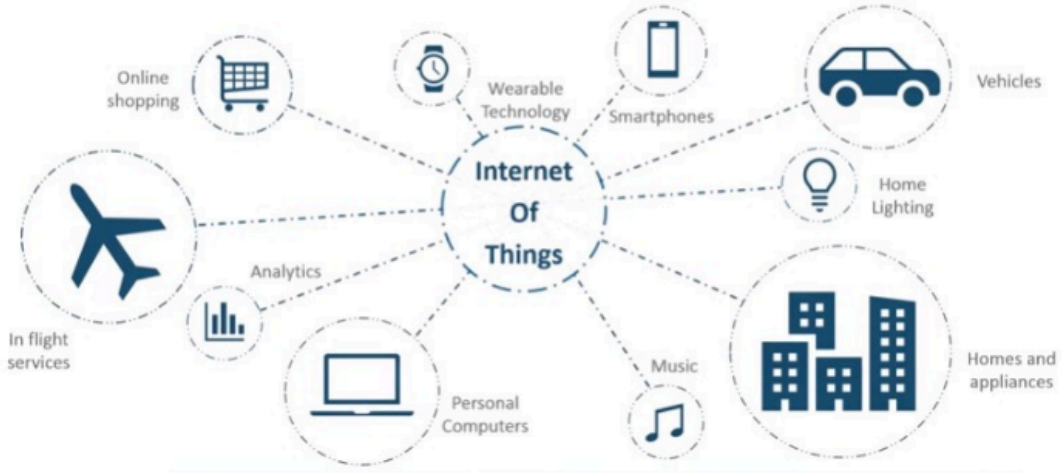
UNIT - 1 IOT Notes - R2

UNIT 1: Introduction to IoT

1. What is IoT (Internet of Things)?

- **Definition:** IoT stands for Internet of Things. It refers to the interconnectedness of physical devices (appliances, vehicles, etc.) embedded with software, sensors, and connectivity, enabling them to connect and exchange data.

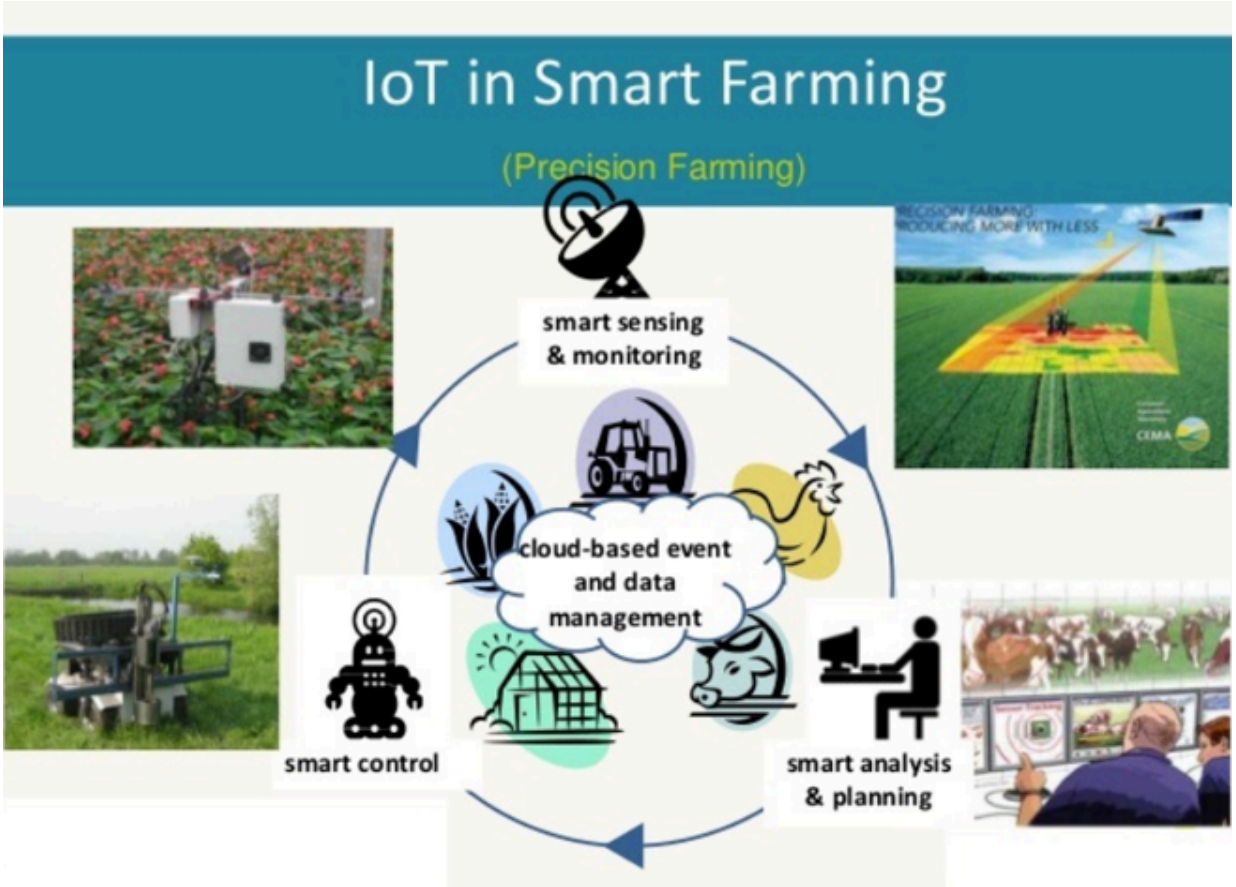


- **Ecosystem:** It's an ecosystem where interacting devices share data through a communication medium (the internet).
- **Core Function:** Allows collection and sharing of data from a vast network of devices, creating opportunities for more efficient and automated systems.
- **"Things":** Physical objects with electronics embedded within their architecture to communicate and sense interactions with each other or the external environment. They are provided with unique identifiers.
- **Layman's Terms:** Taking everyday physical objects ("things") and connecting them to the internet to make them "smarter" by enabling them to send information, receive information, or both.
    - *Example:* A smartphone can access vast amounts of information (music, movies) not stored locally because it's connected.
- **Goal:** To offer advanced services and practically change daily lives through advancements in various fields like medicine, power, agriculture, smart cities, and smart homes.
- **Data Transfer:** Involves the ability to transfer data over a network, sometimes requiring human-to-human or human-to-computer interaction, but often aiming for M2M.

2. Why IoT Matters?

- **Information Exchange:** Connected devices can send or receive information, or both.

- **"Smart" Capability:** This ability makes "things" smart. Smartness doesn't require on-device supercomputing; connection to super storage or supercomputers is sufficient.
- **Three Categories of Connected Things:**
  1. **Things that collect information and then send it:** Primarily involves sensors.
     - **Sensors:** Like human senses, they allow machines to understand the world (e.g., temperature, motion, moisture, air quality, light sensors).
     - **Benefit:** Automatic data collection enables more intelligent decisions.
     - *Example:* Soil moisture sensor informs farmers when to water crops, optimizing water usage and improving yield.
  2. **Things that receive information and then act on it:** Involves actuators.
     - *Examples:* A printer receiving a document and printing it; car doors unlocking upon receiving a signal from keys.
     - **Remote Control:** Allows machines to be controlled from afar.
  3. **Things that do both (Collect, Send, Receive, Act):** The true power of IoT.
     - *Farming Example:* Sensors collect soil moisture data, and the irrigation system automatically turns on/off. If connected to weather data, it can decide not to water if rain is predicted. Data can be sent to supercomputers for algorithmic analysis to optimize crop growth.



3. History of IoT [PYQ for timeline awareness]

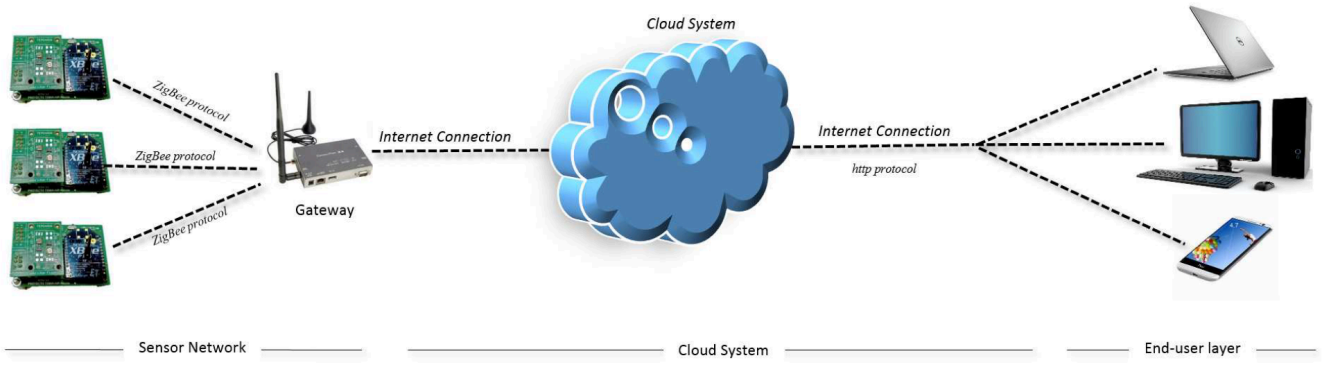
- **1982 – Vending Machine:** Carnegie Mellon University vending machine connected to the internet to report inventory/status (remote monitoring).

- **1990 – Toaster:** Toaster connected to the internet for remote control (foreshadowing smart home devices).
- **1999 – IoT Coined (Kevin Ashton):** Term "Internet of Things" coined to describe interconnected devices sharing data.
- **2000 – LG Smart Fridge:** Enabled remote checking/management of refrigerator contents.

- **2004 – Smart Watch:** Introduced IoT to wearable tech (fitness tracking, notifications).
- **2007 – Smart iPhone:** Integrated IoT capabilities with apps, transforming smartphones into hubs.
- **2009 – Car Testing:** IoT entered the automotive industry (real-time diagnostics, performance monitoring).
- **2011 – Smart TV:** Brought IoT to living rooms (streaming, apps, interactive content).
- **2013 – Google Lens:** Showcased IoT potential in image recognition.
- **2014 – Amazon Echo (Alexa):** Demonstrated voice-activated IoT, making smart homes intuitive.
- **2015 – Tesla Autopilot:** Exemplified IoT in automobiles with semi-autonomous driving through interconnected sensors/software.

4. How does an IoT System Actually Work? [PYQ for conceptual model]

- **Data Collection:** Sensors (single or a collection called devices) collect data from the environment or user inputs (e.g., GPS, LDR, temperature sensors).
- **Data Transmission:** Collected data is sent to the cloud via a connection (WiFi, LAN, satellite, Bluetooth, etc.).
- **Data Processing:** In the cloud, software reads and analyzes the data according to programmed logic (e.g., reading temperature, weather conditions, image processing) to make predictions or decisions.
- **User Information/Action:** Processed information is sent to the user (e.g., alerts for high temperature, weather updates) or triggers an action by an actuator.



5. Characteristics of the Internet of Things [PYQ]

- **1. Connectivity:** Essential requirement. Things of IoT *must* be connected to the IoT infrastructure. Guaranteed "anyone, anywhere, anytime" connection (people via devices, devices to devices like routers, sensors).
- **2. Intelligence and Identity:**
  - **Intelligence:** Extracting knowledge from generated data is crucial (e.g., sensor data needs proper interpretation).
  - **Identity:** Each IoT device has a unique identity for tracking and status querying.
- **3. Scalability:** IoT setups must handle massive expansion in connected elements and the enormous data generated.
- **4. Dynamic and Self-Adapting (Complexity):** Devices should dynamically adapt to changing contexts/scenarios (e.g., a surveillance camera adapting to different light conditions).
- **5. Architecture:** IoT architecture is typically hybrid, supporting products from different manufacturers. It's a multi-domain reality.
- **6. Safety:**
  - **Data Security:** Major challenge due to the risk of compromising sensitive personal user details.
  - **Equipment Safety:** Physical safety of the (often large-scale) equipment and network is critical.
- **7. Self-Configuring:** Devices can upgrade software with minimal user participation and set up networks, allowing new devices to join existing ones.

6. Advantages of IoT

- Smarter control of homes/cities via mobile phones; enhances security and personal protection.
- Saves time by automating activities.
- Easy accessibility to information, updated in real-time, even remotely.
- Efficient electricity use as devices communicate with controllers (e.g., cell phones).
- Personal assistance via IoT apps (e.g., alerts for regular plans).
- Enhanced safety by sensing potential dangers and warning users (e.g., GM OnStar for car crashes).
- Minimizes human effort as devices communicate and perform tasks autonomously.
- More effective real-time patient care without requiring a doctor's visit.

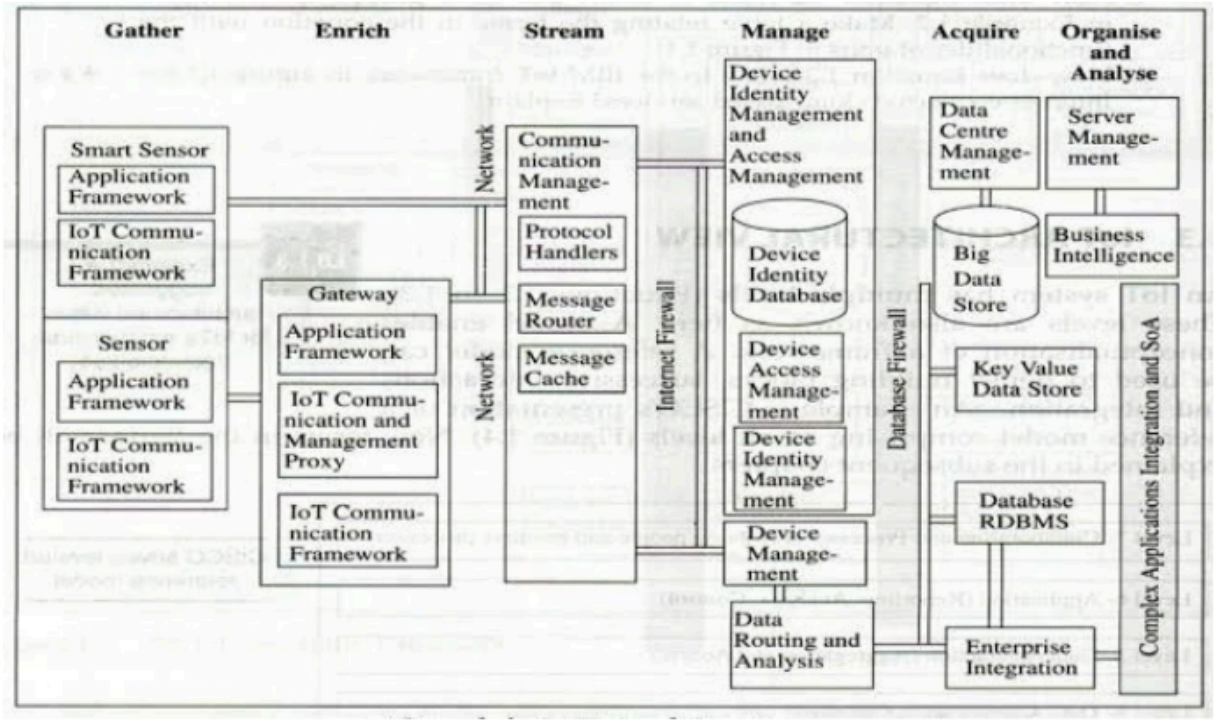
- Cost-effective asset tracking, traffic/transportation tracking, inventory control, etc.

7. Disadvantages of IoT

- Hackers can gain access and steal personal information; misuse of data from many connected devices.
- Heavy reliance on internet connectivity; ineffective without it.
- Complexity of systems leads to many potential points of failure.
- Potential loss of control over lives, becoming fully reliant on technology.
- Overuse can lead to reduced physical work and laziness.
- Risk of job loss for unskilled workers due to automation (e.g., smart surveillance replacing guards).
- Difficult to plan, build, manage, and enable a broad IoT framework.
- Deploying IoT devices can be costly and time-consuming.

8. IoT Conceptual Framework [PYQ]

- **Simple Equation:** Physical object + Controller, Sensor and Actuators + Internet = Internet of Things
- **Process Flow (Oracle's Suggested Architecture):** Gather + Enrich + Stream + Manage + Acquire + Organise and Analyse = Internet of Things (with connectivity to data center, enterprise, or cloud server).

