UNIT - II IOT and M2M:

M2M to IoT – A Basic Perspective– Introduction, Differences and similarities between M2M and IoT, SDN and NFV for IoT,M2M Value Chains, IoT Value Chains, An emerging industrial structure for IoT, international driven global value chain and global information monopolies.

IOT Architecture: IoT Architecture components, Comparing IoT Architectures, A simplified IoT Architecture, core IoT functional stack, IoT data management and compute stack

M2M to IoT – A Basic Perspective

- Machine to Machine refers to networking of machines (or devices) for the purpose of remote monitoring and control and data exchange.
- In M2M communication, objects can talk to each other without human intervention.
- Today there is insufficient integration of real physical world with virtual world.
- Today human translates physical world information in to digital world information to use computers.
- Machine to machine refers to the process of communication of a physical object or device at machine with others of the same type, mostly for monitoring but also for control purposes.
- Each machine in an M2M system embeds a smart device. The device senses the data or status of the machine, and performs the computation and communication functions. A device communicates via wires or wireless systems.

M2M System architecture:

- M2M architecture consists of three domains
- 1.M2M device domain.
- 2. M2M network domain.
- 3.M2M application domain

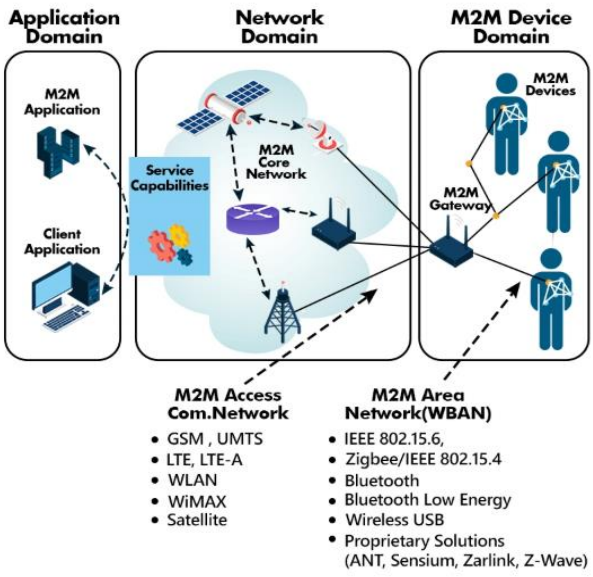


Figure 2.1: M2M System Architecture

Three-domain ETSI reference model:

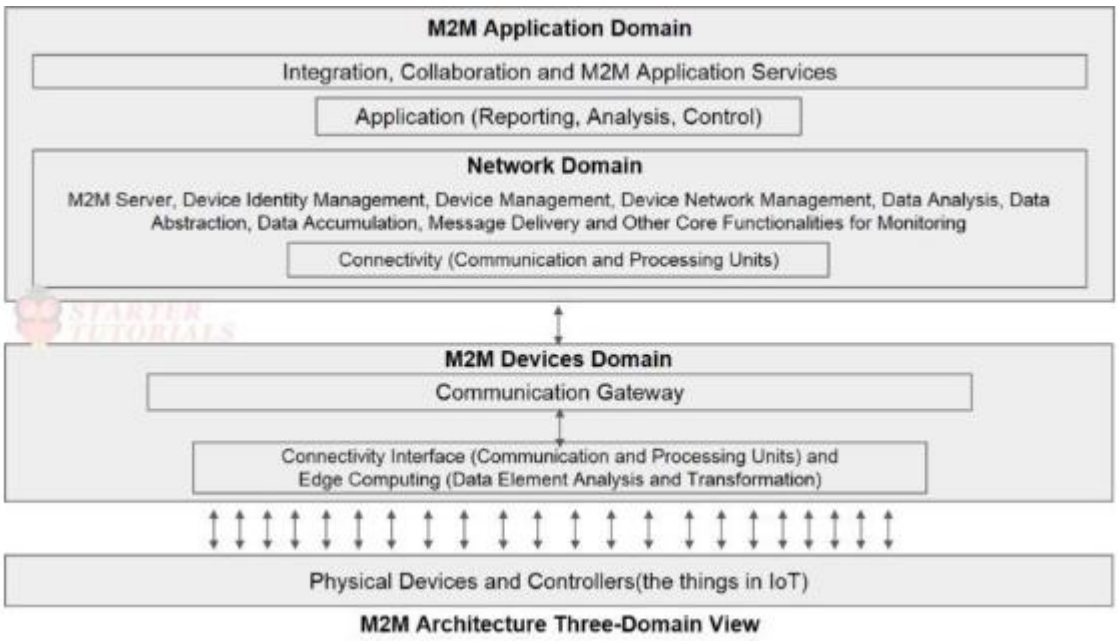


Figure 2.2 : Three-domain ETSI reference model

System Architecture :

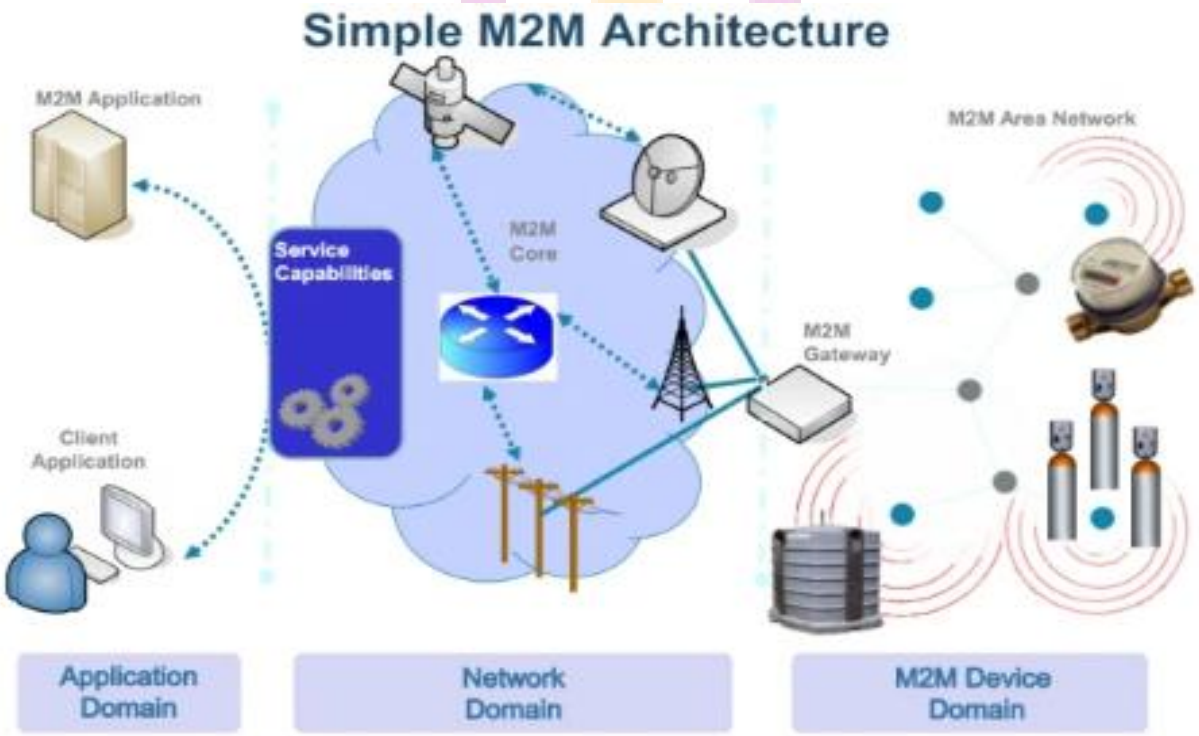


Figure 2.3: System Architecture

Communication Protocols:

Protocols types :

1.LWM2M

2.MQTT

3.XMPP

4.6LowPAN

Note: Each communication device is assigned 48-bits IPv6 address.

M2M gateway:

- Since non-IP based protocols are used within M2M area networks, the M2M nodes within one network cannot communicate with nodes in an external network.
- To enables the communication between remote M2M area networks, M2M gateways are used.
- M2M gateway performs protocol translations to enable IP-connectivity for M2M area networks.
- M2M gateway acts as a proxy performing translation from /to native protocols to/from internet protocol(IP)

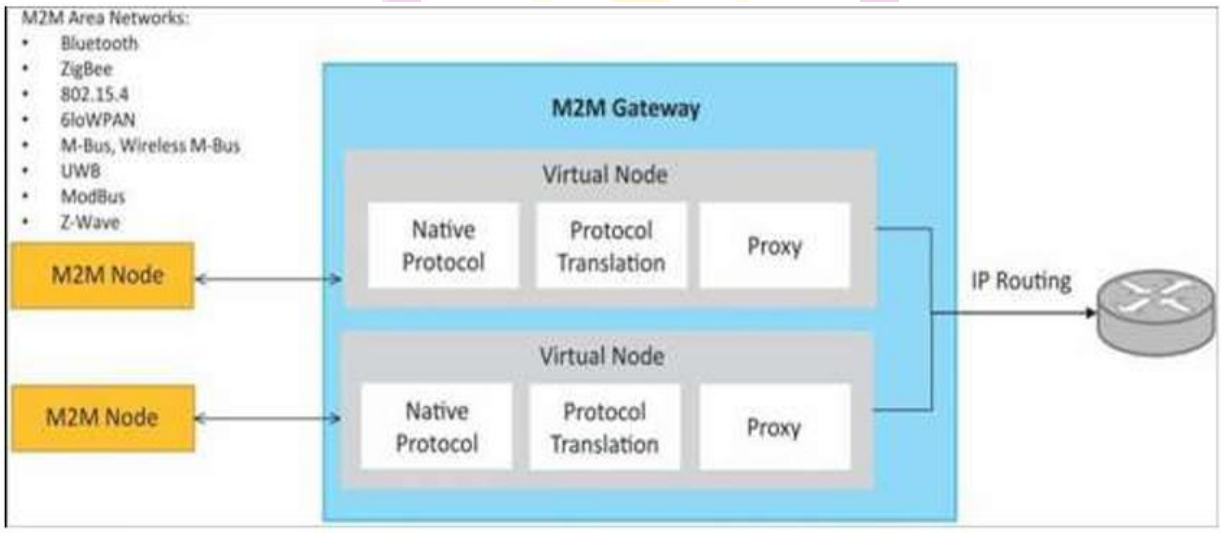


Figure 2.4 : M2M Gateway

Working Process:

