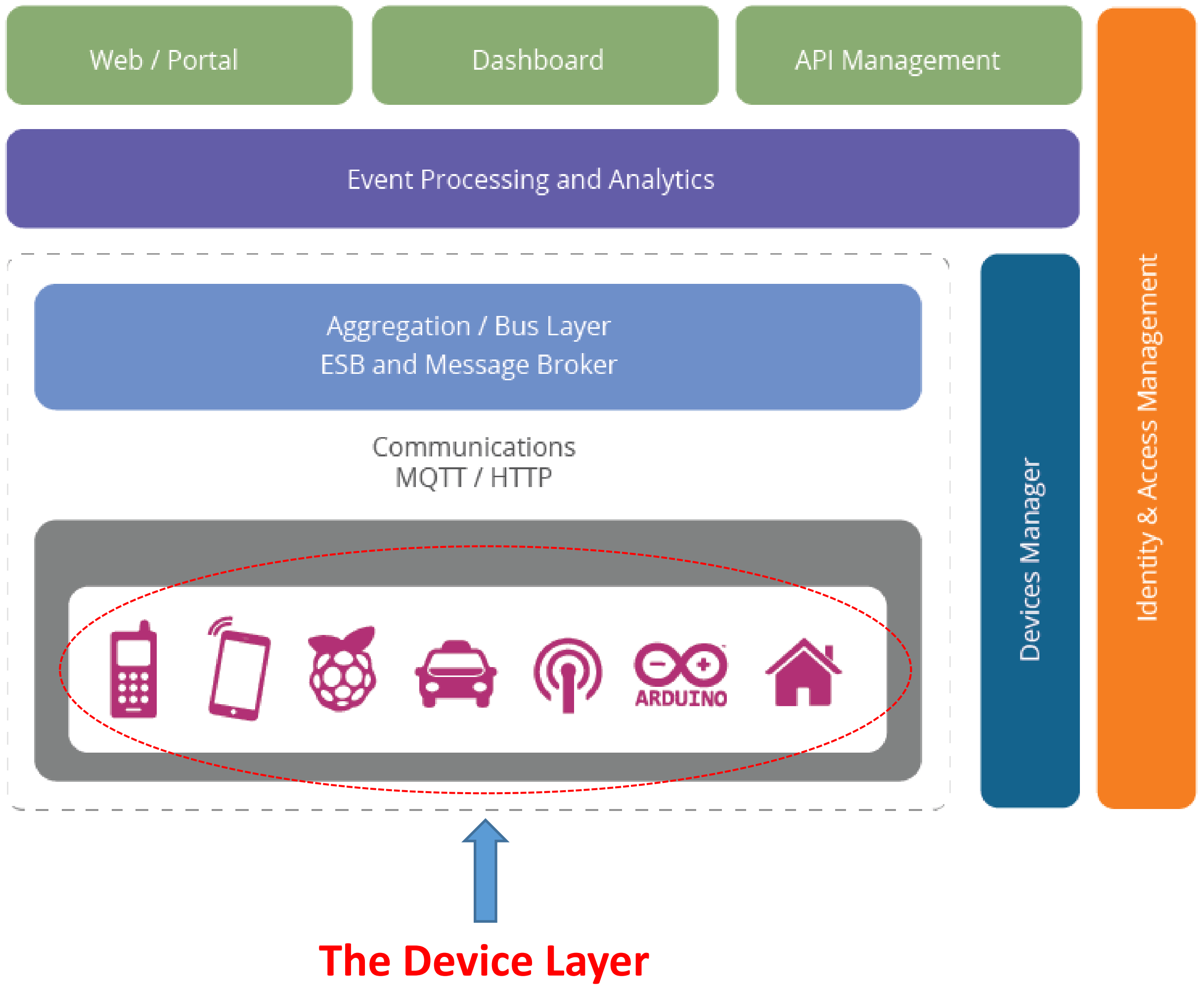
Fig 2 : Reference architecture for IoT

The Device Layer

The bottom layer of the architecture is the device layer. Devices can be of various types, but in order to be considered as IoT devices, they must have some communications that either indirectly or directly attaches to the Internet.