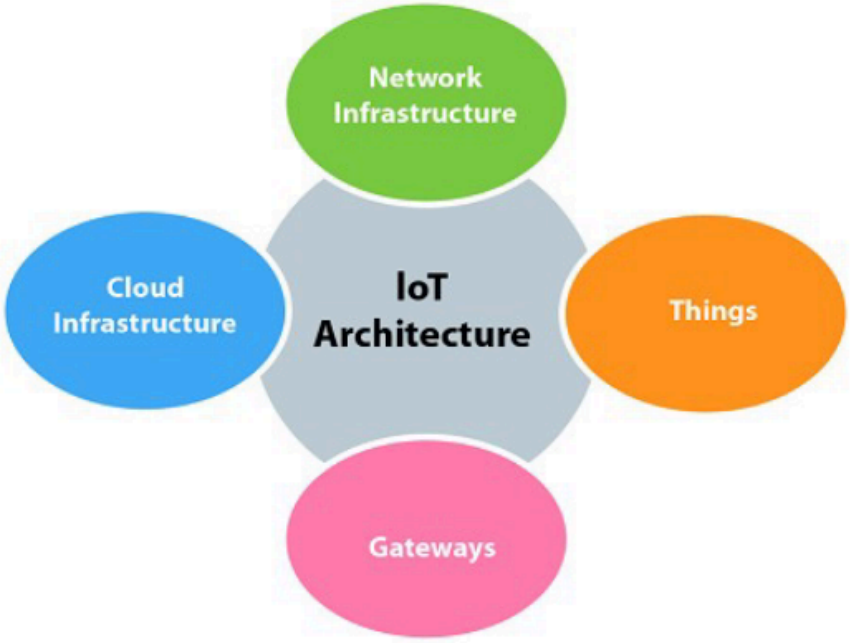


  - **Level 1 - Gather:** Devices (things) gather pre-data from the internet or using sensors.
  - **Level 2 - Enrich:** Data is enriched (e.g., transcoding at a gateway). A sensor connected to a gateway acts as a smart sensor. Transcoding is coding/decoding before data transfer.
  - **Level 3 - Stream:** Communication management subsystem sends/receives data streams.
  - **Level 4 - Manage:** Device management, identity management, access management subsystems receive device data.

    - **Level 5 - Acquire:** Data store or database acquires the data.
    - **Level 6 - Organise and Analyse:** Data is routed, organized, and analyzed (e.g., for business intelligence).
- **General Framework (Cloud-based):** Gather + Consolidate + Connect + Collect + Assemble + Manage and Analyse = Internet of Things (with connectivity to cloud services).
    - **Level-1 Gather:** Information gathered via device through sensor and internet.
    - **Level 2 Enrich:** Gathered information improved by processes like transcoding (encoding/decoding) and acts as a gateway between 2 devices.
    - **Level 3 Stream:** Transmission/reception of data stream managed by communication subsystem.
    - **Level 4 Managed:** Device data received by device management and access management subsystem.
    - **Level 5 Acquire:** Information stored in the data repository.
    - **Level 6 Organise and Analyse:** Responsible for organizing and analyzing the data set by the object.

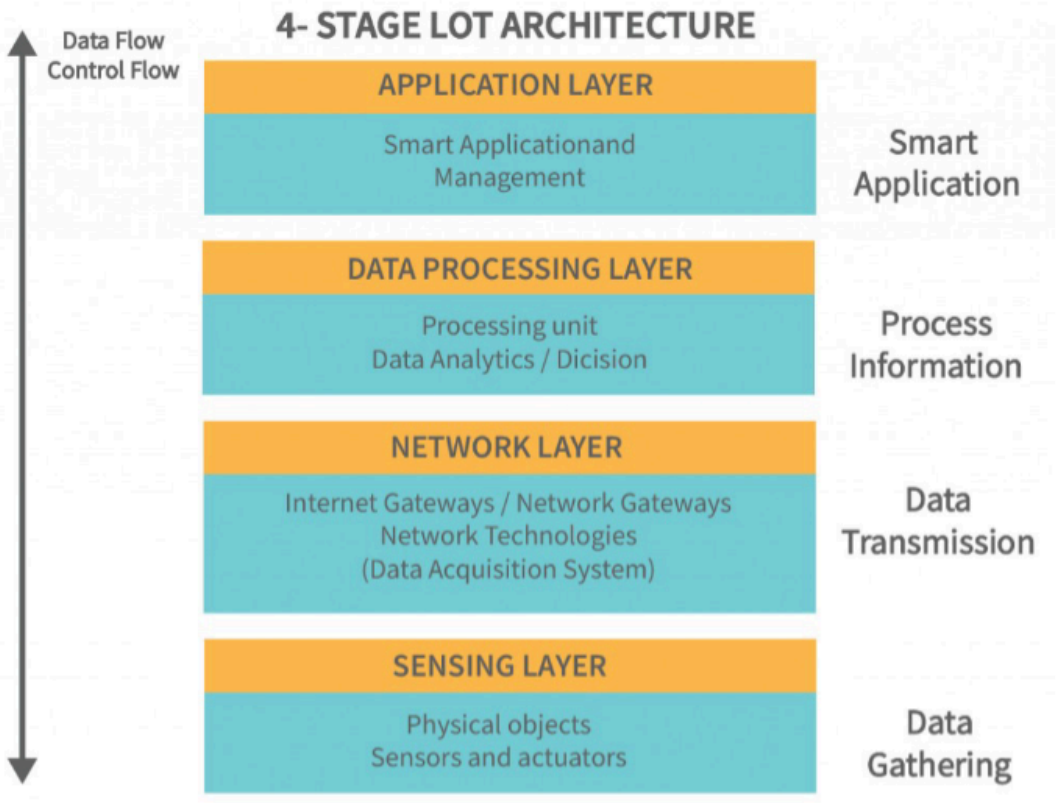
9. What is IoT Architecture? [PYQ for smart city framework]

- **Definition:** Framework defining how IoT elements (devices, networks, sensors, apps) interact. Consists of layers/components performing functions from data acquisition to processing and storage.



- **Components:** Tangle of sensors, actuators, cloud services, protocols, and layers.
- **Purpose:** Allows administrators to evaluate, monitor, and maintain system integrity.
- **Standard Process:** Typically a four-step process for data flow: devices -> network -> cloud (for processing, analysis, storage).
- **No Universal Standard:** Complexity and number of layers vary by business task. A four-layer architecture is widely accepted.

10. Different Layers of IoT Architecture (Standard 4-Layer Model) [PYQ for smart city framework]

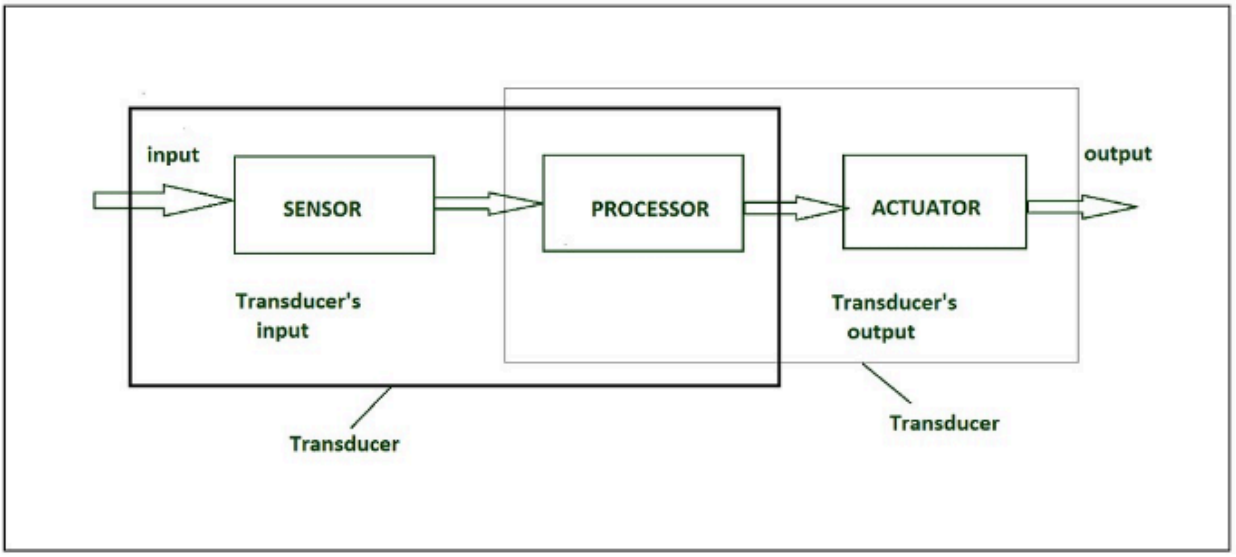


- **1. Perception/Sensing Layer (Physical Layer):**
  - Involves "things" or endpoint devices bridging physical and digital worlds.
  - Includes sensors and actuators collecting, accepting, and processing data.
  - Connects via wired/wireless means.
  - Gathers information like temperature, humidity, light, sound.
- **2. Network Layer (Connectivity Layer):**
  - Moves data throughout the application.
  - Contains Data Acquiring Systems (DAS) for aggregation/conversion (analog to digital) and Internet/Network gateways.
  - Connects devices to servers, smart devices, other network devices. Handles data transmissions.
  - Protocols: WiFi, Bluetooth, Zigbee, cellular (4G/5G).
  - Components: Gateways, routers. May include security (encryption, authentication).
- **3. Processing Layer (Middleware/Data Processing Layer):**
  - The "brain" of the IoT ecosystem.
  - Data is analyzed, pre-processed, and stored before potentially going to a data center.
  - Edge IT / edge analytics often occurs here.
  - Receives raw data, processes it for further analysis/action.
  - Tools: Data management systems, analytics platforms, machine learning algorithms.

- **4. Application Layer (User Interaction Layer):**
  - Delivers application-specific services to the user.
  - User interaction takes place here (e.g., smart home app to control devices, dashboards).
  - Examples: Smart cities, smart homes, smart health.
  - Provides user-friendly interfaces (mobile apps, web portals) to access/control IoT devices.

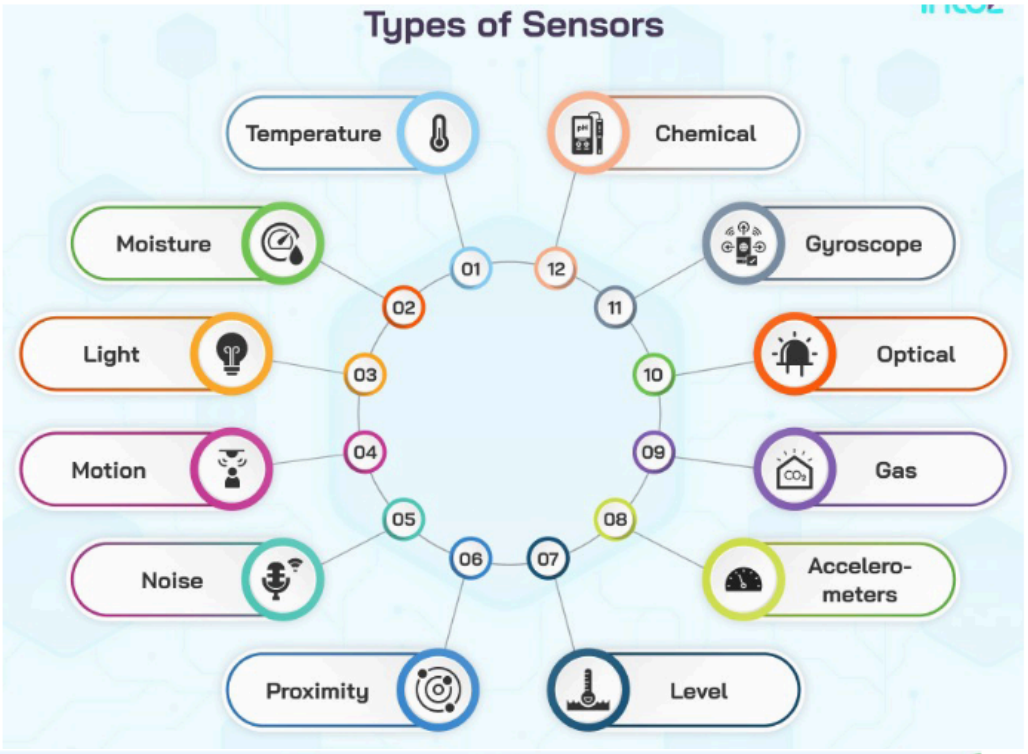
11. Sensors [PYQ for difference with actuators/transducers]

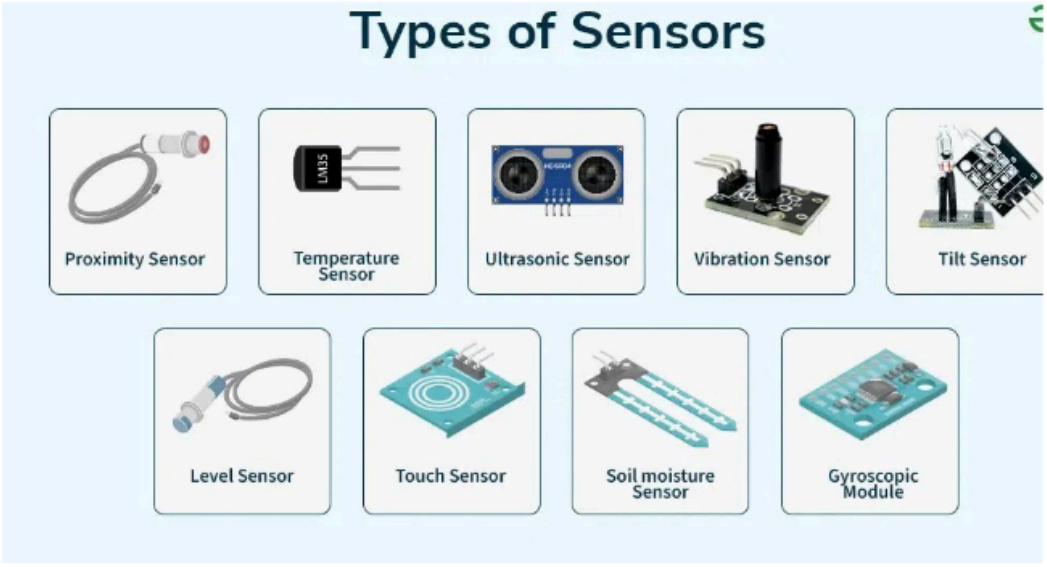
- **Definition:** A device that converts physical events or characteristics into electrical signals. It's a hardware device taking input from the environment.



  - *Example:* Thermometer converts temperature (physical) into electrical signals.
- **Role in IoT:** Bridge the gap between the physical and logical worlds, enabling data collection.
- **Transducer:** Converts signals from one physical form to another (energy converter). *Example:* Microphone converts sound to electrical signals. Based on conservation of energy. [PYQ]
- **Classification of Sensors:**
  - **Based on Power Requirement:**
    - **Active Sensors:** Require an external excitation signal or power source.
    - **Passive Sensors:** Do not require external power; directly generate output response.
  - **Based on Means of Detection:** Electrical, biological, chemical, radioactive.
  - **Based on Conversion Phenomenon:**
    - **Photoelectric:** Light to electrical signals.
    - **Thermoelectric:** Temperature difference to electrical voltage.
    - **Electrochemical:** Chemical reactions to electrical signals.
    - **Electromagnetic:** Magnetic fields to electrical signals.
    - **Thermoptic:** Temperature changes to electrical signals.
  - **Based on Output Type:**

    - **Analog Sensors:** Output signal (voltage, current, resistance) proportional to measured quantity.
    - **Digital Sensors:** Provide discrete or digital data as output.
- **Types of Sensors (Examples from text):**



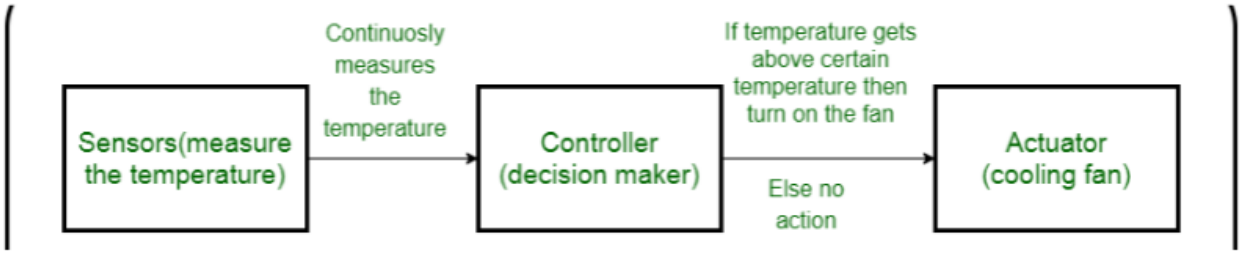


    - **Temperature:** Records heat amount. (Described in two places with similar info)
    - **Moisture:** Records humidity.
    - **Light:** Records ambient light; used in smartphones, automated lighting.
    - **Motion:** Detects unauthorized activity; used in security systems (radar, infrared, ultrasonic).
    - **Noise:** Records noise levels for safe environments.
    - **Proximity:** Records nearby activity using electromagnetic waves (infrared). Used in cars, retail.
    - **Level:** Detects quantity/level of substances (granular to liquid).
    - **Accelerometers:** Measure object's acceleration and changes in gravity. Used in fleets, pedometers. [PYQ - smart sensor context]

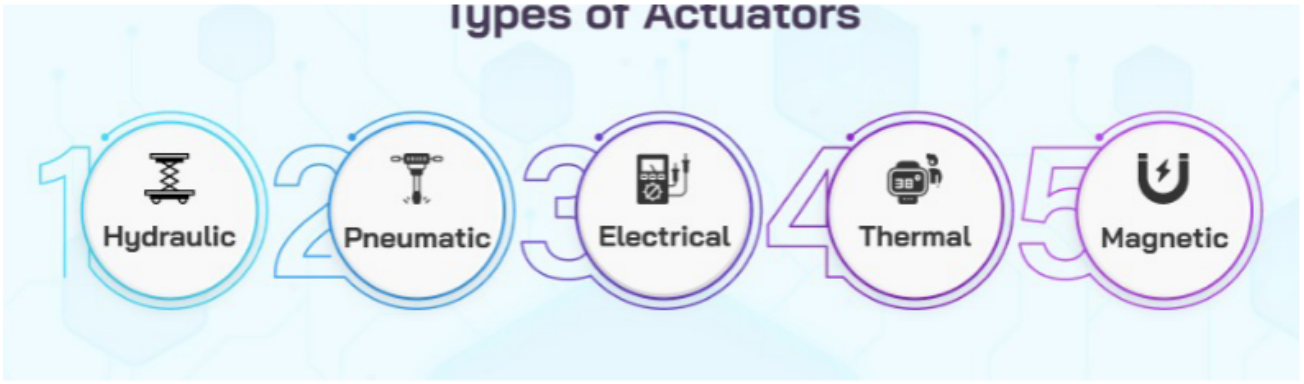
- **Gas:** Detects changes in air quality, presence of toxic/hazardous gases.
- **Optical:** Detect signals/signs for environmental information (driverless cars).
- **Gyroscope:** Measures velocity (speed and rotation) of a moving object.
- **Chemical:** Detects/measures various chemicals (similar to gas sensors).
- **(Other listed):** Alcohol, Radiation, Position, Torque, Touch, Image sensors.