Figure 2.5 : Vending Machine

Example : Vending Machine:

- One can buy soft drinks, flowers, etc from vending machine in self service manner
- Once the vending machine detects the item in out-of-stock, it sends message to order management server through 3G/4G communication link which further send information to vendor
- The vendor re-stocks the vending machine
- Vending machine stores daily sales data in internal database and sends information to vendor
- Vendor will know which product has been sold and the total daily revenue
- Companies and organizations depend on M2M machines as they use less energy such as oil and gas. M2M machines can also bill users such as smart metres. M2M machines can be used in factories to detect conditions such as pressure, temperature, and equipment status.
- In telemedicine, M2M allows remote check up on patients. It allows dispensing medicine and allows doctors to track the health status of patients.
- M2M also contributes heavily to financial activities to allow different purchasing options and examples include Google Wallet and Apple Pay.
- Smart home systems incorporate M2M communication. It allows devices and home appliances to communicate with each and send message over a network
- Finally, M2M also plays a huge part in robotics, traffic management, remote-control software, logistics, fleet management, and automation.

Difference between M2M and IoT :

M2M is a point to point to communication. Incase of IoT , devices are always connected to internet either using wired and wireless internet. The connectivity to internet is for processing data and delivering it through a middle layer which is hosted in the cloud.

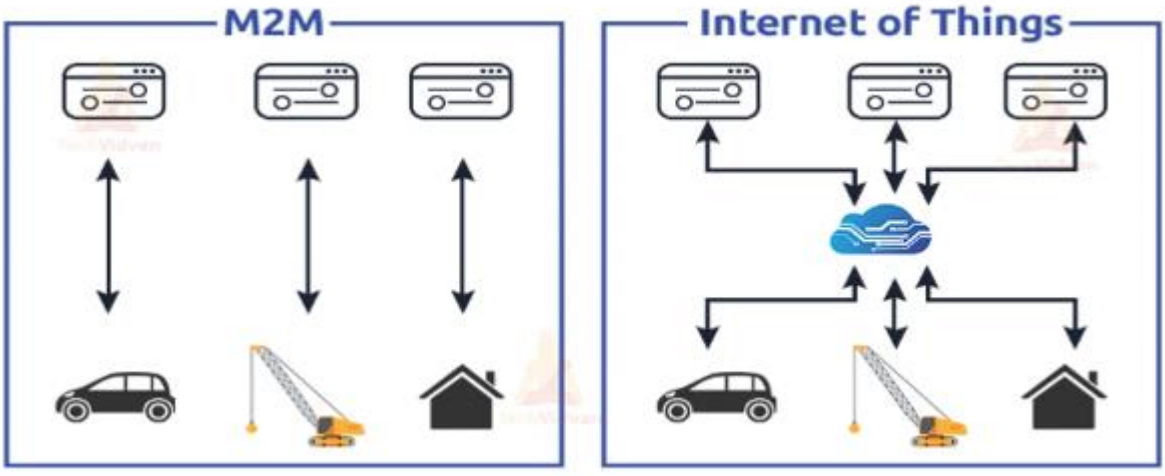


Figure 2.6 : Difference between the M2M and IoT

M2M	IoT
Communication protocols	
Primarily M2M uses point – to –point communication . Devices in M2M send and receive information directly from each other and generally , devices in M2M communication are not connect to internet.	Incase of IoT , device are always connected to internet either using wired or wireless internet.
Primarily communication in Non-IP based	Primarily communication in IoT is IP based.
The focus of communication in M2M is usually on the protocols below the network layer such as ZigBee , Bluetooth, 6LoWPAN, IEEE 802.15.4,Z-Wave…	The focus of communication in IoT is usually on the protocols above the network layer such as HTTP, CoAP, Websockets, MQTT, XMPP, DDS, AMQP….
Machines in M2M vs Things in IoT	
M2M involves homogenous machines within an M2M area network.	IoT involves heterogenous machines within an IoT network.
Hardware and Software Emphasis	
The emphasis of M2M is more on hardware with embedded modules.	The emphasis of IoT is more on software which are used for sensor data collection , data analysis and interfacing with the cloud through IP-based communication.
Data Collection & Analysis	
M2M data is collected in point solution and often in on-premises strong infrastructure.	In contrast to M2M , the data in IoT is collected in the cloud (can be public . Private or hybrid)
Application	
M2M data is collected in point solution and can be accessed by on – premises applications such as diagnosis applications service management applications and on premises enterprise applications.	IoT data is collected in the cloud and can be accessed by cloud applications such as analytics application, enterprise application , remote diagnosis and management application etc…

Software Defined Networking (SDN):

- Traditional networking vs SDN
- SDN basics
- SDN Architecture
- SDN benefits

Traditional networking :

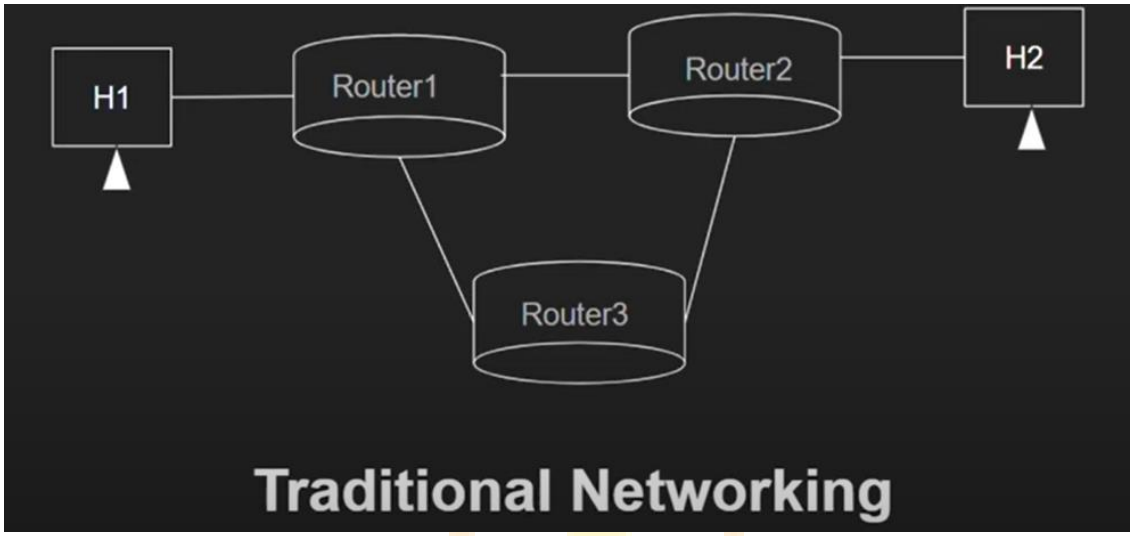


Figure 2.7 : Traditional Networking

Software Defined Networking (SDN) :

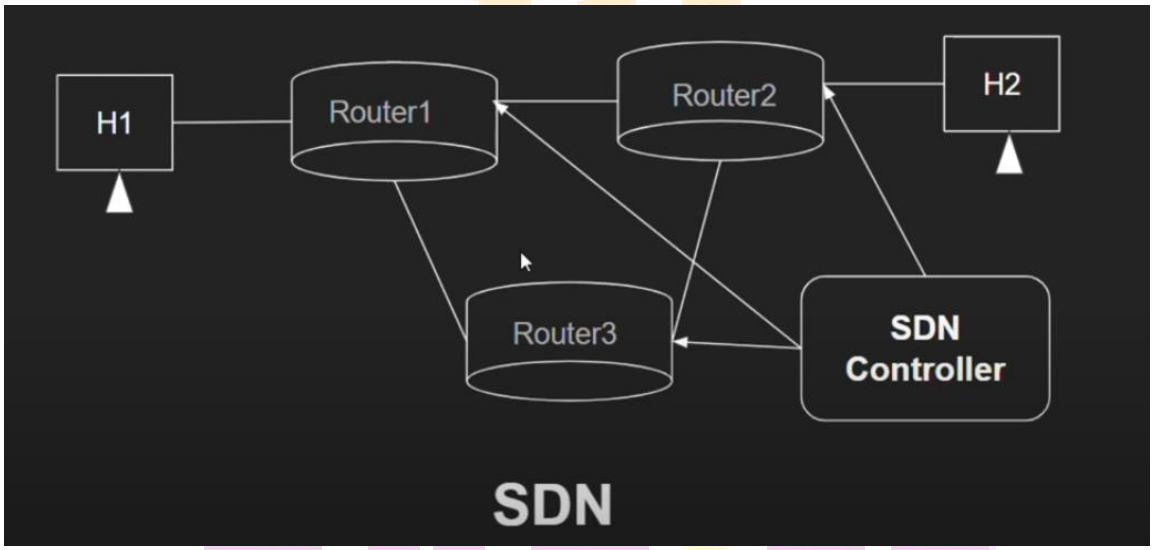


Figure 2.8: Software Defined Networking (SDN)

Software Defined Networking (SDN) Basics:

Software Defined Networking (SDN) simplifies data communication in modern networks such as IoT networks, cloud computing environments, computer networks, and Network Function Virtualization (NFV) systems. Traditional networking devices like routers consist of two main components: the **control plane** and the **data forwarding plane**. The control plane is responsible for performing computational tasks such as routing decisions, while the data forwarding plane handles the actual transfer of data packets.