Examples of direct connections are :

- Arduino with Arduino Ethernet connection
- Arduino Yun with a Wi-Fi connection
- Raspberry Pi connected via Ethernet or Wi-Fi
- Intel Galileo connected via Ethernet or Wi-Fi Examples of indirectly connected device include
- ZigBee devices connected via a ZigBee gateway
- Bluetooth or Bluetooth Low Energy devices connecting via a mobile phone
- Devices communicating via low power radios to a Raspberry Pi

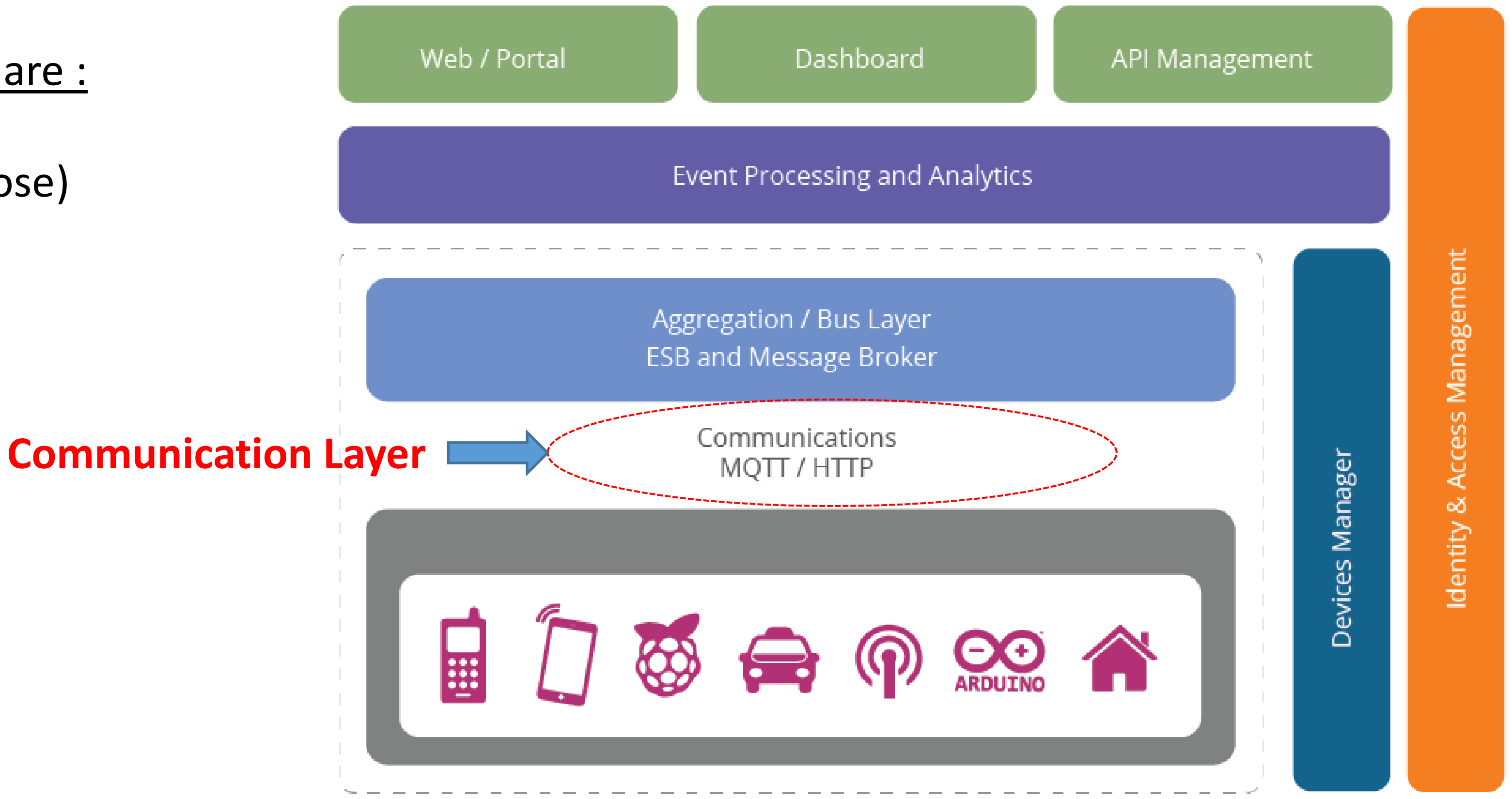