12. Actuators [PYQ for difference with actuators/transducers]

- **Definition:** A device that converts electrical signals into physical events or characteristics. Takes input from the system and gives output to the environment.

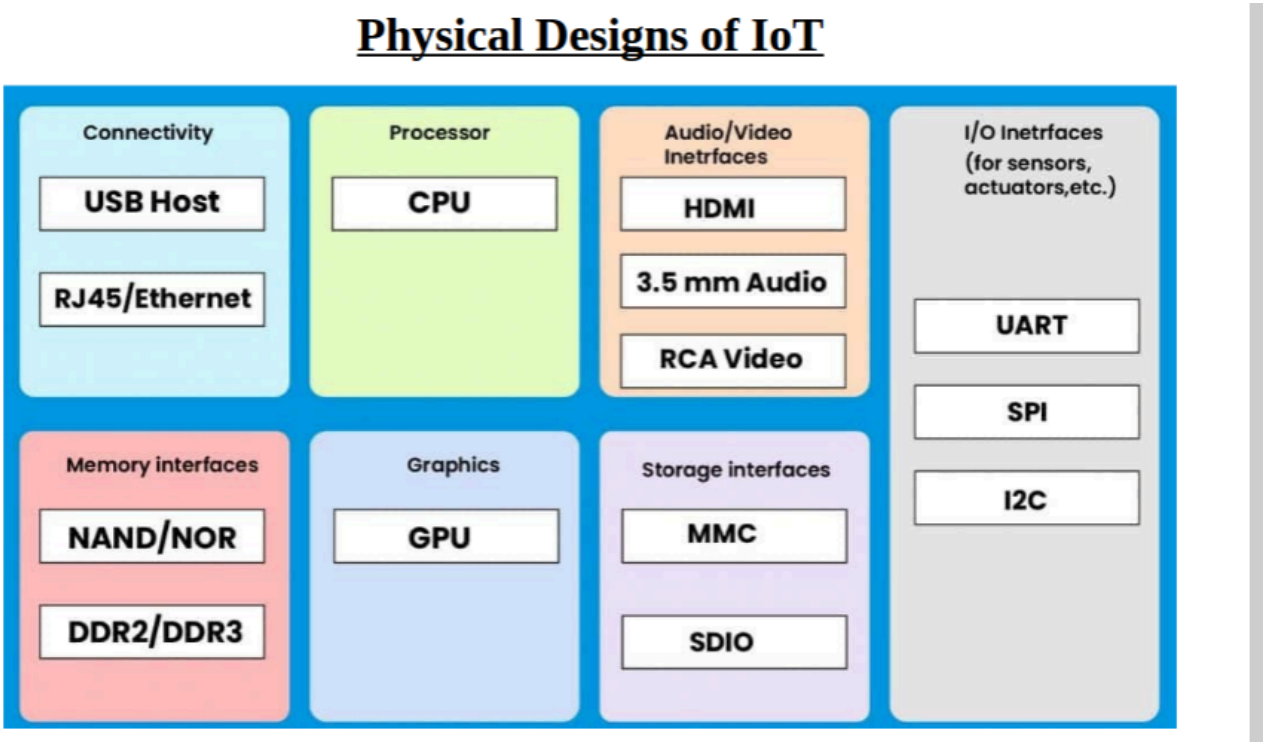


  - *Examples:* Motors, heaters.
- **Function:** Control system acts on the environment through the actuator. Requires energy source and control signal.
- **Types of Actuators:**



  - **Hydraulic:** Use hydraulic power (cylinder/fluid motor) for mechanical motion (oscillatory, linear, rotary).
  - **Pneumatic:** Use vacuum/compressed air for rotary/linear motion. Low-cost, low-maintenance.
  - **Electrical:** Motor converts electrical energy to mechanical motion. Provide precision control.
  - **Thermal:** Use thermal-sensitive material for linear motion in response to temperature changes (e.g., shutting valves).
  - **Magnetic:** Convert electromagnetic energy to mechanical output (linear/rotary). Continuous operation (automotive, aerospace).

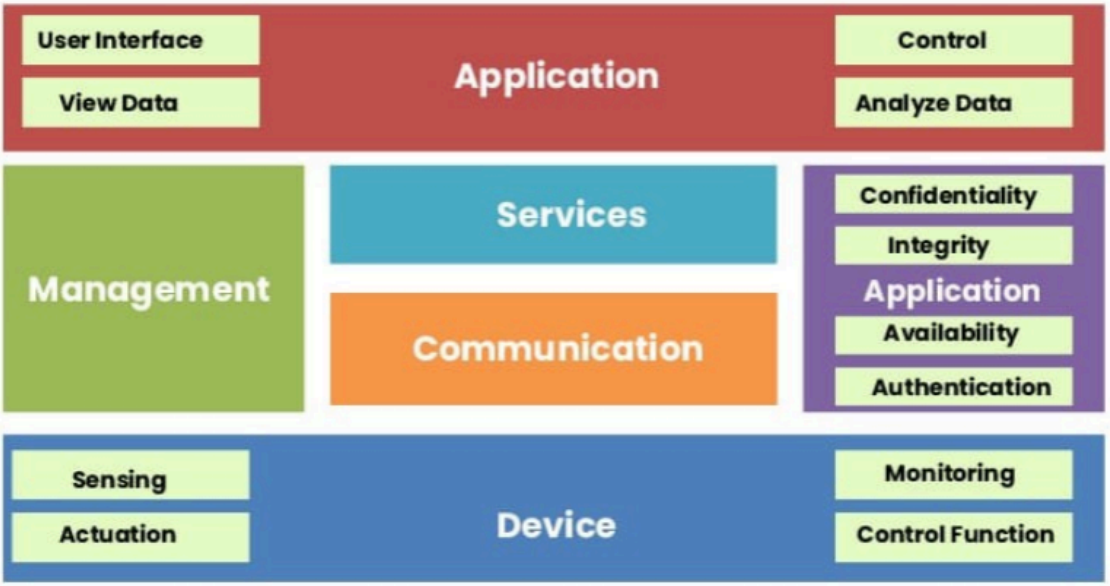
13. Physical Design of IoT (Generic Components)



- **Connectivity:** USB hosts, ETHERNET for device-server connection.
- **Processor:** CPU and other units to process data for decision quality.
- **Audio/Video Interfaces:** HDMI, RCA for recording audio/video.
- **Input/Output Interface:** UART, SPI, CAN for sensor/actuator signals.
- **Storage Interfaces:** SD, MMC, SDIO to store data from IoT devices.
- **Memory/Control:** DDR (Double Data Rate memory), GPU (Graphics Processing Unit) for controlling IoT system activity.

14. Logical Design of IoT

- **Definition:** High-level design of how components (computers, sensors, actuators) are arranged for a specific function, without low-level programming specifics.



- **Functional Blocks:**

- **Device:** Sensing, actuation, monitoring, control functions.
- **Communication:** Handles communication for the IoT system.
- **Services:** Device monitoring, control, data publishing, device discovery.
- **Management:** Governs the IoT system.
- **Security:** Authentication, authorization, message/content integrity, data security.
- **Application:** User interface to control/monitor IoT system, view/analyze processed data.

- **Key Components (Elaborated):**
  - **IoT Functional Blocks:** (As listed above)
  - **IoT Communication Models:** (Request-Response, Push-Pull, Publish-Subscribe, Exclusive Pairs) - *Details in a separate section below.*
  - **IoT Communication APIs:** Facilitate communication via server and IoT system (e.g., client-server models, stateless communication, cacheable interfaces).
  - **IoT Protocols:** Guiding principles for data exchange, enabling interoperability.

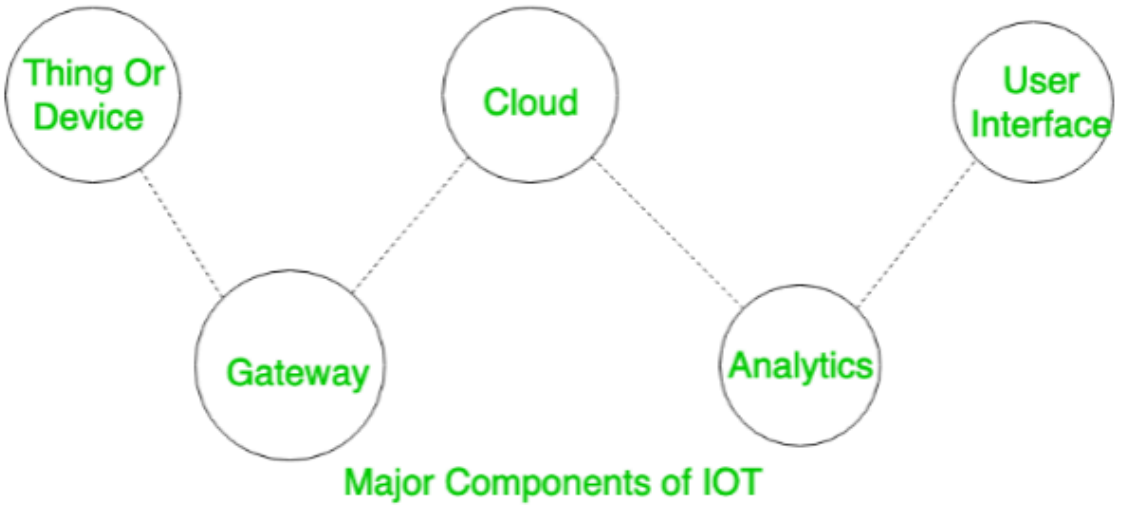
15. Difference Between Physical and Logical Design of IoT (Tabular)

Difference Between Physical and Logical Design of IoT

Physical Design	Logical Design
Physical design is highly detailed.	Logical design is a high-level design and doesn't provide any detail.
Physical design is more graphical than textual; however, it can comprise both.	Logical design can be textual, graphic, or both.
A physical design focuses on specific solutions explaining how they are assembled or configured.	A logical design focuses on satisfying the design factors, including risks, requirements, constraints, and assumptions.

Feature	Physical Designs of IoT	Logical Designs of IoT
Overview Detail	Elaborate and detailed.	High-level and brief.
Emphasis	Configuration and assembling of specific entities.	Design factors: assumptions, requirements, constraints, risk.
Content Composition	More graphic content than textual.	Comprises both textual and graphic content.

16. Major Components of IoT (Simplified View)



- **1. Things or Devices (Sensors/Actuators):** Collect data (sensors) from environment or perform actions (actuators) after data processing.
  - *Sensor Examples:* Temperature, Humidity, Proximity, Motion, Light, Pressure, Gas, GPS.
- **2. Gateway:**
  - Receives data from sensors; can perform pre-processing.
  - Acts as a security layer for network/transmitted data.
  - Intermediate between sensors and central cloud.
  - *Functions:* Data Aggregation, Communication, Security, Protocol Translation, Load Balancing, Latency Reduction. [PYQ for gateway importance]
- **3. Cloud:**
  - Collected data is uploaded here.
  - Set of servers connected to the internet 24/7.
  - Provides management, storage, processing of IoT-generated data.
  - *Key Aspects:* Data Storage, Data Collection, Security, Connectivity, Integration, Cost Efficiency.
- **4. Analytics:**
  - Data processing occurs in the cloud using various algorithms (e.g., Machine Learning).
  - Crucial for harnessing IoT's potential; extracts meaningful insights.
  - *Functions:* Data processing, machine learning, statistical analysis.
  - *Applications:* Anomaly Detection, Environmental Monitoring, Energy Management, Smart Cities, Agriculture.
- **5. User Interface (UI):**
  - User-end application to monitor or control data.
  - Interface for users to interact with applications/systems.

- *Key Points:* Data Visualization, User-Friendly Design, Personalization, Remote Management, Integration, Authentication, Security.

17. Technology Behind IoT

- **Sensors and Actuators:** Collect data and perform actions.
- **Connectivity:**
  - Wireless: Wi-Fi, Bluetooth, Zigbee, cellular (4G/5G).
  - Wired: Ethernet.
- **Communication Protocols:** [PYQ for specific protocols]
  - Lightweight: MQTT (Message Queuing Telemetry Transport), CoAP (Constrained Application Protocol) for efficient data transfer with limited bandwidth/power.
- **Edge Computing:** Processes data closer to the source (IoT devices) to reduce latency and improve real-time decision-making. Allows local analysis/filtering.
- **Cloud Platforms:** Amazon Web Services (AWS), Microsoft Azure, Google Cloud provide scalable infrastructure for storing, processing, analyzing IoT data and offer IoT-specific services.
- **Integration and Interoperability:** Middleware platforms and APIs facilitate integration between diverse IoT devices, protocols, applications, ensuring seamless data exchange.
  - *Example:* Middleware translating data from different sensors into a common format.

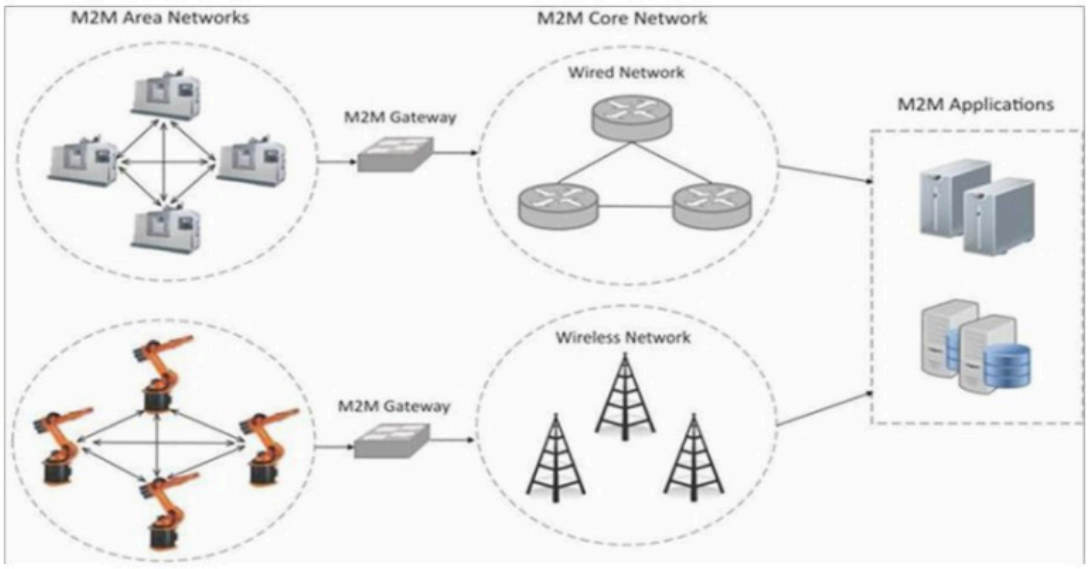
18. Sources of IoT (Connectivity Technologies - Brief Overview)

- **Wi-Fi:** Connects to LANs/internet wirelessly; high-speed. *Example:* Smart home devices.
- **Bluetooth:** Short-range wireless communication. *Example:* Wearable fitness trackers.
- **Zigbee and Z-Wave:** Low-power wireless for smart home mesh networks. *Example:* Smart lighting systems.
- **Cellular Networks (3G/4G/5G):** Wide-area connectivity. *Example:* GPS trackers for vehicles.
- **Satellite Communication:** For remote/rural areas with limited terrestrial connectivity. *Example:* Environmental monitoring in remote regions.