In conventional networks, both these planes reside within the same network device, making network management complex and less flexible. SDN overcomes this limitation by separating the control plane from the data forwarding plane. The control plane functionality is removed from individual routers and centralized in an **SDN controller**.

The SDN controller is responsible for making routing decisions and managing network intelligence centrally.

Network devices then focus only on forwarding data packets based on the rules provided by the controller. This centralized approach allows SDN controllers to maintain a unified and global view of the entire network. As a result, SDN simplifies network configuration, management, and provisioning, improves flexibility, enhances scalability, and enables efficient control over complex network infrastructures.

Software Defined Networking (SDN) Architecture:

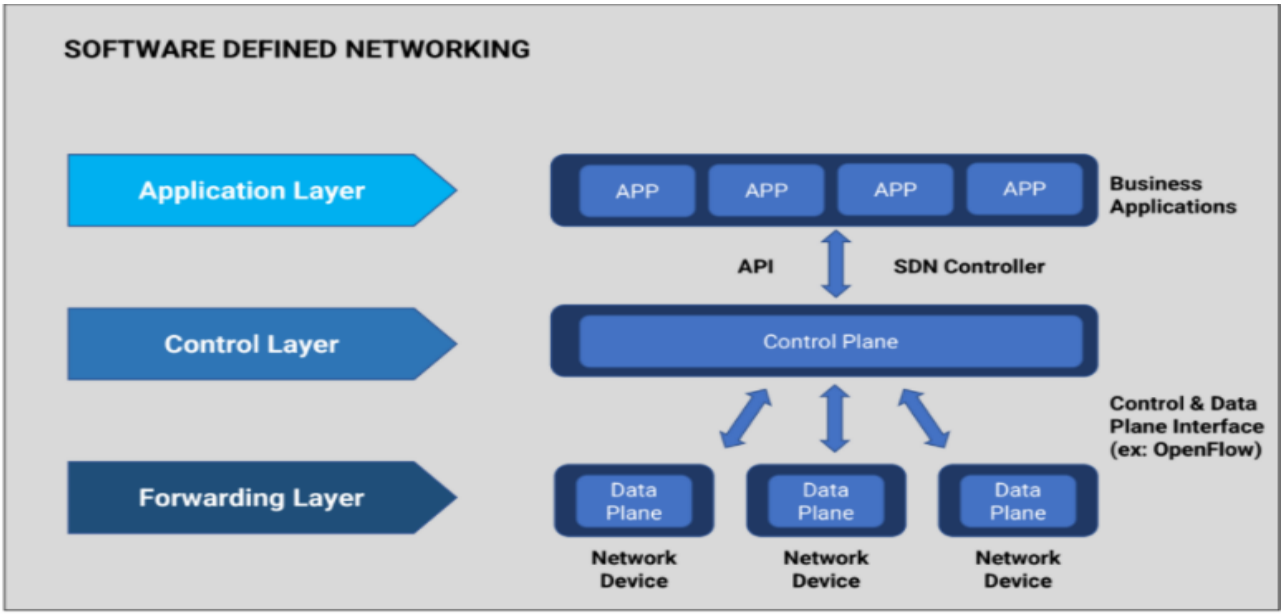
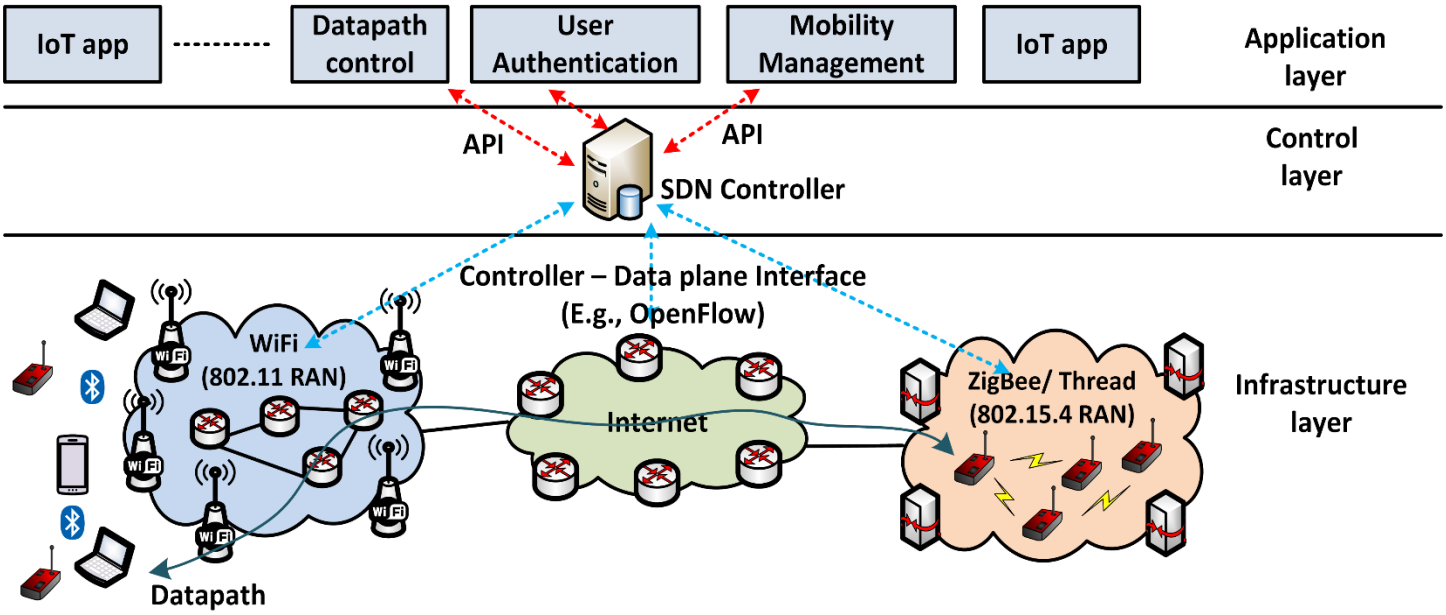


Figure 2.9 : Software Defined Networking (SDN) Architecture



Advantages of Software Defined Networking (SDN)

Software Defined Networking (SDN) offers several advantages over traditional networking architectures. One of the major benefits of SDN is easier network management. By separating the control plane from the data forwarding plane, SDN reduces the complexity involved in configuring and managing individual network devices.

SDN enables a programmable network, allowing administrators to define network behavior through software rather than manual hardware configuration. This programmability makes the network highly flexible and adaptable to changing requirements.

Another significant advantage is centralized network management. The SDN controller maintains a global view of the network, enabling efficient monitoring, configuration, and troubleshooting from a single point of control. SDN also improves network agility, as changes such as traffic rerouting, policy updates, or service deployment can be implemented quickly without modifying each device individually. Additionally, SDN provides improved network visibility, allowing administrators to analyze traffic flows and optimize performance effectively.

From a cost perspective, SDN is a cost-effective networking solution. It reduces dependency on specialized hardware and lowers operational and maintenance costs through automation. Furthermore, SDN enhances network security by enabling centralized policy enforcement, rapid threat detection, and dynamic security rule implementation.

Software Defined Networking:

- SDN is a networking architecture that separates the control plane from the data plane and centralizes the network controller .
- Control plane is the part of the network carries the signaling and routing message traffic while the data plane is the part of the network that carries the payload data traffic.
- The limitations of the conventional network architecture

Complex Network Device:

- conventional networks are getting increasingly complex with more and more protocols being implemented to improve link speeds and reliability. Interoperability is limited due to the lack of standard and open interfaces.
- Due to the complexity of conventional network devices , making changes in the networks to meet the dynamic traffic patterns has become increasingly difficult.

Management Overhead:

- Convention networks involve significant management overhead.
- Network managers find it increasingly difficult to manage multiple devices and interfaces from multiple vendors.
- Upgrading of network requires configuration changes in multiples devices (Switches , routers , firewalls ,….)

Limited Scalability:

- The virtualization technologies used in cloud computing environment has increased the number of virtual hosts requiring network access.
- IoT application hosted in the cloud are distributed across multiple virtual machines that requires exchange of huge data for analytics.

- Such computing environment requires highly scalable and easy to manage network architecture with minimal manual configuration , which is becoming increasingly difficult with conventional networks.

Network Function Virtualization (NFV):

Network Function Virtualization (NFV) Background:

- VM technology used for application–level server functions such as database server , cloud servers, web servers, e-mail server, and soon.
- This same technology , however , can equally be applied to network devices , such as routers, switches, LAN, firewalls, and IDS/IPS devices.

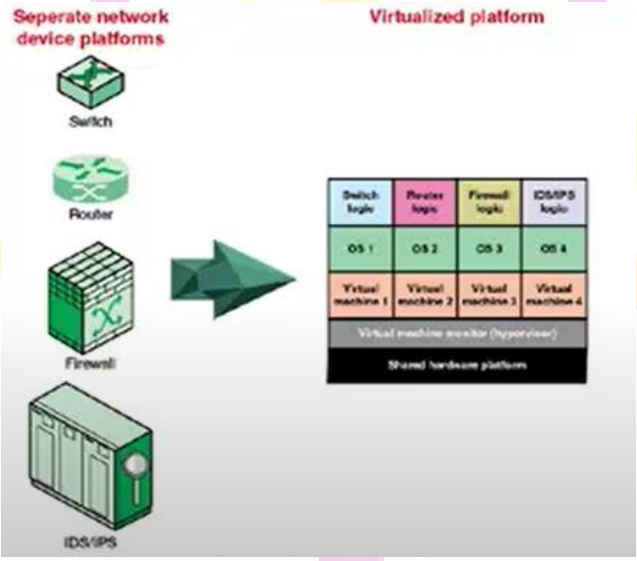


Figure 2.10 : Network Function Virtualization (NFV) Background

Network Function Virtualization (NFV) Basics:

Network Function Virtualization (NFV) is a networking architecture that focuses on virtualizing network services traditionally carried out by dedicated hardware devices. NFV stands for **Network Function Virtualization**, and it enables network functions such as routing, firewalling, load balancing, and intrusion detection to run as software applications.