The Communication Layer

The communication layer supports the connectivity of the devices. There are multiple potential protocols for communication between the devices and the cloud.

The most well known three potential protocols are :

- HTTP/HTTPS (and RESTful approaches on those)
- MQTT 3.1/3.1.1
- Constrained application protocol (CoAP)

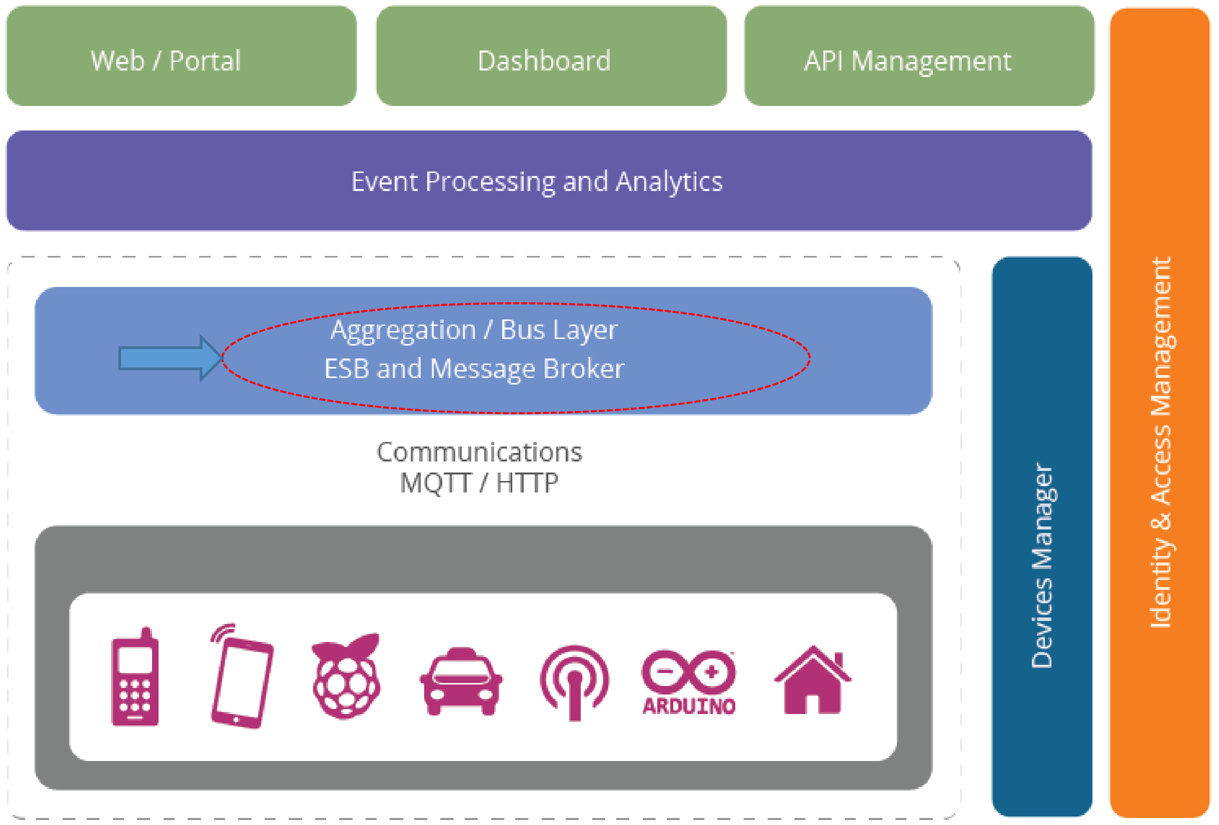


The Aggregation/Bus Layer

An important layer of the architecture is the layer that aggregates and brokers communications.