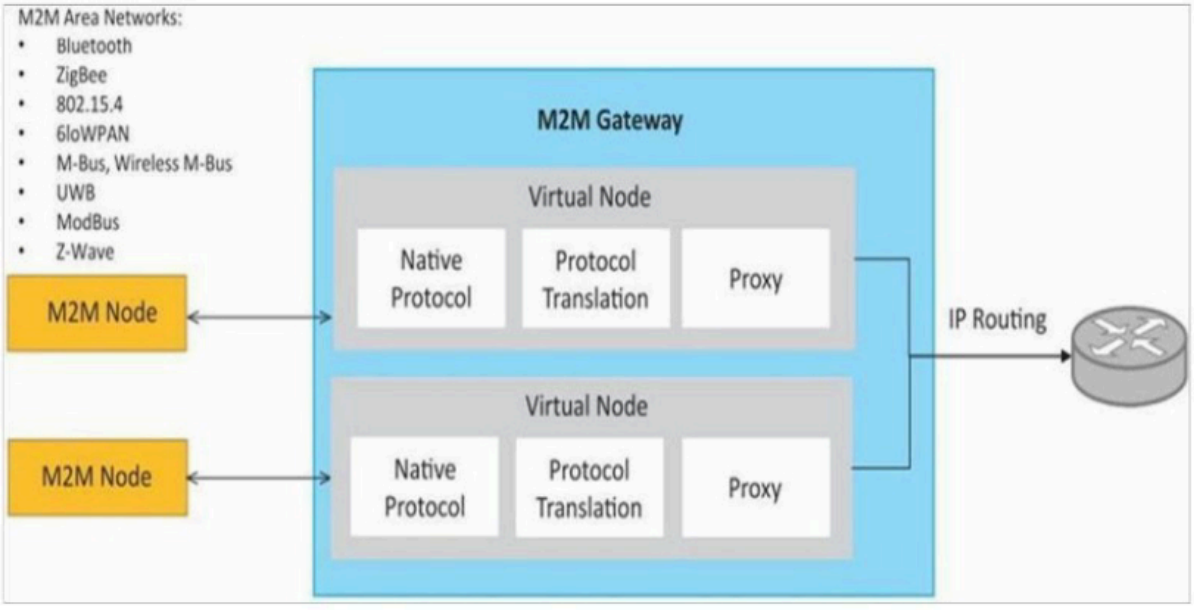
19. M2M (Machine-to-Machine) Communication

- **Definition:** Direct communication between devices (wired or wireless) without human intervention. Collects and shares data with other machines.
  - *Example:* Controlling AC with a smartphone via Bluetooth at home is M2M. Over internet from far away is IoT.
- **IoT and M2M:** [PYQ for M2M arch. vs IoT levels]
  - M2M is often considered a subset or foundation of IoT. IoT expands M2M by creating large "cloud" networks of devices.
  - Term "Machine-to-Machine (M2M)" is often synonymous with IoT.

- End-to-end M2M architecture comprises M2M area networks, communication networks, and application domain.



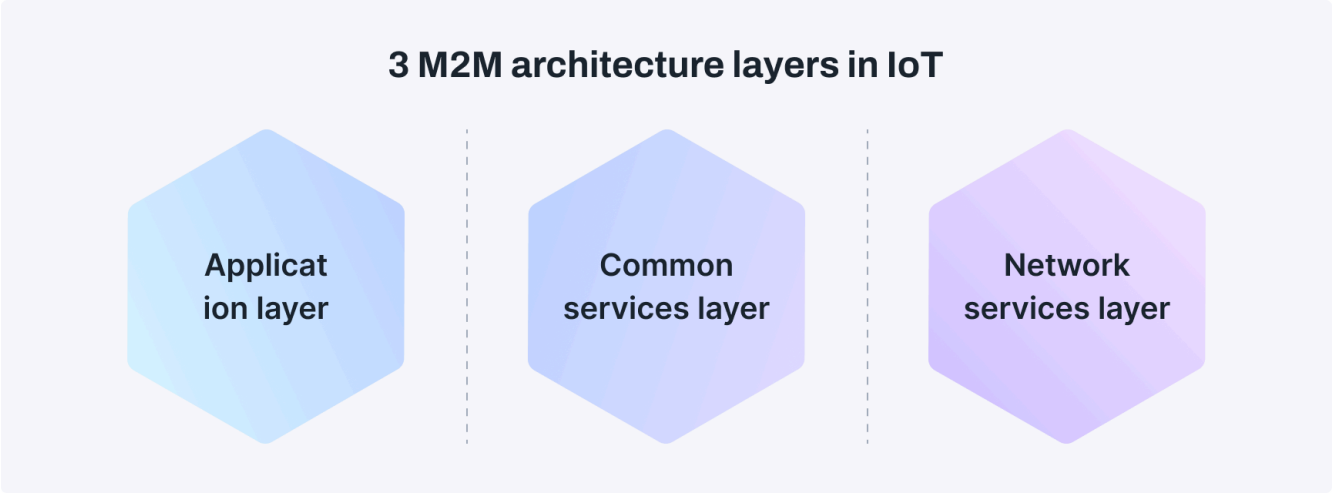
- **M2M Area Network:** Comprises M2M nodes (machines) with embedded modules for sensing, actuation, communication (e.g., ZigBee, Bluetooth, M-bus).
- **Communication Network:** Provides connectivity to remote M2M area networks (can be IP-based).
- **M2M Gateway:** Enables communication between M2M area networks (often non-IP) and external IP-based networks. Performs protocol translations. Each M2M node appears as a virtualized node externally.



- **How M2M is Used / Applications:**
  - **Remote Monitoring:** Vending machines requesting refills.
  - **Asset Tracking:** Warehouse management (WMS), supply chain management (SCM).
  - **Telecommunications:** Network performance monitoring.
  - **Home Automation:** Smart home system integration.
  - **In-car Telemetry Services:** Built-in SIM cards relay car performance data.

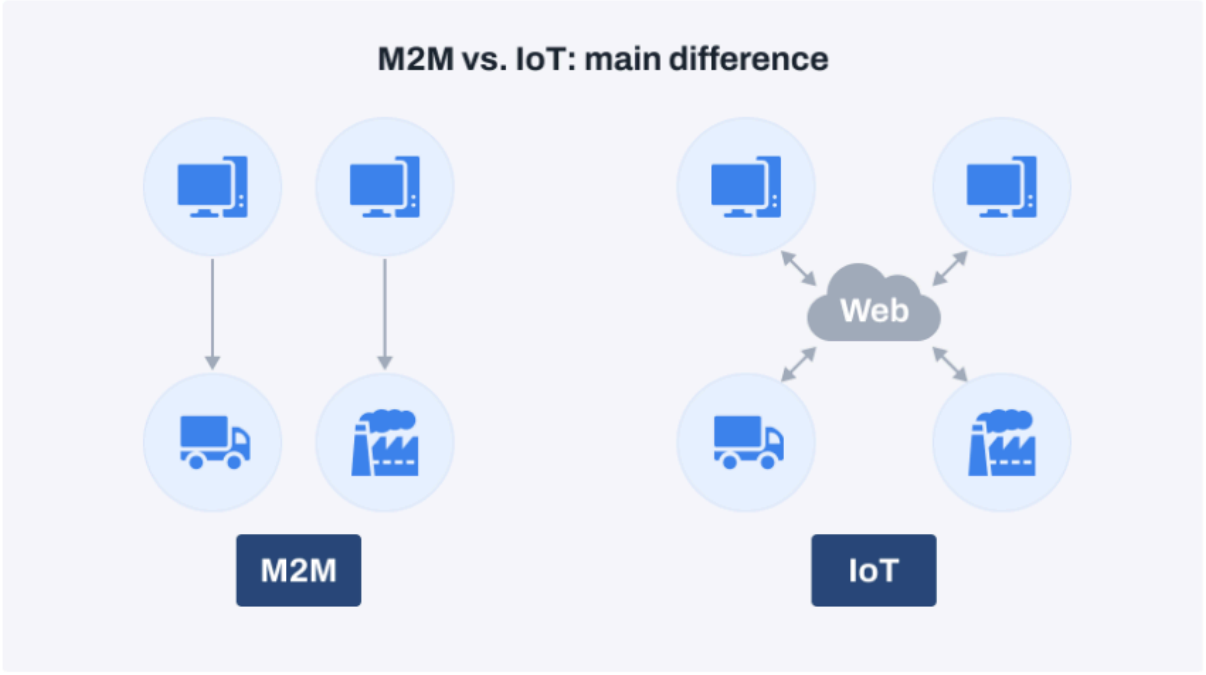
- **Smart Meters:** Track energy consumption in real-time.
- **Smart Asset Tracking Services:** GPS trackers for fleet management.

- **M2M Architecture in IoT (oneM2M perspective):**



  - **Application Layer:** Top layer; IoT applications/services reside here. Communicates with services layer. Integrates devices with apps.
  - **Services Layer (Middleware Layer):** Connects IoT devices and communication networks. Abstracts device connectivity/data transfer difficulties. Standardizes data formats/protocols for interoperability. Background control hub.
  - **Network Layer:** Where all "things" connect. Includes physical network connections (cellular, Wi-Fi). Manages connectivity/data transmission.

- **Differences between M2M and IoT (Tabular) [PYQ awareness]**



Feature	M2M	IoT
Connection type	Point-to-point, often wired/wireless, no human interaction.	Extends internet to machines; various communication types over networks.

Feature	M2M	IoT
Data sharing	Only between communicating parties in direct communication.	Shared between multiple applications for enhanced end-user experience.
Internet dependency	Can function without internet (direct point-to-point).	Requires internet connection for communication/operation.
Communication protocol	Traditional protocols and communication technologies.	Typically internet protocols (HTTP, FTP, Telnet).
Scope	Limited, specific applications/industries, point-to-point.	Broader, many devices/users in a connected ecosystem.
Intelligence	Some degree observed.	Devices have objects for decision-making.
Business Type	B2B	B2B, B2C
Examples	Sensors, data, information.	Smart cities, Big data.

Note: IoT needs M2M, but M2M does not necessarily need IoT. IoT takes M2M to the next level by integrating disparate systems.

- **Advantages of M2M:** Operates over cellular, simple to manage, indoor/outdoor use, aids smart object communication without human interaction, security/privacy feasible.
- **Disadvantages of M2M:** Cloud computing use can restrict versatility/creativity, data security/ownership concerns, interoperability challenges, requires reliable internet (for some advanced uses).
    - *Examples:* Smart washing machine alerts, smart meter energy tracking.

20. IoT Enabling Technologies

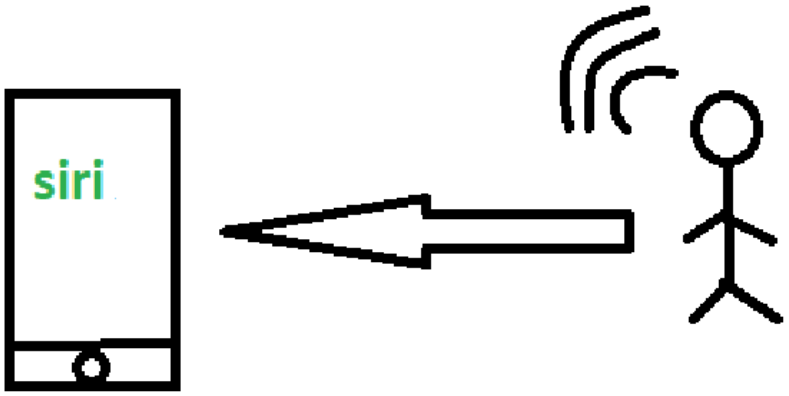
- **1. Wireless Sensor Network (WSN):**
    - Distributed devices with sensors to monitor environmental/physical conditions.
    - Consists of end nodes (with sensors), routers, and coordinators.
    - Coordinator acts as a gateway connecting WSN to the internet.
    - *Examples:* Weather/air quality/soil moisture monitoring, surveillance, health monitoring.
- **2. Cloud Computing:**
    - Access applications as utilities over the internet. Resources (databases, webservers, storage) present in remote locations.
    - *Characteristics:* Broad network access, on-demand self-services, rapid scalability, measured service, pay-per-use.
    - *Services:*

        - **IaaS (Infrastructure as a Service):** Online physical/virtual machines, servers, networking, storage. (e.g., Google Compute Engine, AWS, Azure).
        - **PaaS (Platform as a Service):** Cloud-based environment for building/delivering web apps without managing underlying hardware/software. (e.g., App Cloud, Google App Engine).
        - **SaaS (Software as a Service):** Applications delivered over the internet as a service, no local installation needed. (e.g., Google Docs, Gmail).
- **3. Big Data Analytics:**
    - Method of studying massive data volumes whose volume, velocity, or variety is too large for traditional databases.
    - Sources: Social networks, videos, digital images, sensors, sales transactions.
    - *Steps:* Data cleaning, Munging, Processing, Visualization.
    - *Examples:* Bank transactions, IoT location tracking data, E-commerce data, fitness band data.
- **4. Communications Protocols:**
    - Backbone of IoT systems, enable network connectivity.
    - Allow devices to exchange data. A group of protocols is a "protocol suite."
    - Used in: Data encoding, Addressing schemes.
- **5. Embedded Systems:** [PYQ - IDE relevance]
    - Combination of hardware and software for special tasks.
    - Includes: Microcontroller/microprocessor, memory, networking units (Ethernet/Wi-Fi adapters), I/O units (display, keyboard), storage (flash memory).
    - Collects data and sends it to the internet.
    - *Examples:* Digital camera, DVD player, industrial robots, wireless routers.
    - Prototyping often requires an Integrated Development Environment (IDE). [PYQ]

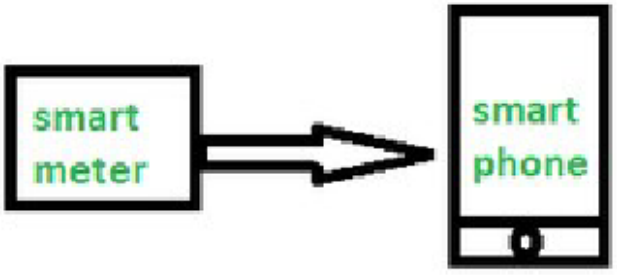
21. Types of Communications in IoT

- **Definition:** Connection of devices over the internet, communicating, exchanging data, and performing tasks without human involvement.
- **1. Human to Machine (H2M):**
    - Human gives input (speech, text, image) to IoT device.
    - Device (Machine) understands, analyzes, and responds (text, visual display).
    - Combo of software/hardware with human interaction.
    - *Merits:* User-friendly, quick fault response, customizable.

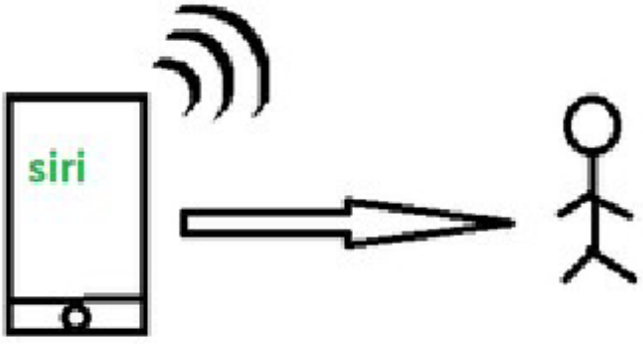
- *Examples:* Facial recognition, Bio-metric attendance, Speech/voice recognition.



- **2. Machine to Machine (M2M):** (Covered in detail in section 19)
  - Information exchange between two or more machines/devices without human intervention.
  - Automating data/programs.