The core concept of NFV is the replacement of specialized network devices like routers and firewalls with software-based network functions running on **general-purpose CPUs or virtual machines** deployed on standard server hardware. This approach eliminates the need for proprietary hardware and allows greater flexibility in deploying and managing network services.

NFV provides the underlying **virtualized infrastructure** on which modern networking paradigms such as Software Defined Networking (SDN) can operate efficiently. While NFV and SDN are often used together to enhance network programmability and flexibility, they are **not dependent on each other**. Instead, they are **mutually beneficial**, with NFV handling the virtualization of network functions and SDN managing the control and orchestration of network traffic.

Overall, NFV improves scalability, reduces operational costs, accelerates service deployment, and supports dynamic and flexible network architectures.

Network Function Virtualization (NFV) Architecture:

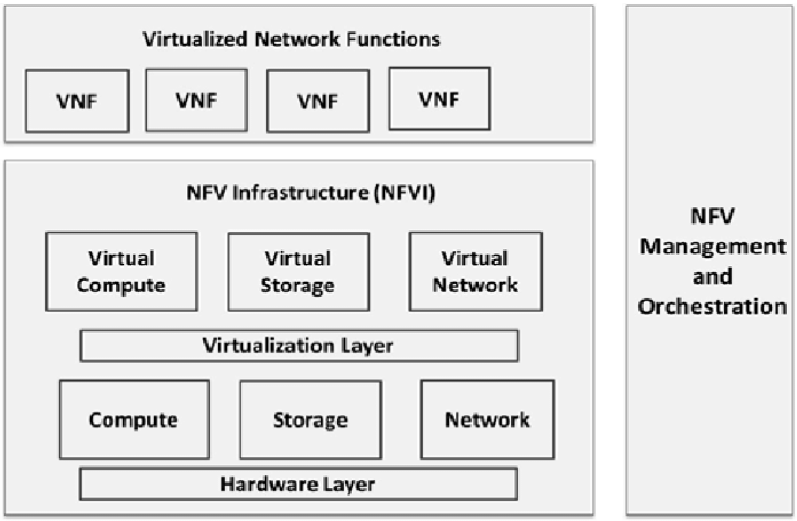


Figure 2.11 : Network Function Virtualization (NFV) Architecture

Key elements of Network Function Virtualization (NFV) :

NFV Infrastructure (NFVI) :

NFV Infrastructure (NFVI) forms the foundational framework that supports Network Function Virtualization (NFV). It provides the physical and virtual resources required to deploy and run virtualized network functions efficiently.

The first layer of NFVI consists of the physical hardware resources. These include computing resources such as CPUs, storage resources like hard disks, and network resources including routers, switches, and firewalls. These hardware components provide the basic infrastructure needed for virtualization.

The second layer of NFVI is the virtualization layer. This layer abstracts and separates the underlying hardware from the software by using virtualization technologies such as hypervisors. It enables hardware resources to be shared and managed efficiently, replacing hardware-specific functions with software-based implementations.

The third layer of NFVI consists of virtualized resources. These include virtual machines or virtual computers, virtual networks, and virtual storage. These resources are dynamically allocated and managed to support various virtualized network functions.

Overall, NFVI enables flexibility, scalability, and efficient utilization of resources, forming the backbone for deploying NFV-based network services.

Virtualized Network Function (VNF):

A Virtualized Network Function (VNF) is a software-based implementation of a network function that traditionally runs on dedicated hardware devices. Instead of relying on specialized hardware, VNFs operate as software applications.

VNFs are designed to run on the NFV Infrastructure (NFVI), which provides the required computing, storage,

and networking resources through virtualization technologies. By running over NFVI, VNFs can be deployed, scaled, and managed dynamically according to network requirements.

Common examples of VNFs include virtual firewalls (vFirewall) and virtual routers (vRouter). These virtualized functions offer the same capabilities as their hardware counterparts while providing greater flexibility, scalability, and cost efficiency.

Overall, VNFs enable rapid service deployment, efficient resource utilization, and support modern, software-driven network architectures.

NFV Management and Orchestration

NFV Management and Orchestration (NFV-MANO) is a framework responsible for managing and coordinating the virtualized resources and services in a Network Function Virtualization (NFV) environment. It ensures efficient deployment, operation, scaling, and monitoring of virtual network functions and network services.

The Virtualized Infrastructure Manager (VIM) controls and manages the NFV Infrastructure (NFVI) resources such as compute, storage, and networking. It also monitors the virtualization layer to ensure optimal utilization of hardware resources and supports interaction between VNFs and the underlying infrastructure.

The VNF Manager (VNFM) is responsible for managing the life cycle of Virtualized Network Functions (VNFs). Its functions include initialization, configuration, updating, querying, scaling, and termination of VNFs. VNFM ensures that VNFs operate correctly throughout their lifecycle.

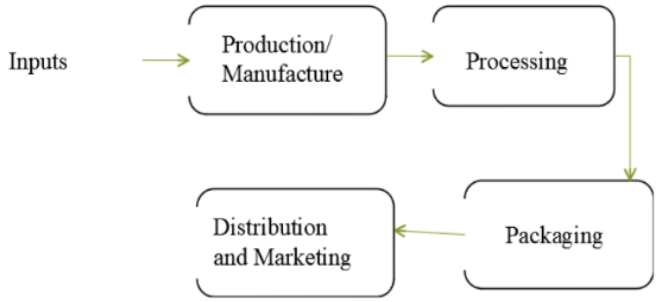
The Orchestrator manages the life cycle of network services by coordinating multiple VNFs and allocating NFVI resources based on defined policies. It handles service orchestration, resource coordination, and ensures compliance with management policies.

NFV-MANO also supports performance measurement, monitoring, and management, enabling service reliability, scalability, and efficient network operations.

Overall, NFV-MANO plays a crucial role in automating network service deployment, improving flexibility, and enabling dynamic management of virtualized network environments.

Value Chain:

- A value chain refers to the full range of activities a company undertakes to deliver a valuable product or service to the customer.
- It encompasses all the steps from design and production to marketing, delivery, and support.
- By analyzing the value chain, businesses can identify opportunities to improve efficiency, reduce costs, and enhance customer value.



M2M value chain

Figure 2.12 : M2M Value Chain

Category	Description
Input	Inputs are the base raw ingredients that are turned into a product.
Example	Cocoa beans for the manufacture of chocolate.
M2M example	Data from an M2M device that will be turned into a piece of information.
Production / Manufacture	Production/Manufacture refers to the process that the raw inputs are put through to become part of a value chain.
Example	Cocoa beans may be dried and separated before being transported to overseas markets.
M2M example	Data from an M2M needs to be verified and tagged for provenance.
Processing	Processing refers to the process whereby a product is prepared for sale.
Example	Cocoa beans may now be made into cocoa powder, ready for use in chocolate bars.
M2M example	M2M refers to the aggregation of multiple data sources to create an information component that is ready to be combined with other data sets to make it useful for corporate decision-making.
Packaging	Packaging refers to the process whereby a product can be branded as would be recognizable to end-user consumers.
Example	A chocolate bar would now be ready to eat and have a red wrapper with the words “KitKat” on it.
M2M example	M2M data will have to be combined with other information from internal corporate databases, for example, to see whether the data received requires any action. This data would be recognizable to the end-users that need to use the information, either in the f
M2M example	M2M data will have to be combined with other information from internal corporate databases, for example, to see whether the data received requires any action. This data would be recognizable to the end-users that need to use the information either in the form of

Category	Description
	visualizations or an Excel spreadsheet.
Distribution / Marketing	This process refers to the channels to market for products.
Example	A chocolate bar may be sold at a supermarket, a kiosk, or even online.
M2M example	Will have produced an Information Product that can be used to create new knowledge within a corporate environment. Examples include more detailed scheduling of maintenance based on real-world information or improved product design due to feedback from the M2M solution.

IoT value chains :

Category	Description
Inputs	Significantly more inputs than for an M2M solution.
Devices / Sensors	Data from devices and sensors is used to provide a different and much broader marketplace than M2M does.
Open Data	A piece of data is open if anyone is free to use, reuse, and redistribute it, subject only (at most) to the requirement to attribute and/or share-alike. **Example:** City maps provided by organizations such as Ordnance Survey in the United Kingdom.
OSS / BSS	The Operational Support Systems (OSS) and Business Support Systems (BSS) are closed information marketplaces that allow operators to deliver services to enterprises. **Example:** Phone usage data already owned by the company.
Corporate Databases	Companies of a certain size generally have multiple corporate databases covering various functions, including supply chain management, payroll, and accounting. As the use of devices and sensors increases, these databases will be connected to this data to create new information sources and new knowledge.
Production / Manufacture	Process will need to include tagging and linking of relevant data items in order to provide provenance and traceability across the information value chain.
Asset Information	Asset information may include data such as temperature over time of a container during transit or air quality during a particular month.
Open Data Sets	Maps, rail timetables, or demographics about a certain area in a country or city.
Network Information	GPS data, services accessed via the mobile network.

Category	Description
Corporate Information	The current state of demand for a particular product in the supply chain at a particular moment in time.
Processing	The data from the various inputs from the production and manufacture stage are combined together to create information.
Packaging	The packaging section of the information value chain creates information components. These components could be produced as charts or other traditional methods of communicating information to end-users.
Distribution / Marketing	The final stage of the Information Value Chain is the creation of an Information Product.

Information products for improving internal decision-making	These information products are the result of either detailed information analysis that allows better decisions to be made during various internal corporate processes, or they enable the creation of previously unavailable knowledge about a company's products, strategy, or internal processes.
Information products for resale to other economic actors	These information products have high value for other economic actors and can be sold to them.