This is an important layer for three reasons:

1. The ability to support an HTTP server and/or an MQTT broker to talk to the devices
2. The ability to aggregate and combine communications from different devices and to route communications to a specific device (possibly via a gateway)
3. The ability to bridge and transform between different protocols, e.g. to offer HTTP based APIs that are mediated into an MQTT message going to the device.

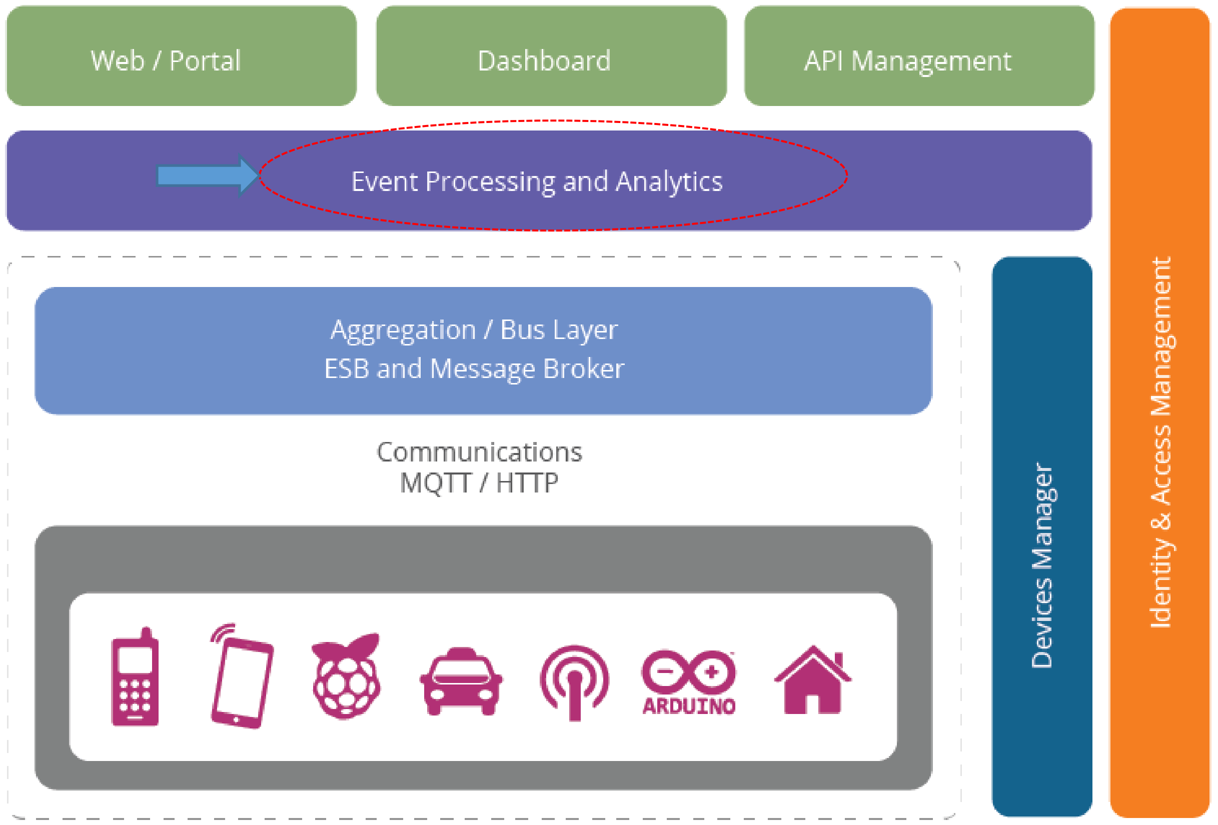


The Event Processing and Analytics Layer

This layer takes the events from the bus and provides the ability to process and act upon these events. A core capability here is the requirement to store the data into a database