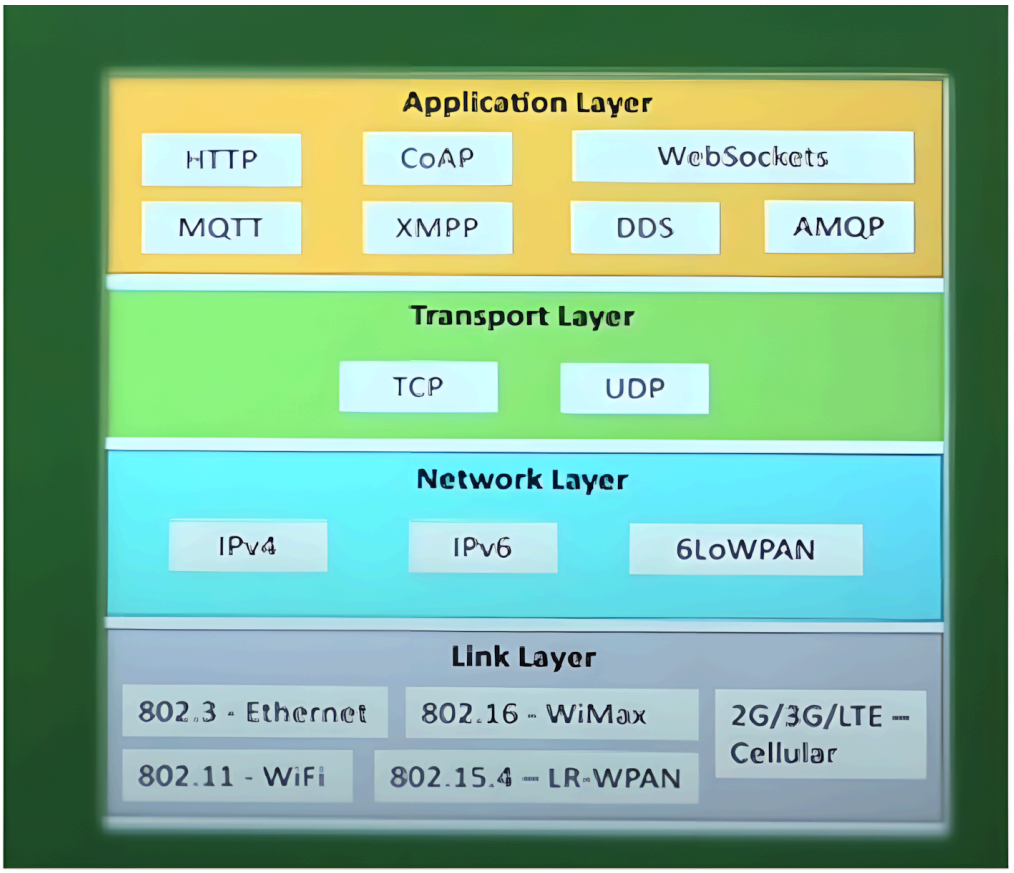


- **3. Machine to Human (M2H):**
  - Machine interacts with Humans. Machine triggers information (text, images, voice, signals) irrespective of human presence.
  - Common where machines guide humans in daily life.
  - *Examples:* Fire Alarms, Traffic Light, Fitness bands, Health monitoring devices.

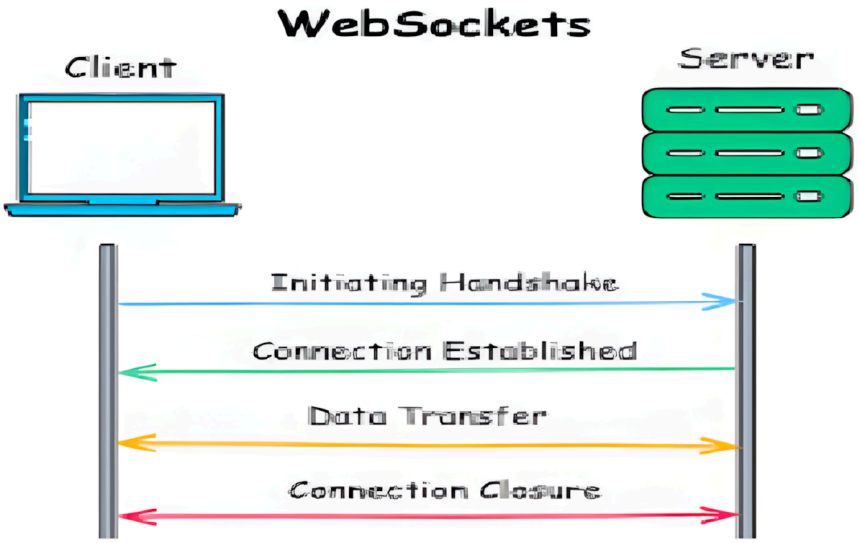


22. IoT Protocols (Detailed Breakdown) [PYQ for CoAP, HTTP, MQTT, XMPP]

- **Purpose:** Establish communication between node device and server over internet; send commands, receive data. Managed by network layers.



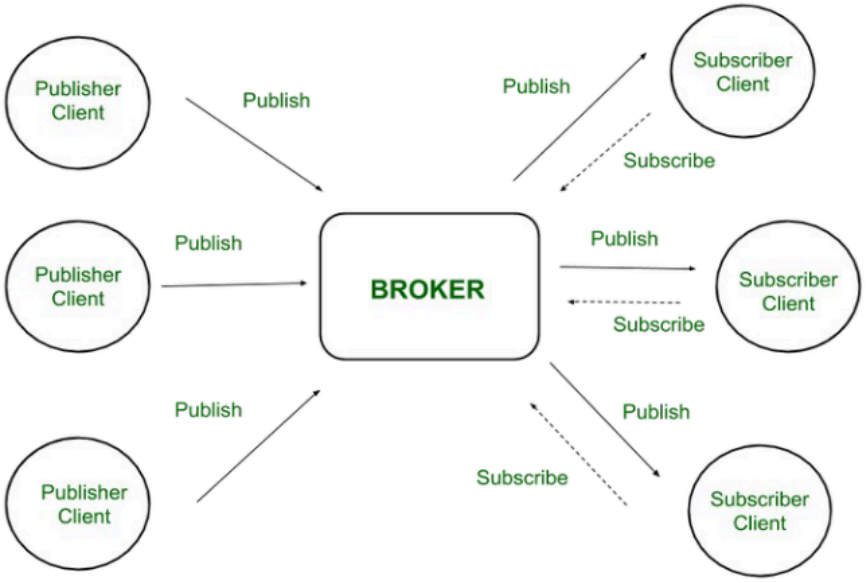
- **Application Layer Protocol:** Define how data is sent using application interface.
  - **HTTP (Hypertext Transfer Protocol):** For transmitting media documents (web). Request-response between browser/server. Stateless. [PYQ]
  - **CoAP (Constrained Application Protocol):** For constrained environments. Resembles HTTP (request-response). Uses UDP for secure M2M communication (broadcast/multicast). No prior connection needed. [PYQ]
    - *Applications:* Real-time grid monitoring, defense utilities (intrusion detection), aircraft utilities.
  - **WebSocket:** Enables two-way, persistent communication between client/host (e.g., web browsers). Uses TCP. HTTP/HTTPS handshake for connection. Supports text/binary data.



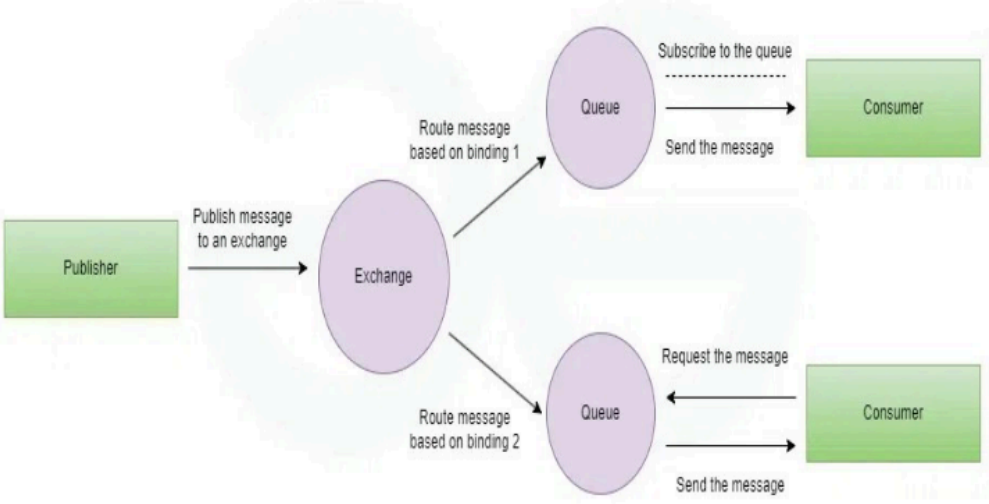
Working:

- WebSocket uses a TCP connection to create a persistent connection between a client and a server
- The client and server use an HTTP/HTTPS handshake to establish a connection
- The client sends an Upgrade: websocket header, and the server responds with 101 Switching Protocols
- The WebSocket protocol supports text and binary data formats

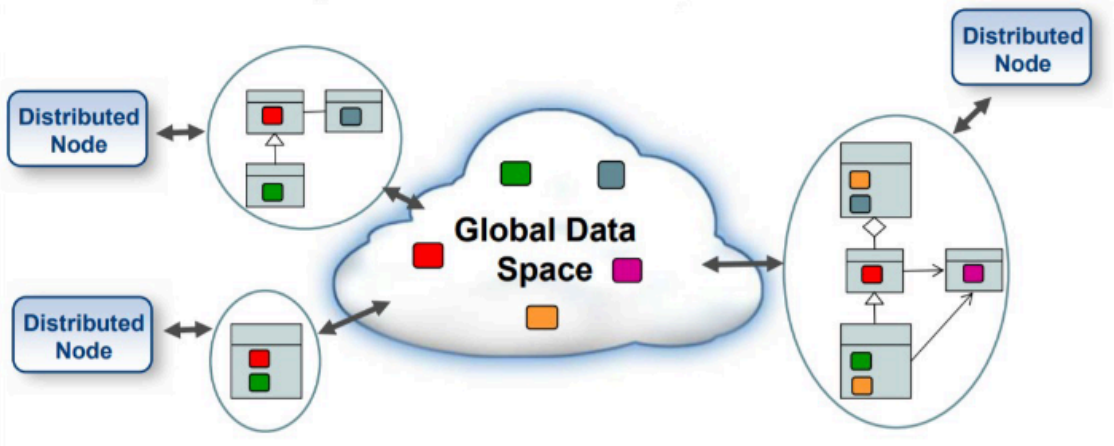
- **MQTT (Message Queuing Telemetry Transport):** Lightweight publish/subscribe messaging for remote locations, small code footprint. Uses TCP/IP. [PYQ]
  - *Architecture:* Clients (Publishers - input data, Subscribers - receive data) and Brokers (sort topics, send to subscribers).



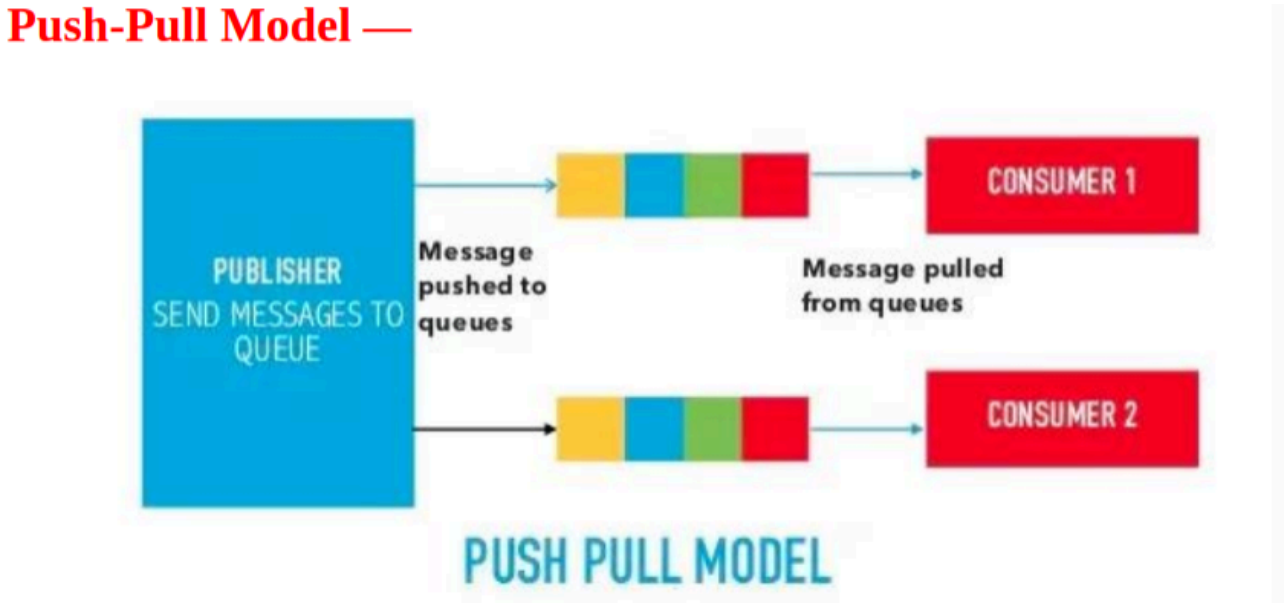
- **AMQP (Advanced Message Queuing Protocol):** Lightweight for application communication. protocol that supports the applications for data transfer, Scalable, modular.
  - *Components:* Exchanges (fetch/arrange messages), Channel (multiplexed virtual connection), Message Queue (connects messages to resources), Binding (instructions for queuing/exchanging), Virtual Host (isolation within broker).



- **XMPP (Extensible Messaging and Presence Protocol):** Open-source, XML-based for rapid structured data exchange. Real-time messaging, efficient push. Decentralized (like email). [PYQ]
  - *Uses:* Instant messaging, voice/video calls, IoT (Google Cloud Print, Logitech Harmony Hub).However, the protocol also serves IoT function properly as it's flexible for connection protocols, secure, and enables middleware communication without requiring human intervention. A few applications of IoT with XMPP include the Google Cloud Print and Logitech Harmony Hub (home automation and media control)
  - *XMPP:* eXtensible, Messaging, Presence, Protocol.
    - **X :** It means eXtensible. XMPP is an open-source project which can be changed or extended according to the need.
    - **M :** XMPP is designed for sending messages in real time. It has very efficient push mechanism compared to other protocols.
    - **P :** It determines whether you are online/offline/busy. It indicates the state.
    - **P :** XMPP is a protocol, that is, a set of standards that allow systems to communicate with each other.
- **DDS (Data Distribution Service):** Open interoperable middleware (OMG). Secure, real-time data distribution. Publisher/Subscriber (no Brokers). Topic distribution via Global Data Space (GDS) with QoS. GDS is a virtual memory concept.
- DDS is the first open interoperable middleware protocol, developed by the Object Management Group (OMG). Its operation claims to provide a secure and real-time data distribution. Like MQTT, DDS works in a Publisher/Subscriber architecture. However, the protocol doesn't implement the use of Brokers together with its Clients, hence its topic distribution occurs across its Global Data Space (GDS) by applying a QoS (Quality of Service) contract system. The GDS acts as a 'memory' during DDS transmission application. However, it's actually not a physical memory in the DDS server, it's just a virtual concept. The GDS is actually the combination of local stores in nodes connected to the system.



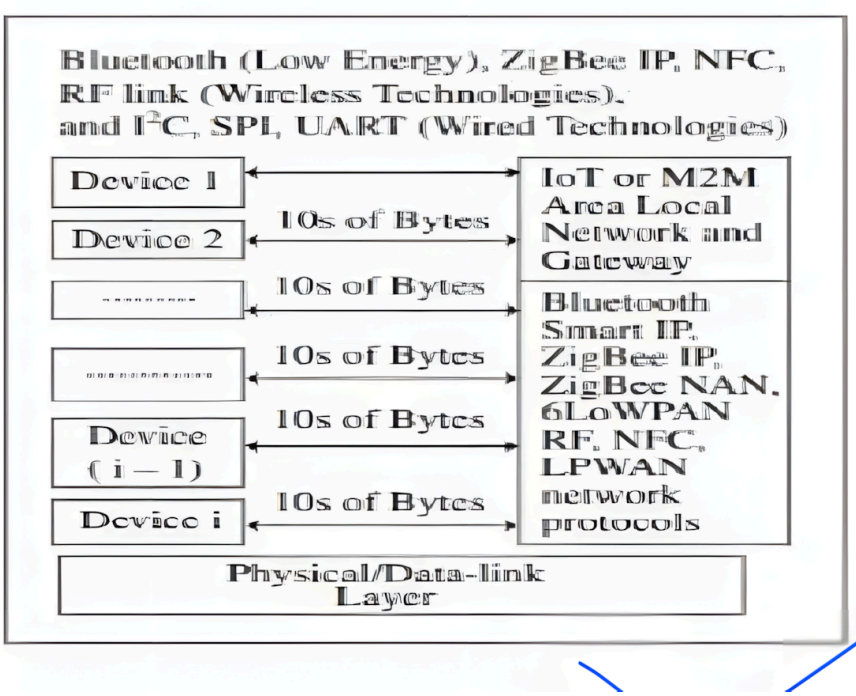
- **Transport Layer Protocol:** Controls data segment flow, error control, end-to-end message transfer.
  - **TCP (Transmission Control Protocol):** Defines how to establish/maintain a network for data exchange. Connection-oriented.
  - **UDP (User Datagram Protocol):** Connectionless protocol. No connection establishment required.
- **Network Layer Protocol:** Sends datagrams source to destination. Uses IPv4, IPv6 for host identification.
  - **IPv4:** Unique 32-bit numerical label for host/location addressing.
  - **IPv6:** Successor to IPv4, 128-bit address.
- **Link Layer Protocol:** Sends data over physical layer; determines packet coding/signaling.
  - **Ethernet:** Technologies/protocols for LANs. Defines physical layer and MAC for wired networks.
  - **WiFi:** LAN protocols for wireless local area networks (WLANs).
- **Other Communication Models:**
  - **Push-Pull Model:**



  - **Exclusive Pair Model:**

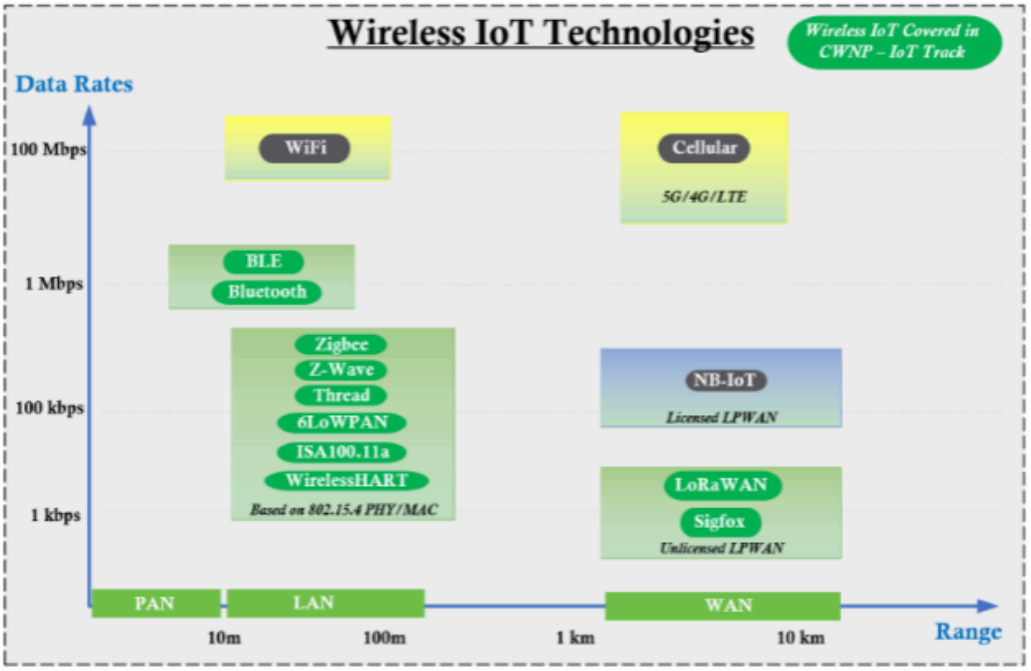
23. Communication Technologies (Detailed - from Document 4)

- Physical cum data-link layer uses wireless or wired technologies.