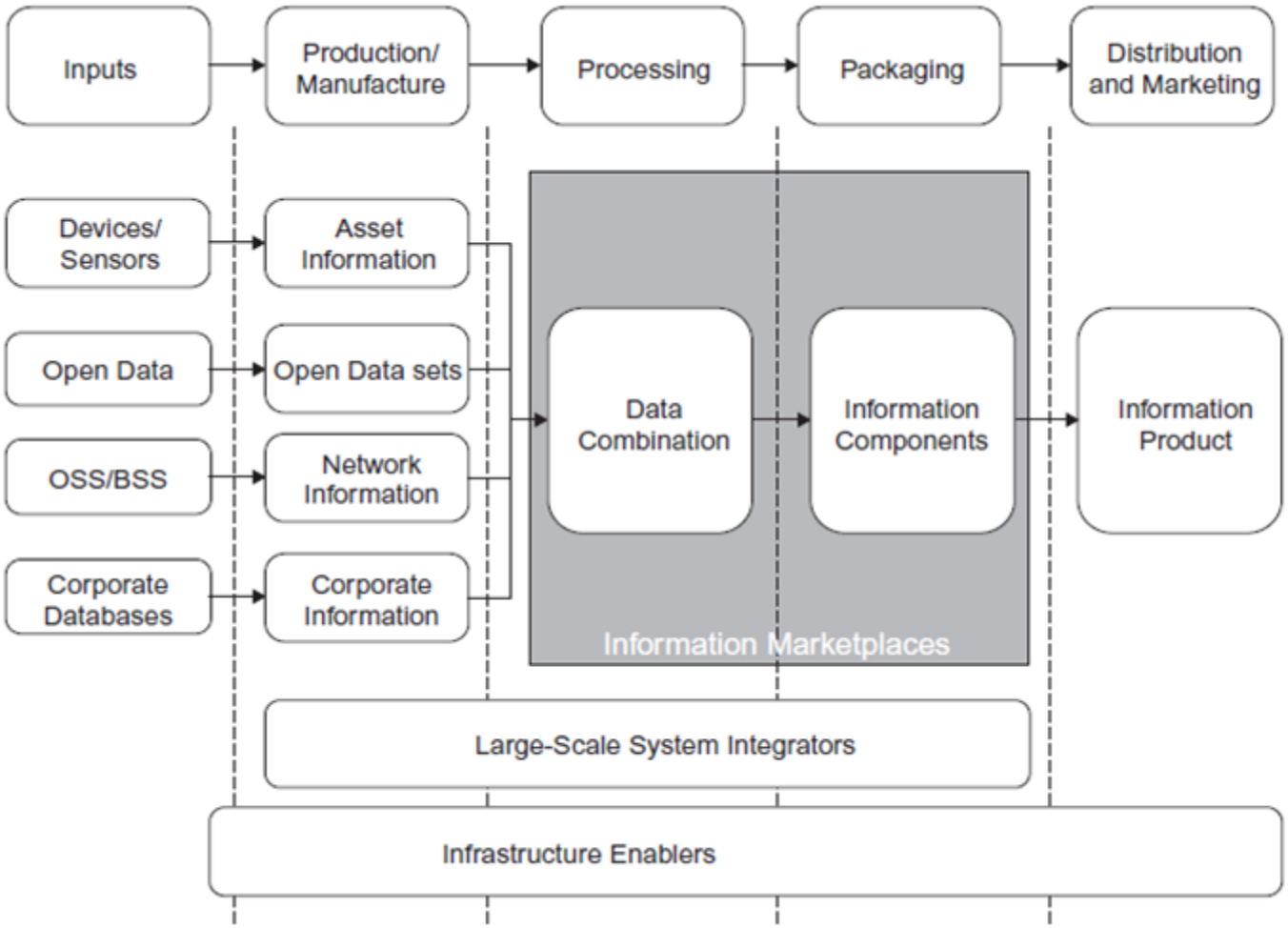


Figure 2.13: IoT Value Chain

An emerging industrial structure for IoT:

- M2M and IoT are about joining data and workflows that form the base of Global Economy at increasing the speed and output.
- A new form of Value chain is coming up – driven by Information product.
- In it information gathered from sensors and RFID 's and information from smartphones.
- Information about individuals is captured , stored , processed and REUSED.
- Actors that performs this collection , storage , and processing are forming the basis of Information driven global chain.

Example :

The value of information varies depending on the user and the purpose for which it is used. For an individual, contextual information such as location data may have relatively low perceived value. For example, when a person arrives in a new city and looks for a coffee shop, they may choose to use a smartphone's GPS-enabled search or simply walk around until they find a suitable place. In this case, the individual may not highly value the use of location-based information.

In contrast, the same information can have significantly higher value for organizations. For a coffee company,

knowing that a large number of potential customers (e.g., several hundred women stepping off a train in search of coffee) represents valuable market intelligence. Such information can help businesses identify profitable locations for new stores.

Furthermore, combining location data with demographic information—such as age group, education level, and consumer preferences—enables companies to better understand their target market. This allows organizations to tailor products, services, and store designs more precisely, thereby increasing competitiveness and profitability.

Emerging IGVC- Actors :

- Information Driven global chain

There are five fundamental roles within the I-GVC that companies and other actors are forming around.

- Inputs
  - ❑ Sensors, RFID, and other devices
  - ❑ End-Users.
- Data factories.
- Service Providers/Data wholesalers.
- Intermediaries.
- Resellers.

Emerging IGVC- Actors :

- Inputs
  - ❑ Sensors, RFID, and other devices
  - ❑ End-Users.
- **Production Unit**
- **Data Factories:**

They produce data in the digital form for use in other parts of IGVC

Ex: Maps information , weather station.

- **Service Providers/Data Wholesalers :**
- They collect data from various data sources world wide and through the creation of massive databases use it either to improve their information product or sell information in various forms.
- EG, FB, Twitter, Google.
- **Intermediaries :**
- It acts on a person behalf, tagging the information in some form to ensure it is not used in the manner which is previously not agreed to.
- Eg: DATA from FB has privacy and regional issues in Europe.
- **Resellers :**
- They are the entities which combine inputs from several intermediaries.
- They combine it together , analyze and sell it to either end users or corporate.

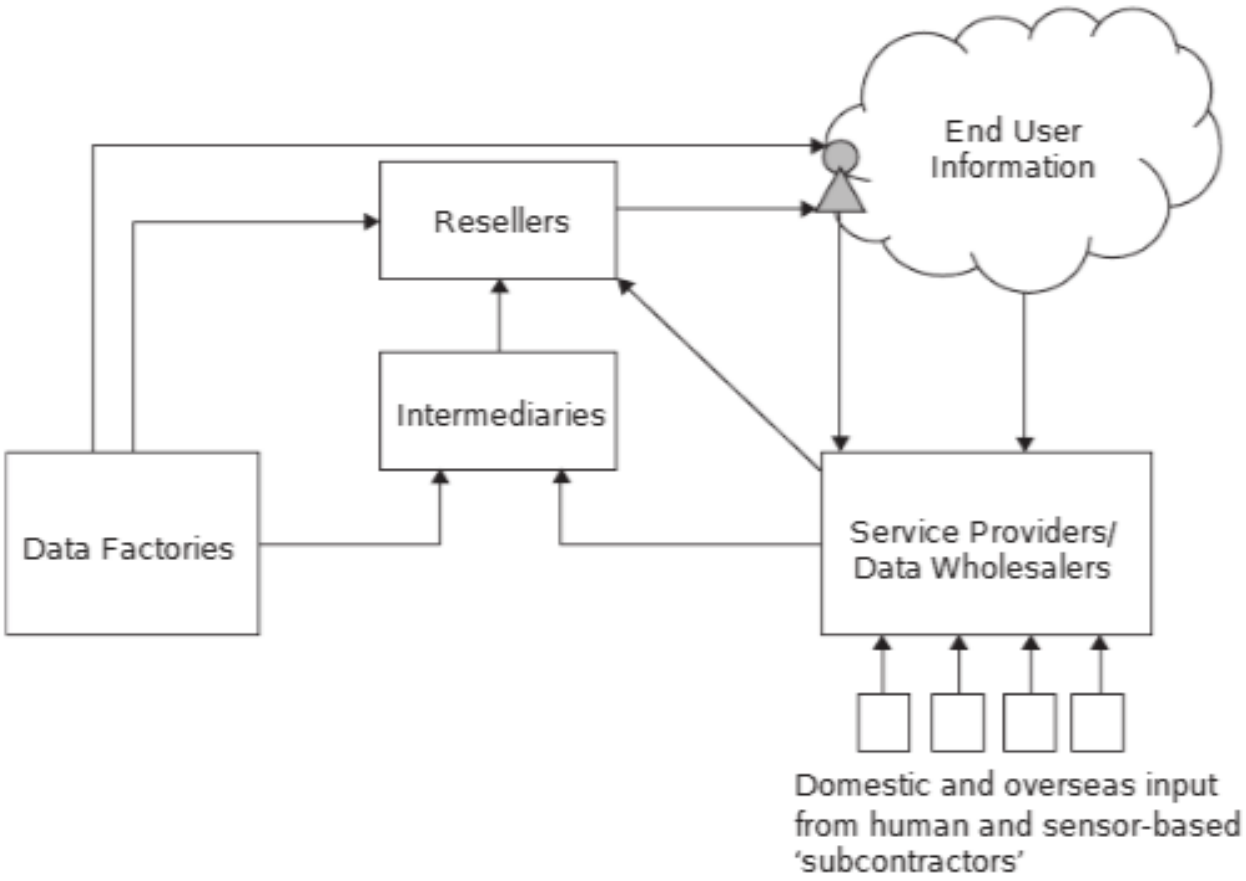


Figure 2.14 : Emerging IGVC- Actors

International driven global value chain and global information monopolies :

- Currently within the industrial structure of the converged communications industry, there is a large regional disparity between those companies that produce the infrastructure for the I-GVC and those that make a significant profit from it.
- Through positioning themselves within the correct part of the GVC, these companies are able to take the lion's share of the profit.
- Through the breakdown of regional boundaries for collection of data by the development and implementation of a global converged communications infrastructure, these companies are able to enlist every person using a mobile device worldwide as a contributor to the development of their information products in effect, every person worldwide is working for these corporations so that they are able to sell aggregated data for a huge profit.
- Despite this data being collected from people in every corner of the globe, from the UK, Thailand, Australia, China, and Africa, to even there motest parts of Kashmir, the surplus value of the mobile broadband plat-form is currently being captured, developed, and molded into information products, overwhelmingly by U.S. companies.
- Through being able to collect and analyze data without being restricted by the same level of privacy

regulation as in Europe, for example, they are able to create a much better information product.