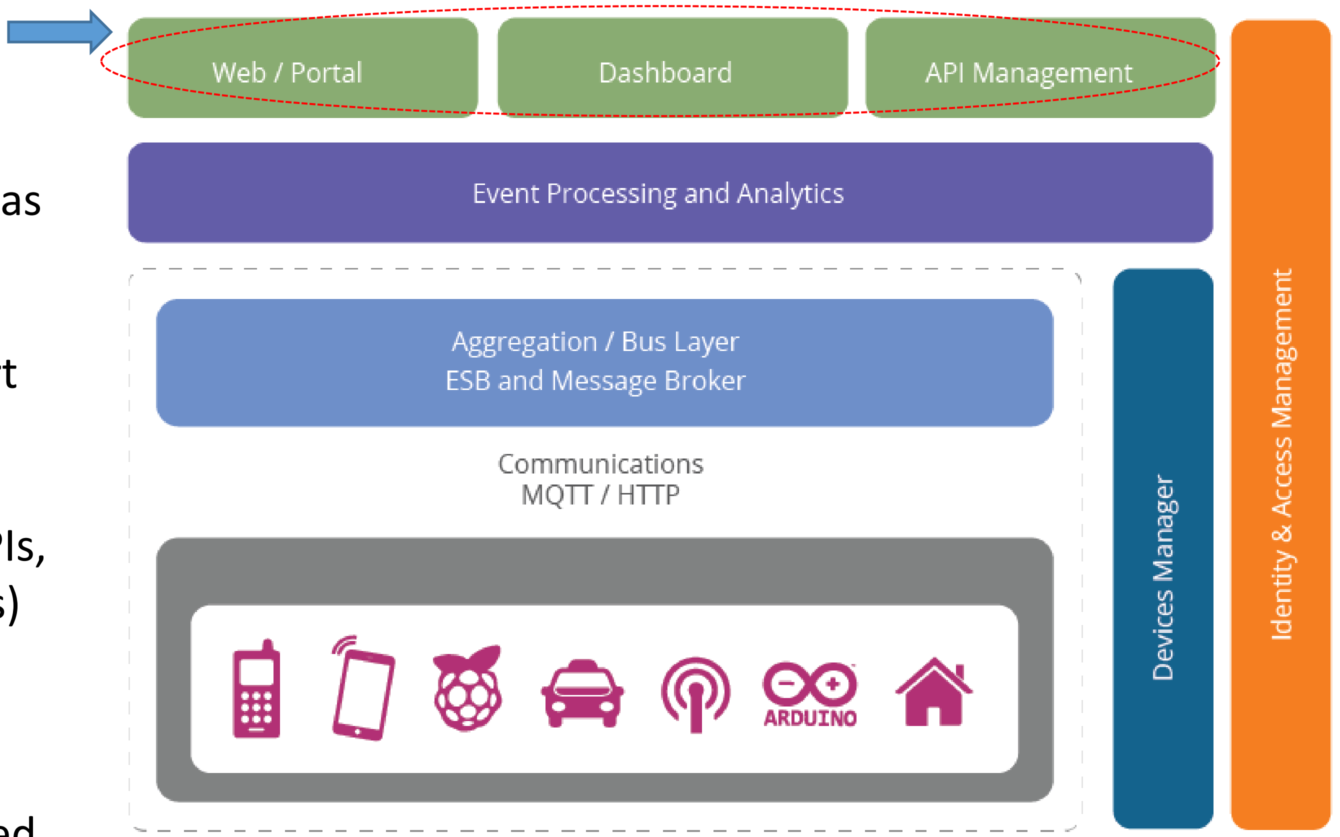
Our recommended approach in this space is to use the following approaches:

1. Highly scalable, column-based data storage for storing events
2. Map-reduce for long-running batch-oriented processing of data
3. Complex event processing for fast in-memory processing and near real-time reaction and autonomic actions based on the data and activity of devices and other systems
4. In addition, this layer may support traditional application processing platforms, such as Java Beans, JAX-RS logic, message-driven beans, or alternatives, such as node.js, PHP, Ruby or Python.