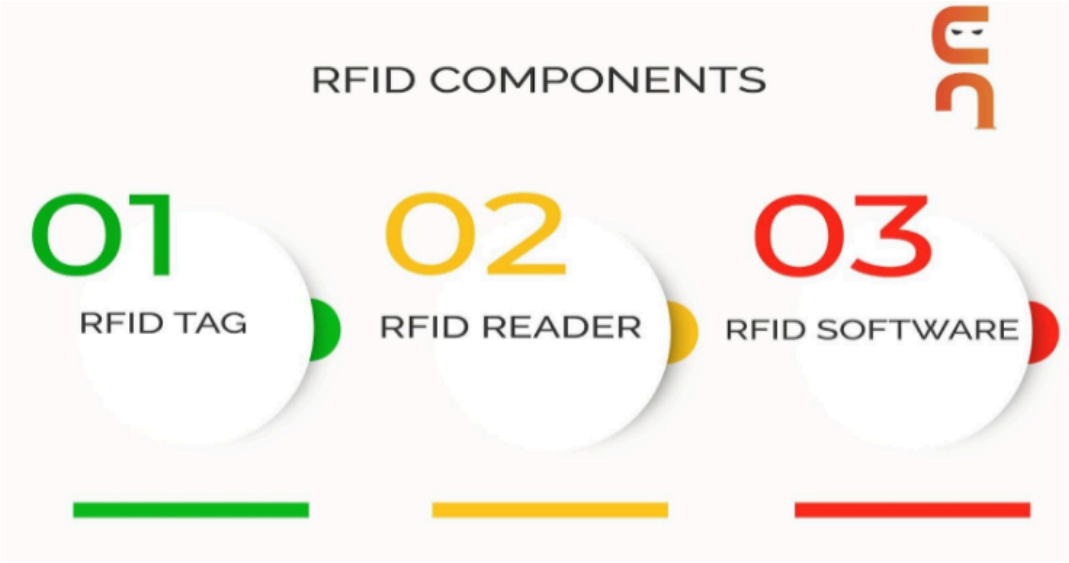


Physical cum data-link layer in the model consists of a local area network/personal areanetwork. A local network of IoT or M2M device deploys one of the two types of technologies—wireless or wired communication technologies. The following figure shows connected devices(1st to ith) connectivity using different technologies for communication of data from and todevices to the local network connectivity to a gateway. It shows number of devices present in anIoT or M2M devices network. The figure shows the local area network of devices. Theconnectivity between the devices (left-hand side) is by using RF, Bluetooth Smart Energy,ZigBee IP, ZigBee NAN (neighbourhood area network), NFC or 6LoWPAN or mobile. Tens ofbytes communicate at an instance between the device and local devices network

- **Wireless IoT Technologies:**

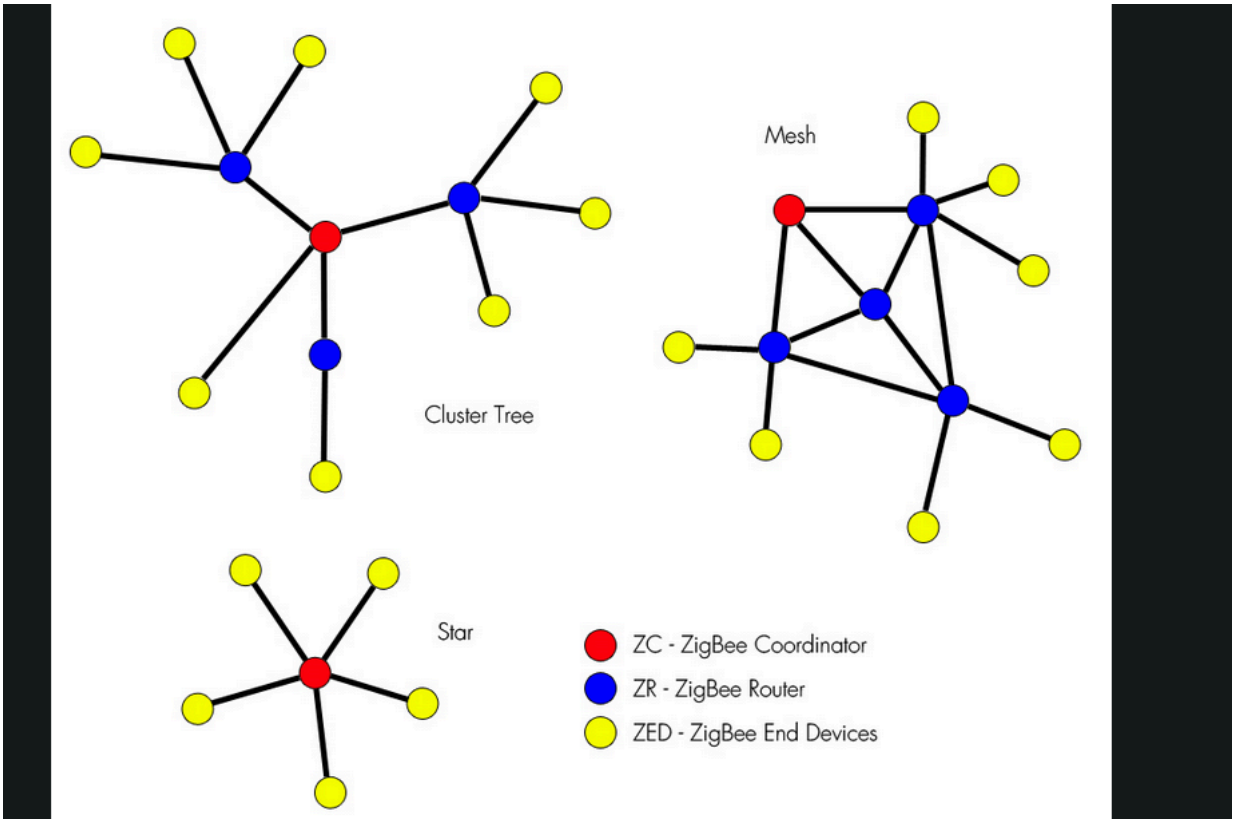


- **NFC (Near Field Communication):** [PYQ]
  - Short distance (0-5 cm, up to 20 cm), secure. Data exchange between cards/devices.
  - *Uses:* Mobile payment, e-keys, biometric passports, smart posters, access control, Bluetooth pairing.
  - *Advantages:* Security via short range, present in most smartphones, tap-to-connect, tag powered by phone, tag can harvest energy/run software.
  - *Disadvantages:* Low range, expensive, low speed, less data transfer.
- **RFID (Radio Frequency Identification System):** [PYQ]
  - Uses electromagnetic waves to capture/identify/track tags (transponder, reader, antenna). Active (computational capacity) vs. Passive tags.
  - *Working:* Reader interrogates tag, antenna transmits/receives. Used in agriculture, defense, tracking, cashless transactions. Each tag has unique ID.
  - *Components:* RFID tag (IC, substrate, antenna), RFID reader, RFID software.



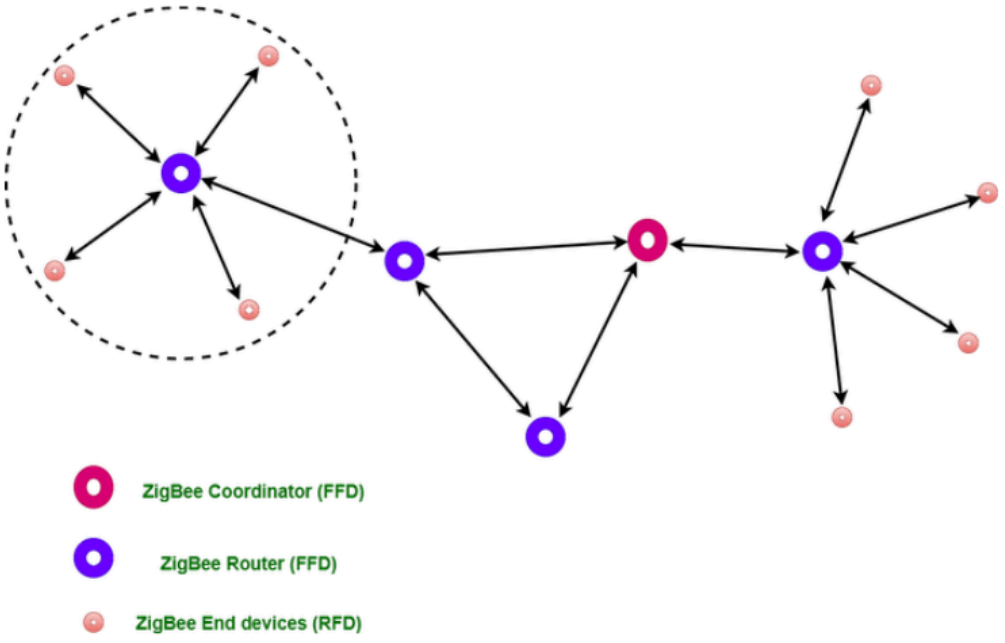
  - *Advantages:* Multiple usages, durable, more secure than barcodes.
  - *Disadvantages:* High cost, metal interference, overhead reading.
  - *Applications:* Tracking objects, livestock, humans, smart home items, patient medical info.
- **ZigBee:**
- ZigBee is a Personal Area Network task group with low rate task group 4. It is a technology of home networking. ZigBee is a technological standard created for controlling and sensing the network. As we know that ZigBee is the Personal Area Network of task group 4 so it is based on IEEE 802.15.4 and is created by Zigbee Alliance.
  - Low-rate Personal Area Network (IEEE 802.15.4 based). Home networking, control/sensing. Low cost, low power, short-range.

- *Topologies:* Star, Mesh, Cluster tree.



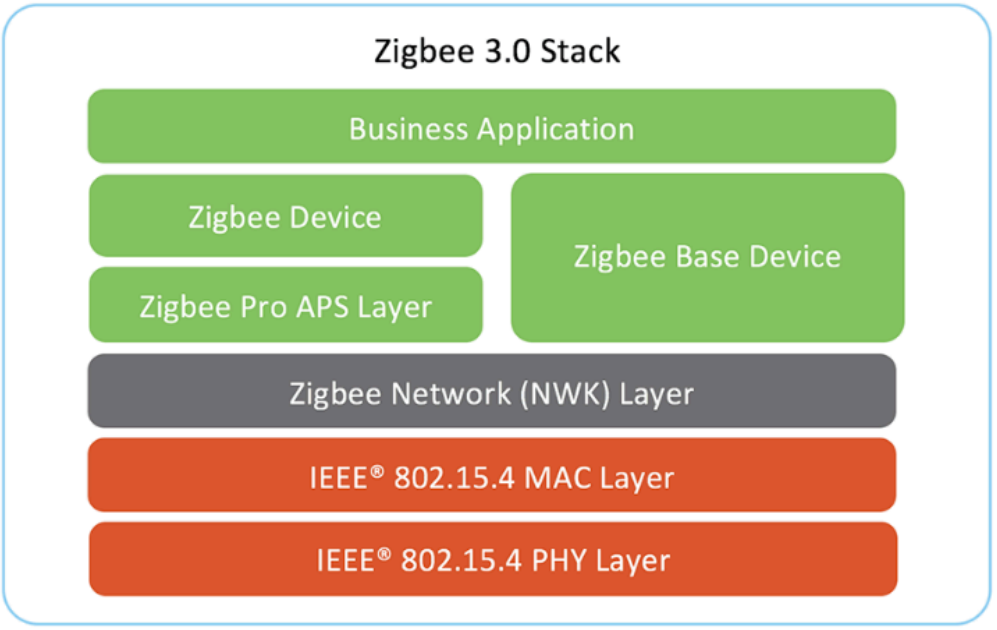
Types of ZigBee Devices:

- **Zigbee Coordinator Device:** It communicates with routers. This device is used for connecting the devices.
- **Zigbee Router:** It is used for passing the data between devices.
- **Zigbee End Device:** It is the device that is going to be controlled.

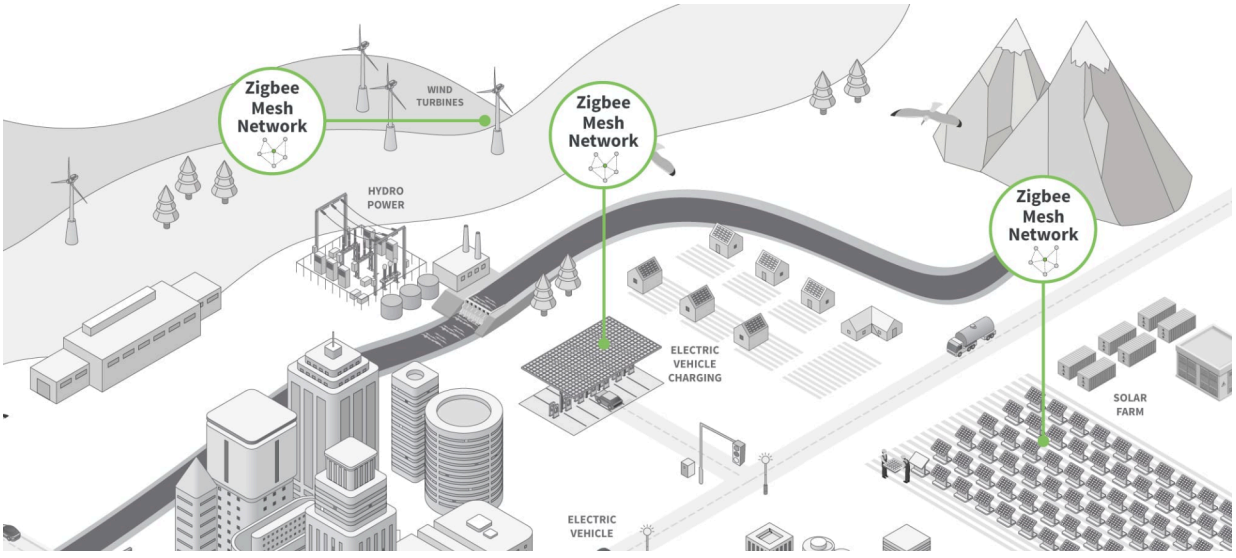


- *Device Types:* Coordinator (ZC - smart home controller), Router (ZR - mains powered, backbone), End Device (ZED - basic, battery powered).
- Zigbee devices seamlessly form a mesh network, enabling efficient data backhaul through a central node connected to a gateway for remote Internet access. A Zigbee app allows a user to control smart devices from anywhere. Zigbee is built on the Physical layer and Medium

Access Control sub-layer defined in the IEEE 802.15.4 standard which manages low-level network operations



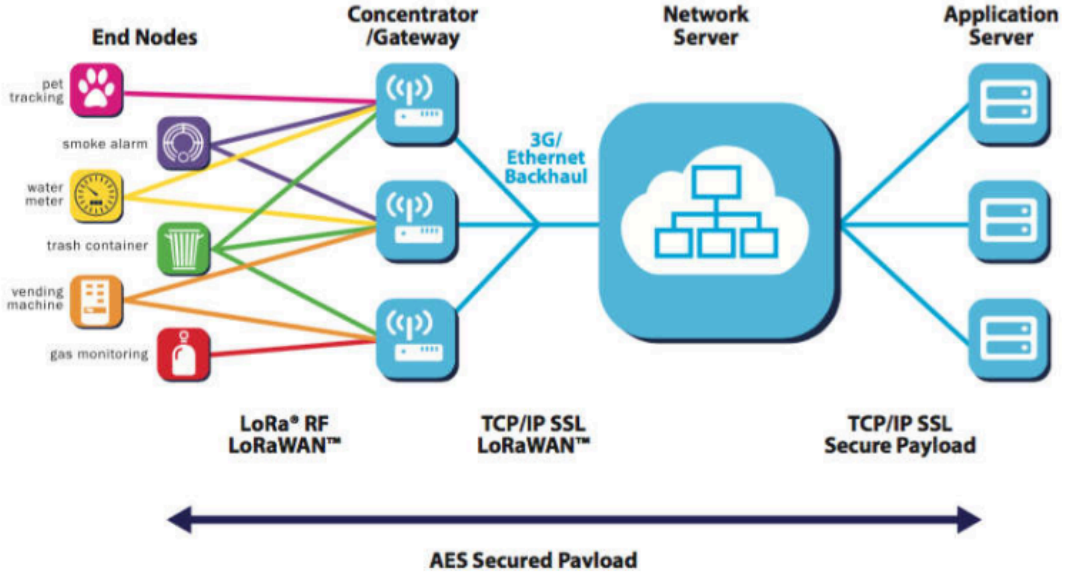
- *Applications:* Green tech (solar/wind farms, EV charging), smart home (lights, locks), smart energy, medical (patient Vitals), industrial automation.