- Companies in Europe, Asia, and other parts of the globe are therefore dependent on these companies in order to gain the most appropriate knowledge for their companies' needs.
- Companies are therefore compelled to use the most effective information product for their needs.
- In effect, the I-GVC, rather than breaking down the digital divide as many have predicted, is in fact leading to a new form of digital discrimination and a new sort of dependency relationship between large multinationals and those participants, or "workers," within the I-GVC.
- While there may appear to be huge differences between the industrial revolution and the birth of the digital planet in the nature of how workers are treated, in particular with so much being advertised as "free" for end-users, there are in fact many similar parallels in the aggregation of human endeavor in the processes of the accumulation of capital.
- A multitude of workers contribute to the information products developed, but only a few large corporations capture the surplus value.

IoT architecture components :

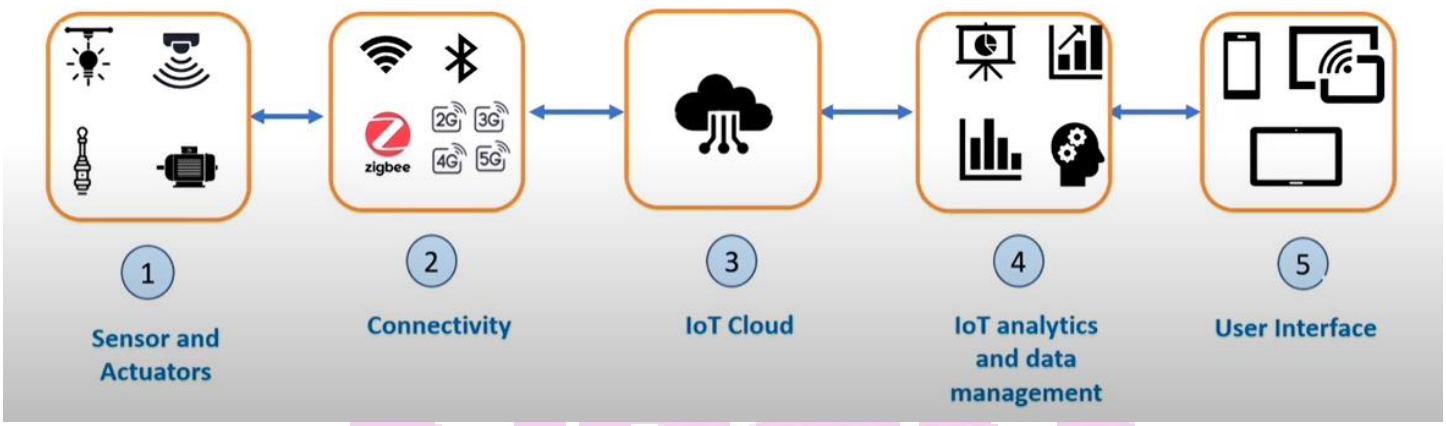


Figure 2.15 : IoT Architecture Components

1.Sensors & Actuators :

Sensors :

- IoT sensors play a key role in data acquisition systems by enabling the collection of raw data from the environment.
- These physical devices can be embedded within connected devices or remotely located to gather information about their surroundings.
- **Practical applications include:**
- Temperature detectors for smart homes or smart buildings
- Smoke detectors for improving security during natural disasters
- Cameras and CCTVs for monitoring systems in healthcare facilities or the manufacturing industry

Acutators :

- Actuators are devices that respond to data inputs to automate tasks without direct human intervention.
- They transform instructions into physical actions, ensuring seamless operations.
- **Examples include:**
- Smart lights turning on or off
- Door locks opening or closing
- Thermostats adjusting temperatures in smart homes

2. Connectivity &Gateway :

- Protocols
    - Job of IoT Protocols
- Requires a Medium
- Eg: Cellular, Bluetooth, LoRaWAN, etc…

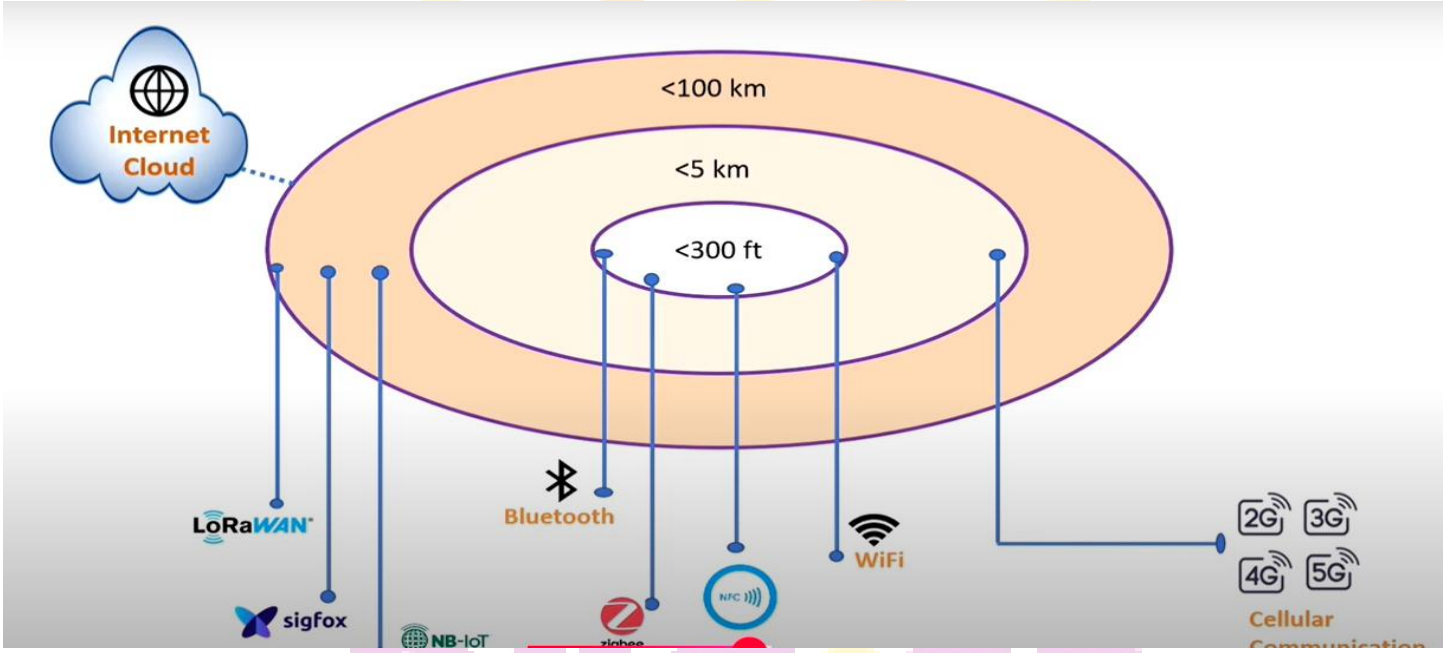


Figure 2.16 : Range of Protocol types

IoT Gateways:

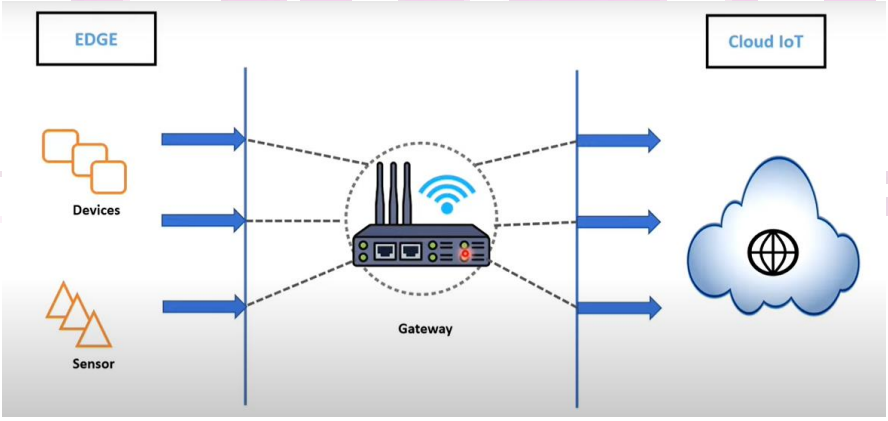


Figure 2.17: IoT Gateways

Role of Gateways :

- Ensure Seamless Communication.
- Easy management of Data Traffic.
- Offers security.
- Data Processing
- Minimize the large data.
- Intelligent IoT gateways.

3. IoT Cloud :

- Role of an IoT cloud
    - Data needs to be processed.
    - To make or break a deal.
    - Latency cannot be compromised.
    - The brain of the IoT ecosystem.
    - Cloud is optional.
    - Edge computing.

4. IoT Analytics and Data management :

Figure 2.18: IoT Analytics and Data management :

- Data fuel.
- Converting raw data into useful insights.
- Data extraction, data aggregation , data classification.
- Basic vs Complex analytics.
- Deep learning
- Storage power and intelligent computation.
- Hosted on the cloud , depending on the IoT architecture.