Client/External Communications Layer

The reference architecture needs to provide a way for these devices to communicate outside of the device-oriented system. This includes three main approaches. Firstly, we need the ability to create web-based front-ends and portals that interact with devices and with the event-processing layer. Secondly, we need the ability to create dashboards that offer views into analytics and event processing.



The API management layer provides three main functions:

1. The first is that it provides a developer-focused portal (as opposed to the userfocused portal previously mentioned), where developers can find, explore, and subscribe to APIs from the system. There is also support for publishers to create, version, and manage the available and published APIs;
2. The second is a gateway that manages access to the APIs, performing access control checks (for external requests) as well as throttling usage based on policies. It also performs routing and load-balancing;
3. The final aspect is that the gateway publishes data into the analytics layer where it is stored as well as processed to provide insights into how the APIs are used.

Machine to Machine (M2M) Architecture

Introduction to M2M

Define the role of M2M (03 Marks)

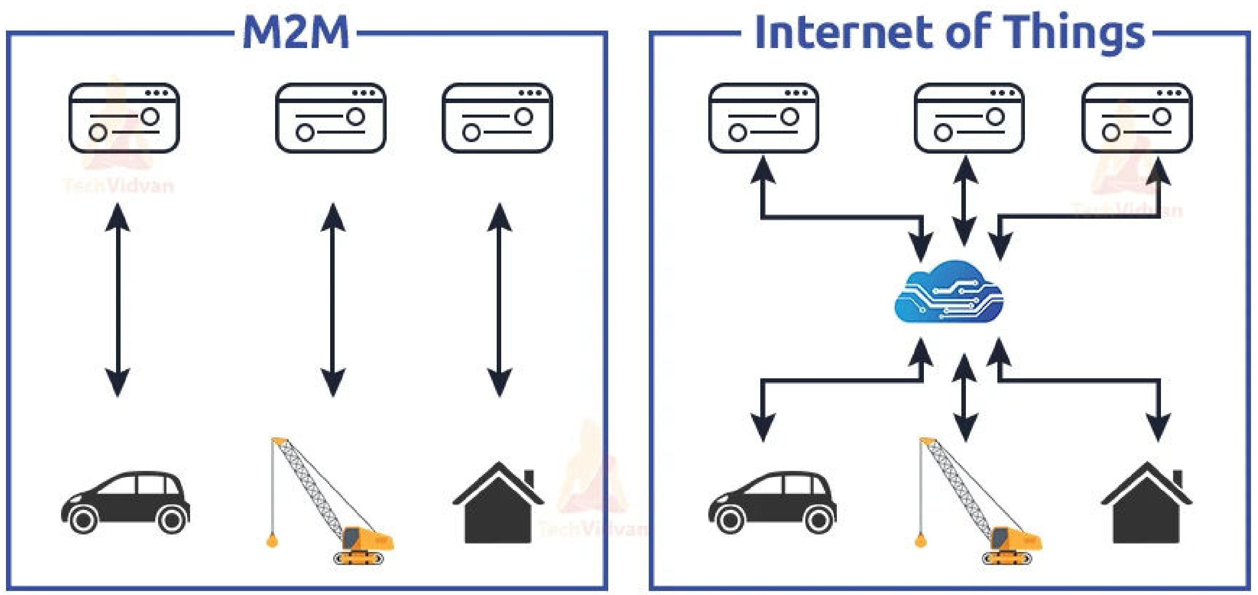
- This is commonly known as **Machine to machine** communication.
- It is a concept where two or more than two **machines communicate** with each other **without human interaction** using a **wired** or **wireless** mechanism.
- M2M is an technology that helps the devices to **connect** between **devices without** using **internet**.
- M2M communications offer several applications such as **security**, **tracking**, **manufacturing** and facility management.
- M2M is also named as **Machine Type Communication (MTC)** in **3GPP** (3rd Generation Partnership Project).
- M2M is communication could carried over **mobile networks**, for ex- **GSM-GPRS**, **CDMA EVDO** Networks .