- *Pros:* Very low power, mesh networking, designed for low data/intermittent comm, good security.

- *Cons:* Limited data rate (250 kbps), short range (10-100m), requires hub/gateway.

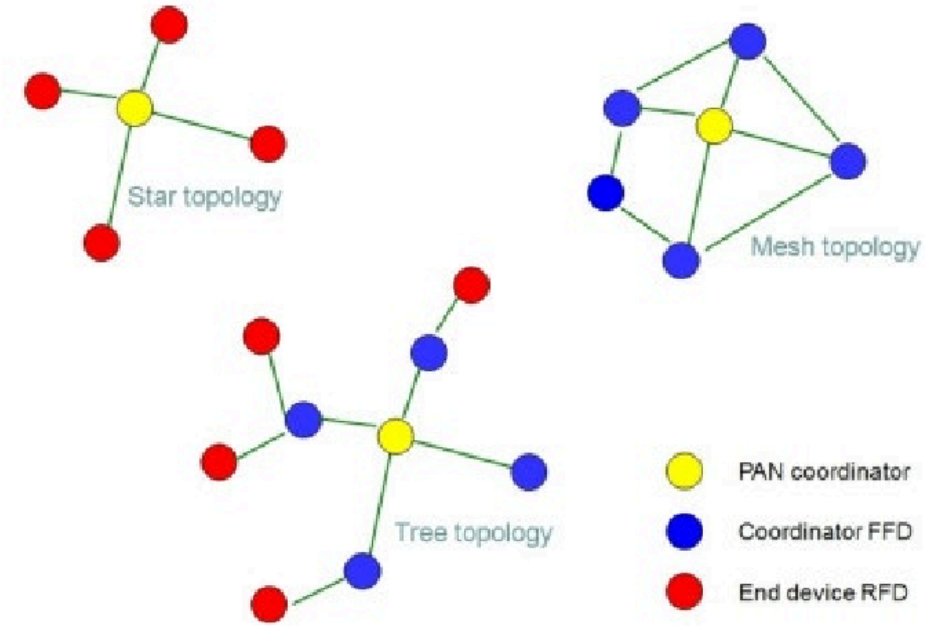
- **LoRaWAN (Long Range Wide Area Network):**
  - Communication protocol & system architecture for low-power, wide-area networks (LPWAN). Point-to-multipoint. End-to-end encryption. Range up to 10 miles.
  - Long Range Wide Area Network helps define communication protocol as well as systemarchitecture. It is a point to the multipoint communication network. acts as a gateway that doesencryption as well as identification. It has a wide range of applications like smart cities, smartindustrial control, home security systems, etc.LoRaWAN technology consists of a low-power, wide-area network protocol built onto LoRamodulation useful for securing dependable bi-directional IoT communications. The LoRaWANprotocol provides end-to-end encryption to deliver advanced security features at scale.
  - *How it works:* Chirp signal spreads spectrum, resists interference.
  - *Architecture:* End Devices (sensors/actuators), Gateways (receive/forward), Network Server (manages network), Application Servers (process data).



  - *Applications:* Smart agriculture, smart buildings, Industrial IoT, manufacturing, logistics, environmental monitoring, oil & gas.
  - *Device Classes:* Class A (bidirectional, scheduled RX windows after TX), Class B (adds synchronized RX ping slots), Class C (continuous RX, lowest latency, highest power).
  - *Pros:* Very long range (15-20km rural), extremely low power, good for limited infrastructure, scalable.
  - *Cons:* Low data rate (up to 50 kbps), typically needs gateway.

- **IEEE 802.15.4 (LR-WPANs):**
- IEEE 802.15.4 is a wireless networking technology that provides the technical specifications for low-rate wireless personal area networks (LR-WPANs), allowing networked devices to communicate with one another in a variety of industrial and commercial settings, including healthcare, environmental monitoring, smart energy, home automation, and more
  - Low-cost, low-data-rate wireless access for battery-operated devices.

- Low power wide area networks (LPWAN) provide long-range communication using small, inexpensive batteries
  - For PANs in IoT, embedded systems, WSNs. Low power, extended battery, mesh, cost-effective. 2.4 GHz band, up to 250 kbps. AES encryption.
  - *Architecture:* Physical, MAC, Networking layers. Mesh & Star topologies.



  - *Device Types:* Coordinator (initiates/manages PAN), Full Function Device (FFD - can be coordinator/router), Reduced Function Device (RFD - communicates only with FFDs).
  - *Pros/Cons/Applications:* Similar to LoRaWAN (very long range, low power, low data rate). Used in Agricultural IoT, smart cities, environmental monitoring.

Pros:

- Very long range (up to 15-20 kilometers in rural areas).
- Extremely low power consumption, enabling years of operation on a single battery.
- Suitable for environments with limited infrastructure (e.g., rural areas).
- Good scalability, capable of supporting thousands of devices.

Cons:

- Low data rate (up to 50 kbps), making it unsuitable for high-bandwidth applications.
- Typically, it requires a gateway for internet access.
- Limited interoperability with other networks.

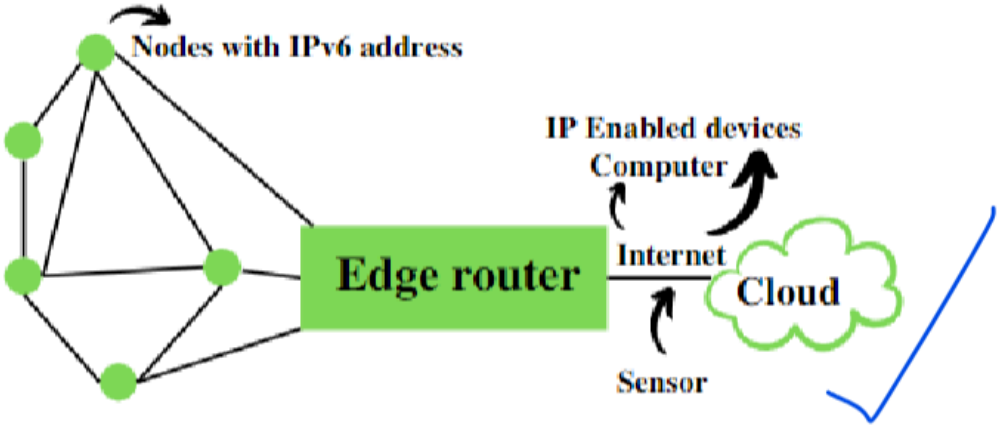
Applications:

- Agricultural IoT (e.g., soil moisture sensors, livestock monitoring).
- Smart cities (e.g., parking sensors, waste management).
- Environmental monitoring (e.g., air quality sensors, flood detection)

- **6LoWPAN (IPv6 over Low Power Wireless Personal Area Networks):**
  IPv6 over Low Power Personal Area Network or 6LoWPAN is an IP-Also Based protocol that

ensures connectivity of even low data rate networks. It ensures that even the smallest or low power device should be a part of IoT. It helps provide end-to-end IP and is widely used in home automation systems.

- IP-based, ensures connectivity for low data rate networks. Allows smallest devices to be part of IoT. End-to-end IP. Low cost, short-range, low memory/bit rate.
- *Components:* Edge Router (bridge to internet/IPv6 networks, data exchange, local routing, subnet management), 6LoWPAN Devices (Hosts/End Devices, Routers), Access Point (AP - connects to broader network).



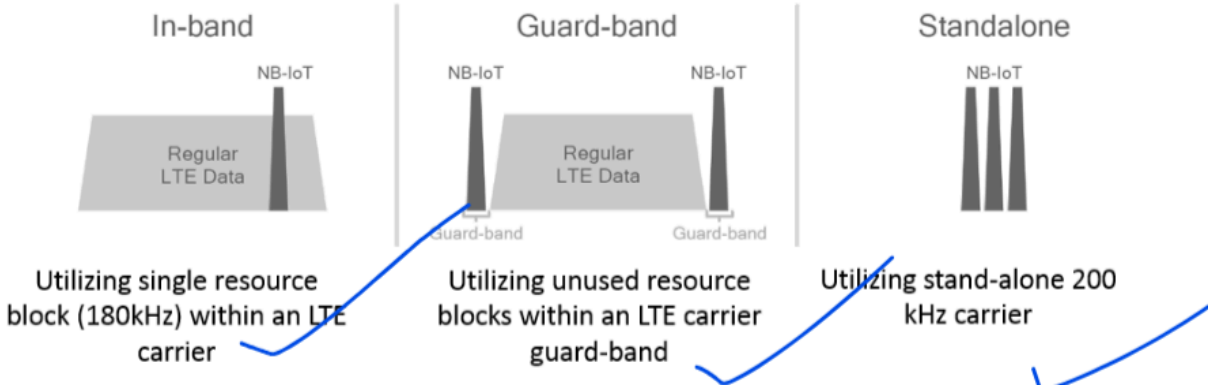
- *Features:* Uses IEEE 802.15.4 (2.4 GHz), ~200m range, 200kbps data, ~100 nodes.
- *Applications:* Smart homes, industrial automation, agriculture, smart cities.
- *Advantages:* Native IP communication, scalable, IPv6 security, reduced processing power on devices.

Basic Requirements of 6LoWPAN

- The device should be having sleep mode in order to support the battery saving.
- Minimal memory requirement.
- Routing overhead should be lowered.

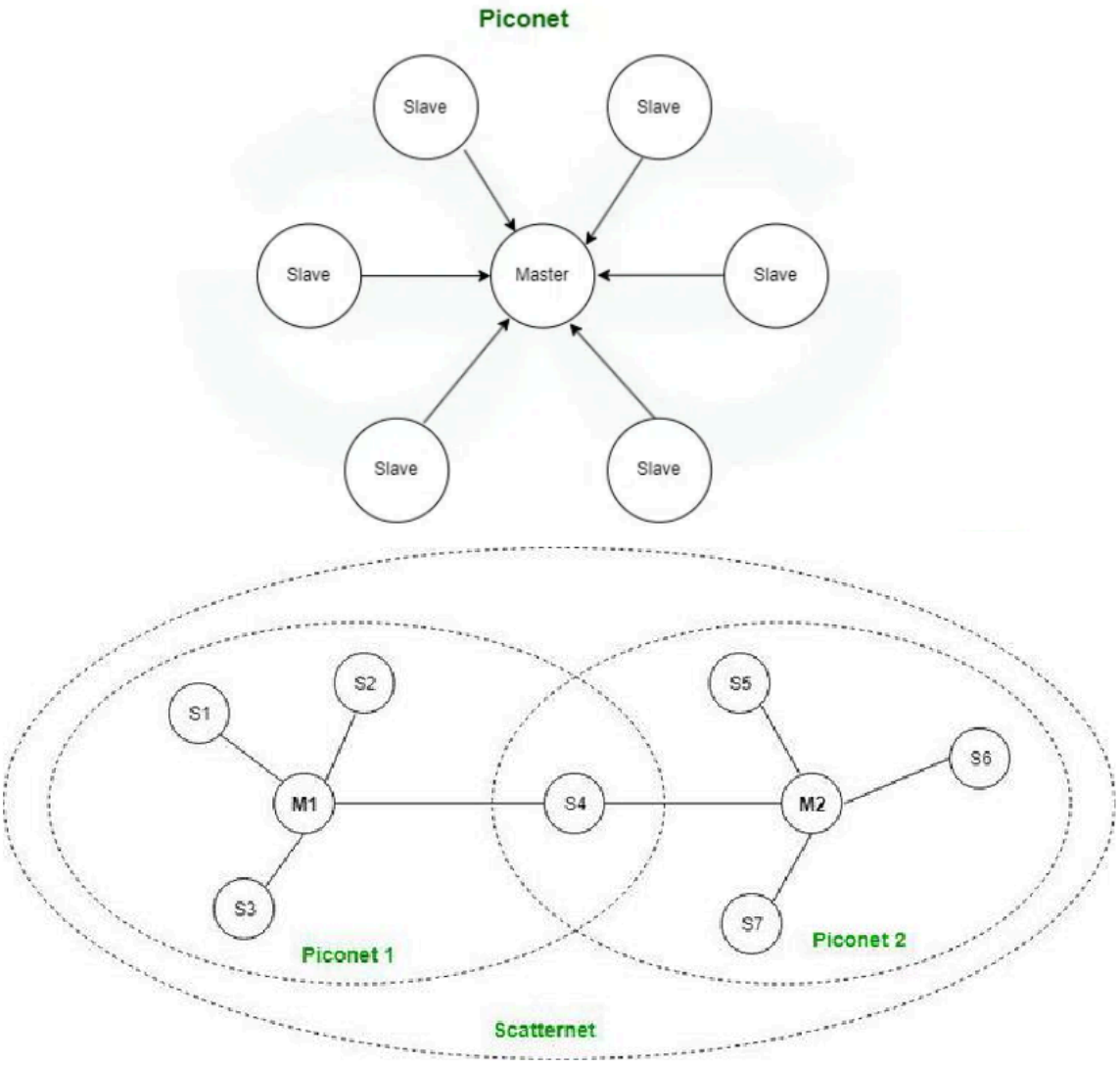
- **NB-IoT (Narrowband IoT):**
- The Internet of Things enables applications to connect and communicate with large numbers of wireless communication devices. It promises to power smart cities, utilities, manufacturing facilities, agricultural applications, remote industrial machinery and more. Any of these applications may use Narrowband Internet of Things (NB-IoT) network protocols.

  - LPWAN technology on cellular networks (limited frequency range). For IoT apps not needing high data/speed but long distance. Indoor/underground transmission, low battery.
  - *Deployment:* In-band (LTE carrier resource blocks), Guard band (unused LTE resource blocks), Standalone (GSM EDGE spectrum).



- *Applications:* Remote asset tracking, sustainable agriculture, smart vending, gas leak detection, smart metering, smart cities, healthcare (wearables), industries.

- **Bluetooth and BLE (Bluetooth Low Energy):**
  - Short-range WPAN. BLE optimized for low power. Used for fitness/medical wearables, smart home.
  - *Architecture:* Piconet (1 master, 7 active slaves, 255 parked slaves, 10m range). Scatternet (interconnected piconets via bridge nodes).



  - *Key Features:* 720kbps (classic), wireless, low-cost, robust, flexible.
  - *Advantages:* Low cost, easy-to-use, wall penetration, Ad-hoc connection, voice/data.

- *Disadvantages:* Hackable, slow data rate (3 Mbps classic), no routing.

Applications of Bluetooth

- It can be used in wireless headsets, wireless PANs, and LANs.
- It can connect a digital camera wireless to a mobile phone.
- It can transfer data in terms of videos, songs, photographs, or files from one cell phone to another cell phone or computer.
- It is used in the sectors of Medical healthcare, sports and fitness, Military.

Types of Bluetooth

Various types of Bluetooth are available in the market nowadays. Let us look at them.

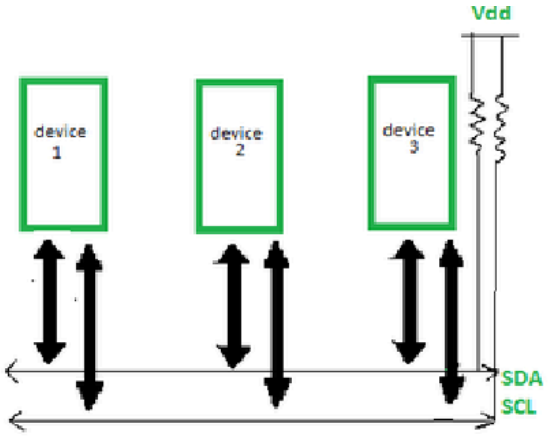
- **In-Car Headset:** One can make calls from the car speaker system without the use of mobile phones.
- **Stereo Headset:** To listen to music in car or in music players at home.
- **Webcam:** One can link the camera with the help of Bluetooth with their laptop or phone.