5. User interface :

- Types of notifications
- Can also send back the command to the field devices.
- It might provide the user with actual live feed or show trends etc.

COMPARING IoT ARCHITECTURES

- In the past several years, architectural standards and frameworks have emerged to address the challenge of designing massive-scale IoT networks.
- The foundational concept in all these architectures is supporting data, process, and the functions that endpoint devices perform.
- Two of the best-known architectures
- • oneM2M
- • IoT World Forum (IoTWF)

THE ONEM2M IOT STANDARDIZED ARCHITECTURE

- European Telecommunications Standards Institute (ETSI) created the M2M Technical Committee in 2008.
- The goal of this committee was to
    - create a common architecture that would help accelerate the adoption of M2M applications and devices.
    - In 2012 launched oneM2M as a global initiative designed to promote efficient M2M communication systems with IoT.
    - One of the greatest challenges in designing an IoT architecture is dealing with the heterogeneity of devices, software, and access methods.

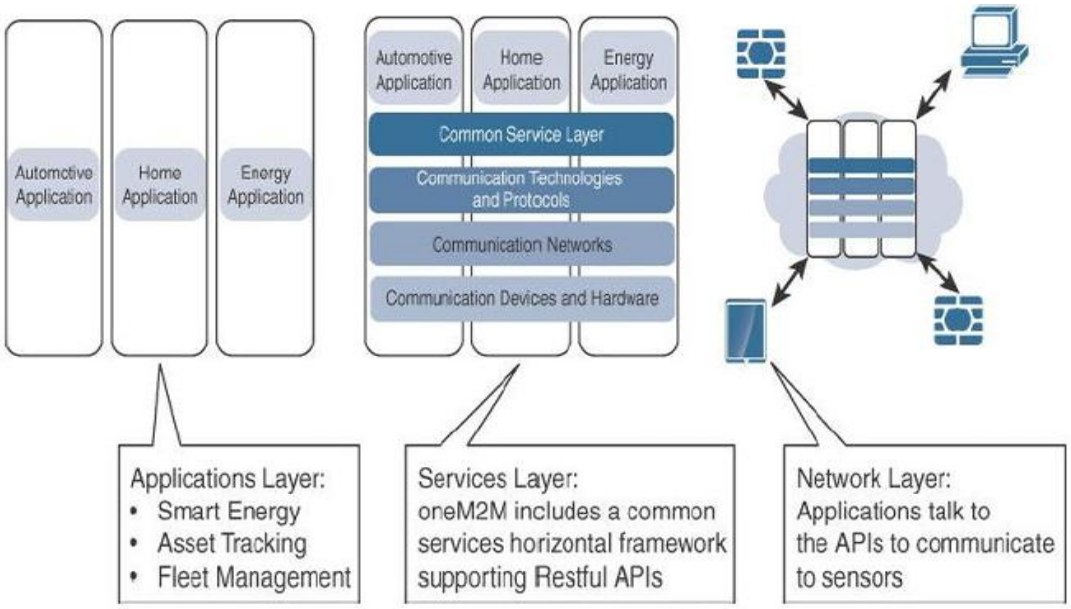


Figure 2.19 : The ONEM2M IoT Standardized Architecture

i. Applications layer

- The primary focus of the applications layer is to enhance connectivity between IoT devices and the software applications they interact with.
- This layer includes the protocols that define how devices communicate with applications, ensuring compatibility and smooth operation.
- The applications layer defines standard APIs (Application Programming Interfaces) to enable seamless interaction between devices, business intelligence systems, and cloud platforms.
- By standardizing APIs, the oneM2M architecture supports cross-industry IoT applications and ensures compatibility in multi-vendor environments.

ii. Service layer

- This layer is designed as a horizontal framework that spans across vertical industry-specific applications.
- It abstracts underlying complexities and provides common services required by various IoT applications, such as device management, data storage, and security.
- **Components:** Includes physical networks, management protocols, and hardware modules essential for IoT functionality.
- **Significance:** Enables interoperability across different IoT systems and ensures consistent performance in complex environments.

iii. Network Layer

- The network layer focuses on communication between IoT devices and the endpoints to which they are connected.
- Includes a variety of communication technologies:
    - Wireless Mesh Networks: IEEE 802.15.4
    - Wireless Point-to-Multipoint Systems: IEEE 802.11 (Wi-Fi)

- Wired Communication: IEEE 1901 (Power Line Communication)

- Purpose: Ensures reliable and efficient data transfer across IoT devices in diverse environments.

THE IOT WORLD FORUM (IOTWF) STANDARDIZED ARCHITECTURE

- In 2014 the IoTWF architectural committee (led by Cisco, IBM, Rockwell Automation, and others) published a seven layer IoT architectural reference model.
- The IoTWF developed a reference model to provide a clear and structured framework for understanding and implementing IoT systems.
- This model enables a layered approach to solving IoT challenges and ensures better scalability, interoperability, and security in IoT deployments.

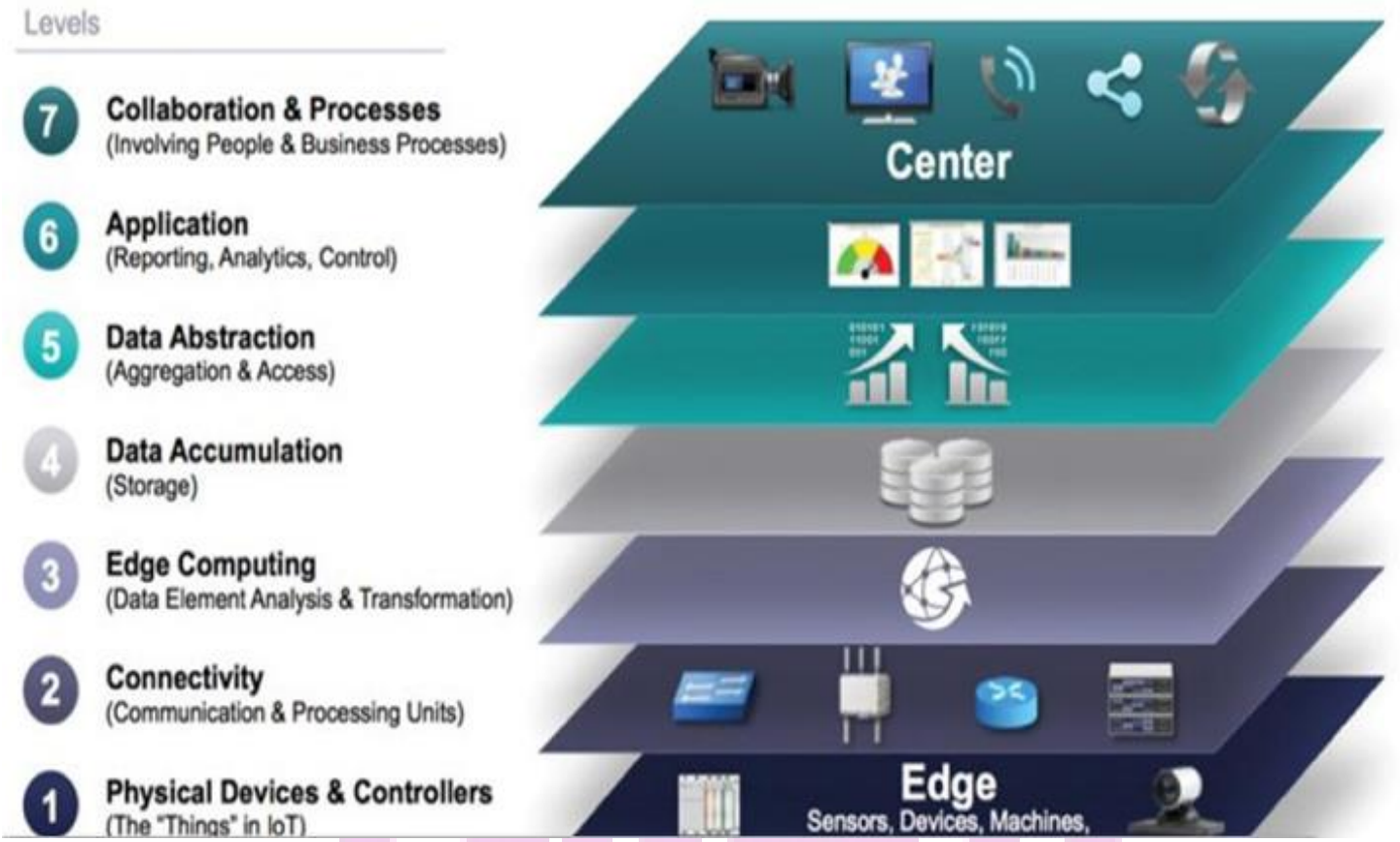


Figure 2.20 : **The IOT WORLD FORUM (IOTWF) Standardized Architecture**

Layer 1: Physical Devices and Controllers Layer

- This layer includes all the physical "things" that form the backbone of IoT, such as sensors, actuators, and machines.
- These devices generate data and often interact with their environment.
- **Device Range:** Devices in this layer vary greatly in size, from microscopic smart sensors to large industrial machinery.
- **Functionality:** These devices perform two primary functions:
- Generating data for further analysis.
- Being controlled remotely via a network.

Layer 2: Connectivity Layer

- The connectivity layer handles the reliable and timely transmission of data from the physical devices to the IoT network.
- **Key Functions:**
- It ensures that data from endpoint devices reaches the right destinations without loss or delay.
- Supports various communication technologies, including wired and wireless networks.
- **Examples of Connectivity Protocols:**
- Wi-Fi, Bluetooth, ZigBee, LPWAN, and Ethernet

Layer 3: Edge Computing

- Also Known As: The "Fog" Layer.
- **Purpose:** At this layer, data is processed closer to the source to reduce the amount of raw data sent to the cloud.
- This helps minimize latency, conserve bandwidth, and improve the speed of response.
- **Functions:**
- **Data reduction:** Aggregates and filters raw data to extract relevant information.
- **Information conversion:** Transforms real-time network data into actionable insights.