Applications of M2M :

List and define any 4 applications of M2M (04 Marks)

- **Security :** Surveillances, Alarm systems, Access control, Car/driver security
- **Tracking & Tracing :** Fleet Management, Order Management, Pay as you drive, Asset Tracking, Navigation, Traffic information, Road tolling, Traffic optimization/steering
- **Payment :** Point of sales, Vending machines, Gaming machines
- **Health :** Monitoring vital signs, Supporting the aged or handicapped, Web Access Telemedicine points, Remote diagnostics
- **Remote Maintenance/Control :** Sensors, Lighting, Pumps, Valves, Elevator control, Vending machine control, Vehicle diagnostics
- **Metering :** Power, Gas, Water, Heating, Grid control, Industrial metering
- **Manufacturing :** Production chain monitoring and automation
- **Facility Management :** Home / building / campus automation

Difference Between M2M and IoT



- M2M and Iot are **similar** but they do **not** exactly **perform** the **same functionality**.
- IoT **depends** on M2M, but this is not the same vice-versa.
- Both terms supply device communication but M2M does not need to depend and is single network equipment.
- Iot on the other hand takes M2M on another level by connecting larger systems on a network that is private or public.
- M2M uses communication between machines such as sensors and hardware from one to another whereas **IoT uses an IP-network** to send and receive data gathered from devices. These devices send the data to large clouds via IoT gateways.

Difference between IoT and M2M : 4/6 Marks

Sr.No.	Parameters	IoT	M2M
1	Connection type used	The connection is via Network and using various communication types.	The connection is a point to point
2	Communication protocol used	Internet protocols are used such as HTTP, FTP, and Telnet.	Traditional protocols and communication technology techniques are used
3	Data Sharing	Data is shared between other applications that are used to improve the end-user experience.	Data is shared with only the communicating parties.
4	Internet	Internet connection is required for communication	Devices are not dependent on the Internet.
5	Type of Communication	It supports cloud communication	It supports point-to-point communication.
6	Computer System	Involves the usage of both Hardware and Software.	Mostly hardware-based technology
7	Scope	A large number of devices yet scope is large.	Limited Scope for devices.
8	Open API support	Supports Open API integrations.	There is no support for Open APIs
9	Centric	Information and service centric	Communication and device centric.
10	Approach used	Horizontal enabler approach	Vertical system solution approach .
11	Components	Devices/sensors, connectivity, data processing, user interface	Device, area networks, gateway, Application server.
12	Examples	Smart wearables, Big Data and Cloud, etc.	Sensors, Data and Information, etc.

Architecture of M2M

Draw and explain architecture of M2M (6 Marks)

- M2M enables machines to communicate with other machines and pass small amounts of information.
- Some examples could include smoke detectors that detect smoke and send the information to various other digital devices.

The three main domains of M2M architecture are:

M2M application

- As the name suggests, the M2M application domain offers applications to use M2M technology conveniently.
- Examples include server and end-user applications.

M2M network domain

- M2M network domain acts as a bridge between the M2M application domain and the M2M device domain.
- It is made of two parts called the M2M core and M2M service capabilities.

M2M device domain

- M2M device domain contains all the devices that can connect to the M2M network easily.
- The device domain can also be called the M2M area network. The M2M device domain includes devices that can connect directly over a network, devices that cannot directly connect to a network and may perhaps require an M2M gateway and proprietary devices.

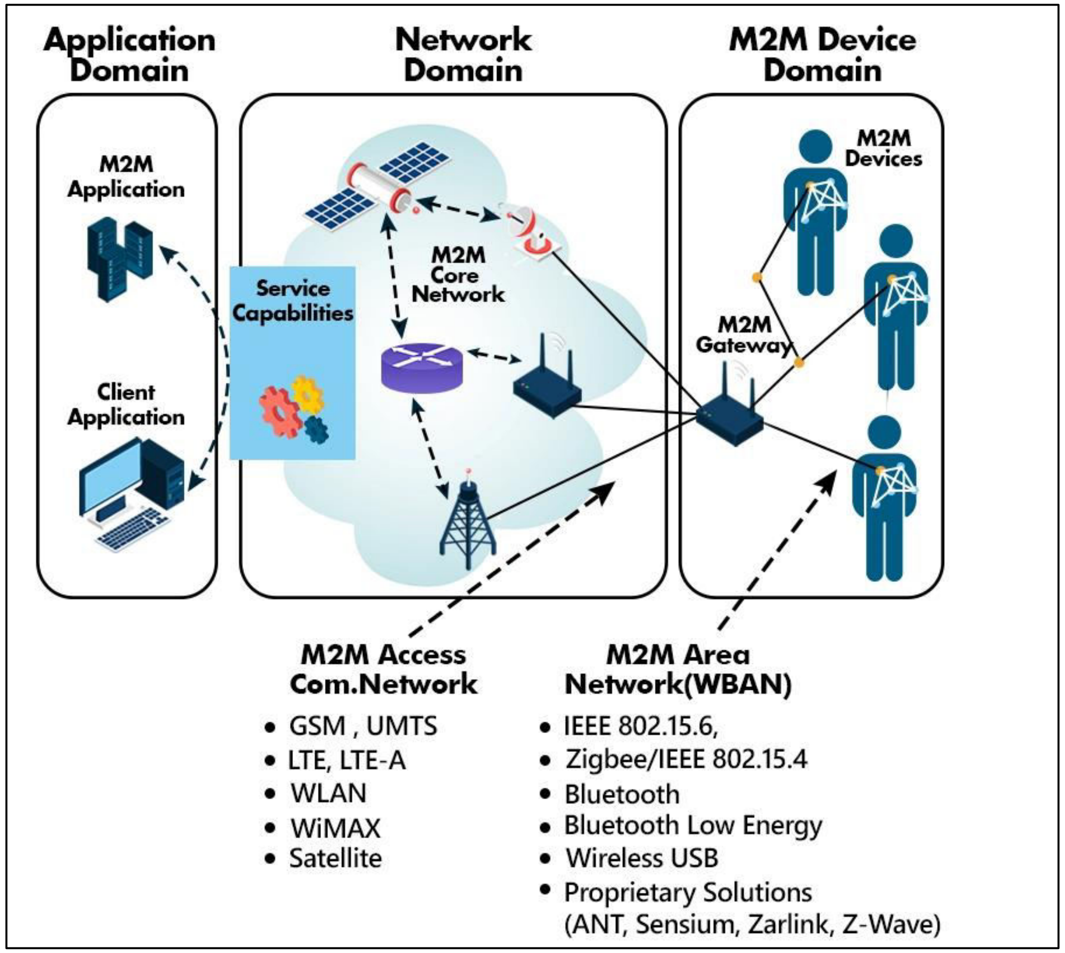


Fig : Architecture of M2M

Working of M2M

- M2M devices send data across a network by sensing information. In order to send the data the machines use **public networks** such as **cellular** and **ethernet**.
- M2M comprises components such as RFID, sensors, Wi-Fi communication links and automated computer software programs that translate the incoming data to generate responses or actions.
- **Telemetry** is one of the most renowned M2M communication.
- It has been in use since the beginning of the last century to transmit data.
- Developers used **telephone lines** for **communication** and later moved to **radio wave transmission** signals in order to monitor the performance of the data that is gathered from remote locations.
- The arrival of the internet improved the standards of wireless technology and now wireless communication is used in everyday real life applications such as hospitals, cities, stations, roads and so on.

Make a choice between IoT and M2M 03 Marks

M2M is a better option for :

- If you want your devices to have a **point to point communication**
- If you want **quick communication** between a **few machines**
- When you need your devices to work even **without** an **internet connection**
- If you are **not concerned** about **scalability**

IoT is a better choice for :

- If your devices need **constant syncing**.
- If you want your devices to **share information** between each other **constantly**.
- When you require **high scalability** and **better performance** in the management of devices.