Layer 4 - Data Accumulation Layer

- This layer captures and stores IoT data, ensuring it is readily available for further processing.
- It converts event-based data into a form that allows query based processing by applications.
- **Importance:** Provides structured and accessible data for advanced analytics, applications, and reports.

Layer 5 - Data Abstraction Layer

- **Purpose:** is a software layer that hides the complexity of underlying hardware, protocols, and data formats. It provides a uniform interface for accessing and managing data from heterogeneous IoT devices and systems.
- **Functions:** Consolidates data into a centralized repository or multiple stores.
- Offers data visualization for easier interpretation and usability.
- **Outcome:** Enables a single, coherent view of the data across the entire IoT system.

Layer 6 - Applications Layer

- Purpose: This layer interprets the collected and processed data using specialized software applications. Applications at this layer are responsible for monitoring, controlling devices, and generating reports based on analyzed data.
- **Examples:** Smart home apps, industrial automation dashboards, and healthcare monitoring systems.
- **Significance**: Acts as the interface between the IoT system and the end-user, allowing decision-making and operational insights.

Layer 7 - Collaboration and Processes Layer

- **Purpose:** This layer focuses on sharing IoT information and enabling collaboration among users and systems. It is the layer where IoT data drives business processes and organizational changes.
- **Functions:** Facilitates multi-step workflows to derive value from IoT data. Enables collaboration among different departments, systems, and users.
- **Key Outcome:** Delivers the true benefits of IoT by integrating IoT driven insights into business operations and decision-making.

Simplified IoT Architecture:

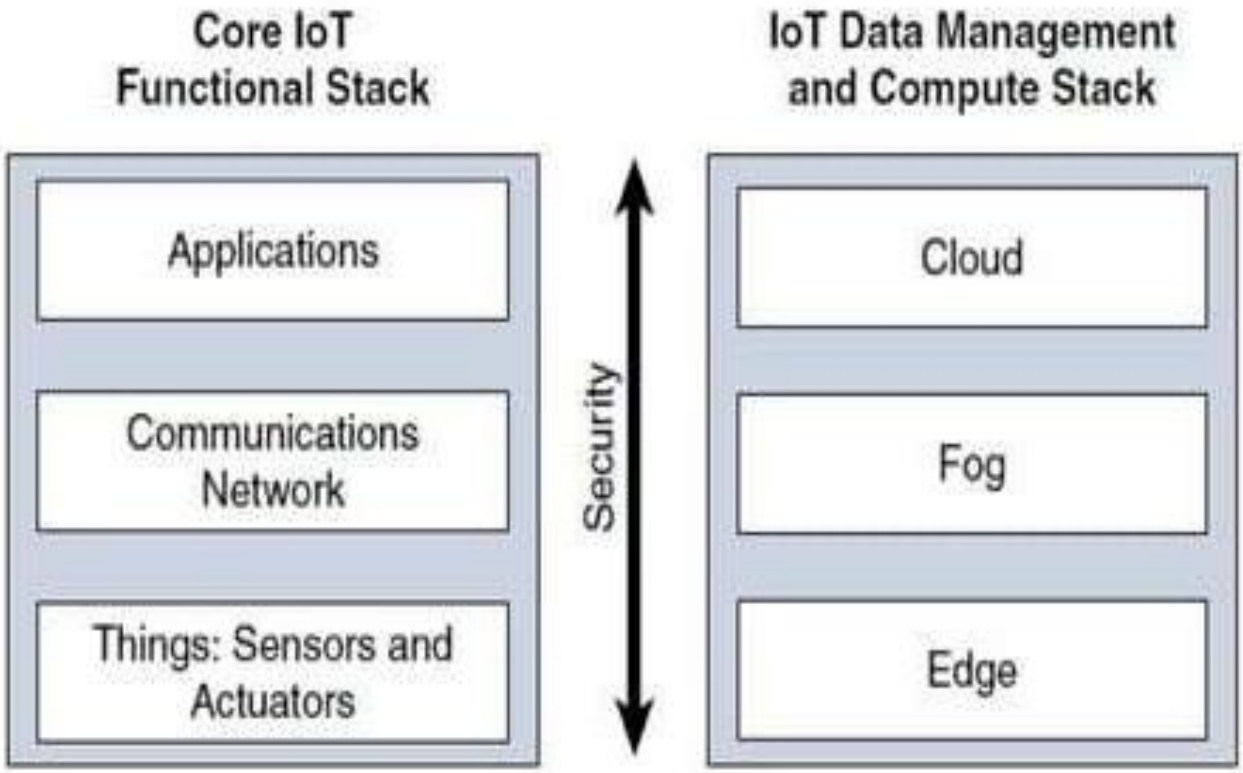


Figure 2.21 : Simplified IoT Architecture:

IoT Core Functional Stack :

Things layer (Sensors and Actuators):

- Most IoT networks start from the **object**, or "**thing**," that needs to be connected.
- At this layer, the physical devices need to fit the constraints of the environment in which they are deployed while still being able to provide the information needed.
- Battery-powered or power-connected
- Mobile or static
- Low or high reporting frequency
- Simple or rich data
- Report range
- Object density per cell

Things layer (Sensors and Actuators):

1. A first step in designing an IoT network is to examine the requirements in terms of mobility and data transmission (how much data, how often).
2. Once you have determined the influence of the smart object form factor over its transmission capabilities (transmission range, data volume and frequency, sensor density an mobility), you are ready to connect the object and communicate.

Communications network layer:

- When smart objects are not self contained, they need to communicate with an external system. In many cases, this communication uses a wireless technology.

- **Access network sublayer** : This layer facilitates communication between IoT devices and gate ways.
- **Gateways and backhaul network sublayer:** Act as bridge between smart devices and the central network, facilitating data collection and transmission
- **Network transport sublayer :** It facilitates data exchange between network access layer and application layer.
- **IoT network management sublayer:** It is responsible for managing and controlling the underlying network infrastructure that supports interconnected devices.

Application and analytics layer:

- At the upper layer, an application needs to process the collected data, not only to control the smart objects when necessary, but to make intelligent decision based on the information collected and, in turn, instruct the "things“ or other systems to adapt to the analyzed conditions and change their behaviors or parameters.
- **Analytics application:** This type of application collects data from multiple smart objects, processes the collected data, and displays information resulting from the data that was processed. The display can be about any aspect of the loT network, from historical reports, statistics, or trends to individual system states.
- **Control application:** This type of application controls the behavior of the smart object or the behavior of an object related to the smart object.
- **Eg.** Thermostat and AC
- In most cases, data is collected from the smart objects and processed in the analytics module.
- The result of this processing may be used to modify the behavior of smart objects or systems related to the smart objects.
- The control module is used to convey the instructions for behavioral changes.

IoT Data management and Compute Stack:

- The IoT Data Management and Compute Stack deals with how and where data is filtered, aggregated,

stored, and analyzed. In traditional IoT models, this occurs in the cloud or the data center.

- However, due to the unique requirements of IoT, data management is distributed as close to the edge as possible, including the edge and fog layers.
- Cloud layer (data management in the cloud or central data center).
- Fog layer (data management in the gateways and transit network)
- Edge layer (data management within the sensors themselves),

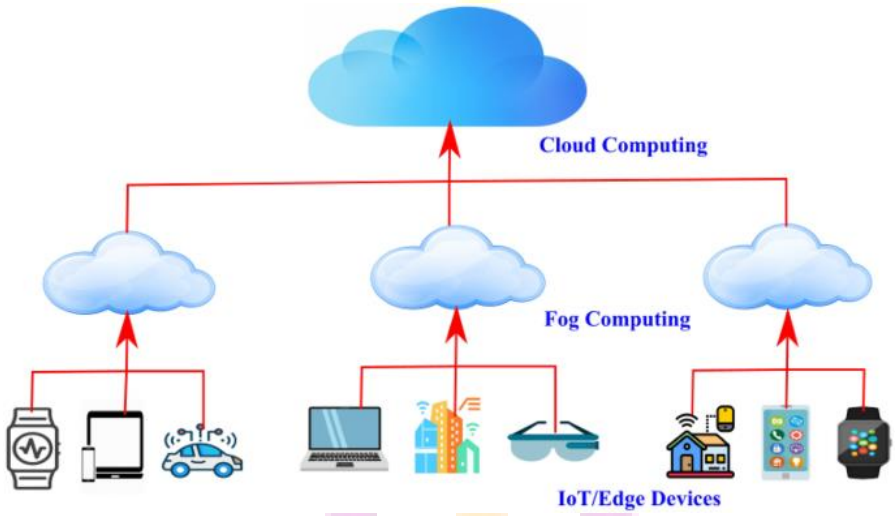


Figure 2.22 : Cloud Vs Fog Vs Edge Devices

1.Edge Computing:

- **Location:** Processing happens directly on or near the IoT devices, such as sensors and actuators.
- **Purpose:** Minimizes latency, conserves bandwidth by pre-processing data before sending it to other layers, and enables real-time responses.
- **Example:** A smart thermostat analyzes temperature data locally and adjusts the heating system without sending the data to the cloud.

2. Fog Computing:

- **Location:** Situated between the edge and the cloud, often in network devices like routers or gateways.
- **Purpose:** Provides distributed intelligence, enabling local data analysis, and real-time interactions.
- **Example:** A fog node on a factory floor analyzes sensor data from machines to detect potential issues and trigger alerts without needing to send all data to a central cloud.

3. Cloud Computing:

- **Location:** Centralized data centers or cloud platforms.
- **Purpose:** Stores large volumes of data for long-term analysis, supports complex analytics, and facilitates global data management.
- **Example:** Storing historical data for trend analysis, machine learning, and developing new applications.