- **Wi-Fi (Wireless Fidelity):**
- Wireless Fidelity is one of the most hassle-free and fast wireless communication technology.It is the choice of many developers due to its various advantages. It allows access to theinternet as well as to connect devices in a specific range. Personal computers, smartphones,laptops, printers, and cars use this protocol.Wi-Fi relies on IEEE 802.11 standards, which define the specifications for wirelesscommunication. Devices communicate through a wireless access point (AP) or router, whichacts as a bridge to the wired network or the Internet. Wi-Fi utilizes various modulationtechniques, like OFDM (Orthogonal Frequency-Division Multiplexing), to efficiently transmitdata, while security protocols such as WPA3 (Wi-Fi Protected Access) ensure encryptedcommunication
  - Fast wireless communication (IEEE 802.11 standards). Uses AP/router. OFDM, WPA3 security.
  - *Pros:* High data rates (Gbps with Wi-Fi 6), widely available, strong encryption, security, high bandwidth.
  - *Cons:* High power consumption, limited range (50-100m indoors), congestion issues reduce performance.
  - *Applications:* Smart home, Industrial IoT (real-time monitoring), connected appliances.

- **Wired Communication Protocols:**
  - **External System Protocols:** USB, UART, Ethernet (communication between devices e.g. laptop & dev board).
  - **Internal System Protocols:** I2C, SPI (communication within the same circuit).
  - **I2C (Inter-Integrated Circuit bus):**
  - I2C stands for Inter-Integrated Circuit bus. It is an internal communication protocol that uses one wire SCL (serial clock) for clock and the other wire SDA (serial data) for transmission. It can connect many slave devices to master devices. Since communication is half-duplex, it can either

send or receive messages at a time. There are 3 types of I2C based on speed : Slow (under 100 Kbps) ,Fast (400 Kbps) ,High-speed (3.4 Mbps).

- Uses SCL (serial clock) and SDA (serial data). Half-duplex. Master/Slave modes. Data in 9-bit packets (Start, Address, Ack).



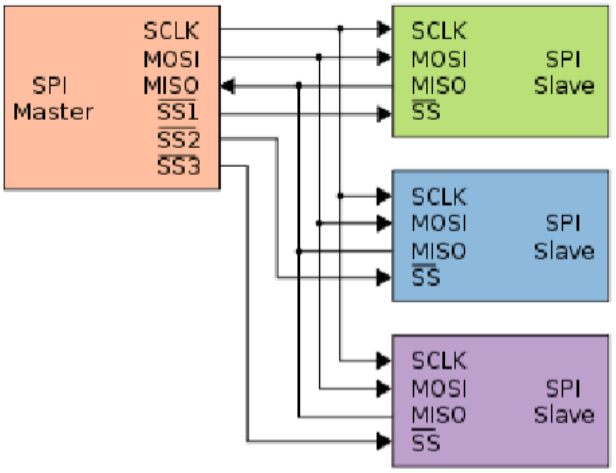
  According to I2C protocols, the data line can not change when the clock line is high, it can change only when the clock line is low. The 2 lines are open drain, hence a pull-up resistor is required so that the lines are high since the devices on the I2C bus are active low. The data is transmitted in the form of packets which comprises 9 bits. The sequence of these bits are –

  1. Start Condition: 1 bit
  2. Slave Address: 8 bit
  3. Acknowledge: 1 bit

- *Steps:* Start -> Address Slave -> ACK -> Data Tx -> Stop.
- *Advantages:* Multi-master/slave, 2 wires, adaptable, simple addressing, flow control.
- *Disadvantages:* Half-duplex, hardware complexity with many devices, address conflicts.
- *Applications:* DACs/ADCs, memory ICs, hardware sensors, multi-microcontroller comm.

- **SPI (Serial Peripheral Interface bus):**
- The Serial Peripheral Interface bus (SPI) is a synchronous serial communication interface specification used for short distance communication, primarily in embedded systems. The interface was developed by Motorola in the late 1980s. Typical applications include Secure Digital cards and liquid crystal displays.

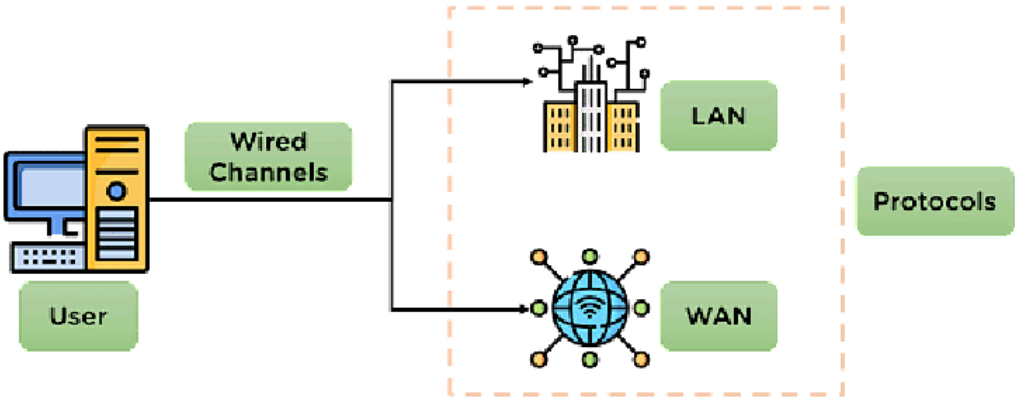
  - Synchronous serial, short distance, embedded systems. Full-duplex.
  - *Lines:* SCLK (Serial Clock), MOSI (Master Out Slave In), MISO (Master In Slave Out), SS (Slave Select).



- *Advantages:* No interruption transfer, simple hardware, full-duplex, no unique slave address needed, uses master clock, simple software, high speed, unidirectional signals.
- *Disadvantages:* Usually 1 master, no error check, more pins, short distance, no ACK.
- *Applications:* SD Card, EEPROM, Sensors (Temp/Pressure), ADC/DAC.
- **I2C vs. SPI (Tabular)**

Feature	I2C	SPI
Lines	2 (SDA, SCL)	3+ (MOSI, MISO, SCLK + SS per slave)
Speed	Up to 400kHz (High-speed 3.4Mbps)	Up to 10 MHz+
Complexity	Built-in addressing, good for multi-master	Simpler for point-to-point, can lack built-in addressing
Overhead	More for point-to-point	Less for point-to-point
Use Case	Occasional access onboard devices	Data streams
Acknowledgement	Yes	No

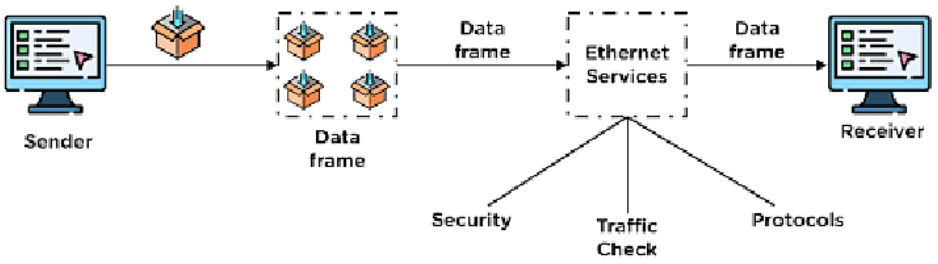
- **Ethernet:**
  - LAN protocol for data transfer, collision avoidance. MAC address (48-bit unique ID). Works at Physical & Data Link layers.



- *Transmission:* Data -> Packets -> Frame.

  Ethernet divides the transmission of data into two parts: packets and frames.

  - Packet–Refers to a unit of data in the network.
  - Frame–Refers to the collection of data packets being transmitted.



  - *Types:* Fast Ethernet (100 Mbps), Gigabit Ethernet (1000 Mbps), Switched Ethernet (uses switches/hubs, 1-10 Gbps).
  - *Pros:* Network starts/ends with it, needs switch, good for in-building.
  - *Cons:* Not for long distance (Fibre needed), many wires can be tough to manage.
  - *Applications:* Cloud Computing, CCTV, Copper/Fiber optic cable.

- **UART (Universal Asynchronous Receiver Transmitter):**
- UART stands for Universal Asynchronous Receiver Transmitter and USART also means for Universal Synchronous Asynchronous Receiver Transmitter. UART converts data into serial data. though, UARTs communicate directly by converting data into serial form and transmits it into the receiving UART that converts serial data into parallel data for the receiving device.
- The flow of data is from the Tx pin of the transmitting UART to the Rx pin of the receiving UART. Hence only two wires are required. UART is asynchronous and hence doesn't require a clock for synchronisation whereas USART uses a clock for synchronisation in case of synchronous communication. It can be used in asynchronous communication also. Hence, it is a dual-type of serial communication.
  - Serial communication, asynchronous (no clock for sync). Two wires (Tx, Rx). Converts parallel to serial and vice-versa.

  - *Pros:* No clock signal, 2 wires.
  - *Cons:* Data frame max 9 bits, no multi-master/slave.
  - *Application:* Serial ports, modems. Requires same bit speed, char length, parity, stop bits.

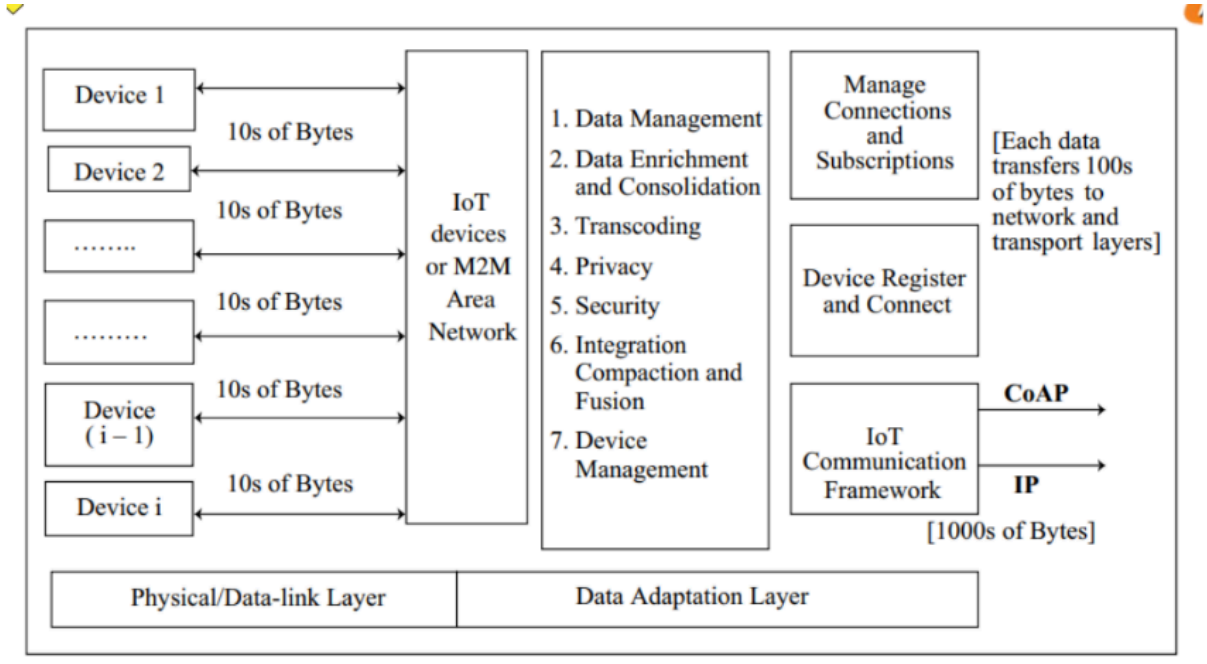
- **USB (Universal Serial Bus):**

- It is a representative peripheral interface.USB stands for Universal Serial Bus.It provides a serial bus standard for connecting devices,usually to a computer, but it also is in use on other devices such as set-top boxes, game consoles and PDAs
  - Peripheral interface, serial bus standard for connecting devices to computers/other devices (set-top boxes, PDAs).

  - *Advantages:* Low power (flash drives), robust data storage, simple, fast, widely accepted.
  - *Disadvantages (Flash Drives):* Limited write/erase cycles, easily lost.

24. Data Enrichment and Consolidation Gateway

- The practice of adding more information to raw data to make it more complete and thorough is known as data enrichment. It entails enhancing accuracy, adding pertinent features, and closing gaps to increase the data's analytical value. Through this process, simple knowledge is transformed into a rich resource that may provide greater understanding.
- Data Enrichment and Consolidation Gateway IoT or M2M gateway consisting of data enrichment and consolidation, device management and communication frameworks at the adaptation layer.



- **Functions:** Transcoding, Privacy/Security, Integration, Compaction/Fusion.
  - Data consolidation is the process of collecting and organizing data from various IoT devices into a single, unified system.

- **Steps in Data Consolidation**
  - **Data Collection:** Gather data from different IoT devices and sources.
  - **Data Storage:** Use cloud storage to hold large volumes of IoT data securely.
  - **Data Organization:** Structure data in a way that makes it easy to retrieve and analyze, such as organizing by timestamp, sensor type, or location.
  - **Data Compression:** Use compression techniques to reduce data volume while maintaining its integrity.

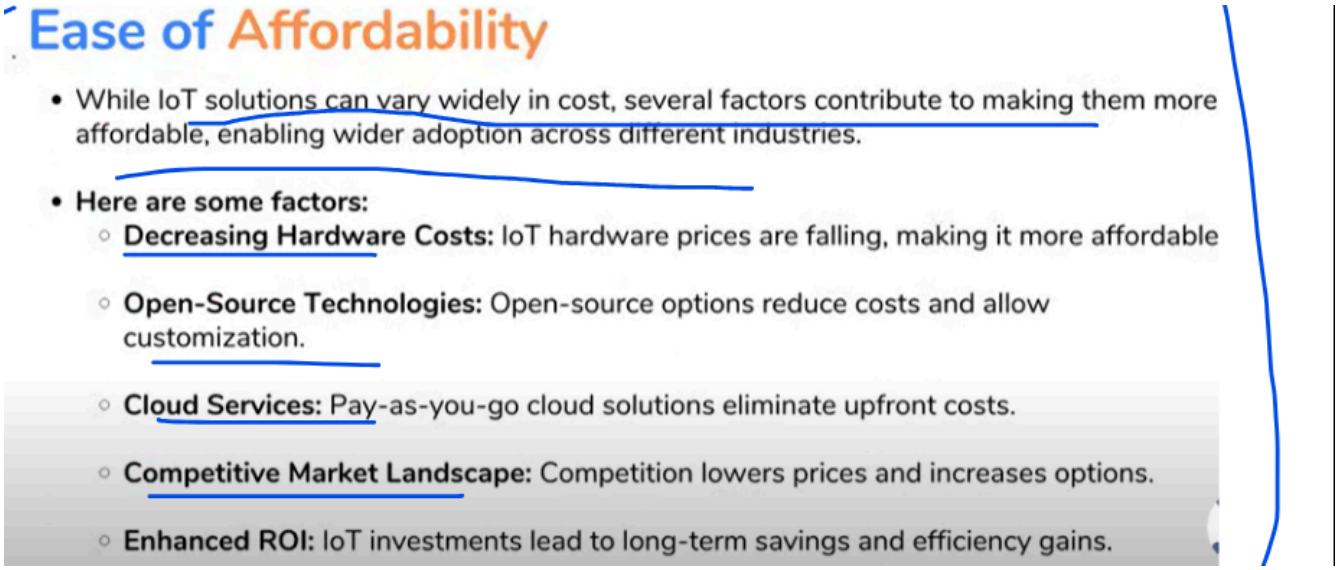
- **Transcoding:** Data adaptation, conversion, change of protocol/format/code. Renders web responses/device requests in acceptable formats.
- **Privacy:** Protecting sensitive data (medical, inventory) from unauthorized transfer. Ensure anonymity.
- **Aggregation:** Joining present/past data frames, removing redundancy.
- **Compaction:** Shortening information without changing meaning (e.g., incremental data).
- **Fusion:** Formatting information from multiple parts/sources, removing redundancy, presenting combined info.

25. **Device Management Gateway [PYQ for gateway importance]**

- **DM Functions:** Provisioning device ID/address, activating, configuring, registering/de-registering, attaching/detaching, subscription management, fault management.
- **OMA(Open Mobile Alliance)-DM:** Standard suggests DM server interacting with devices via gateway for IoT/M2M.
- **Gateway Functions for DM:**
  - Forwarding (if DM server & device interact directly).
  - Protocol Conversion (if different protocols used).
  - Proxy (if intermediate pre-fetch needed in lossy environments).
- **Ease of Designing and Affordability:**

Ease of Designing

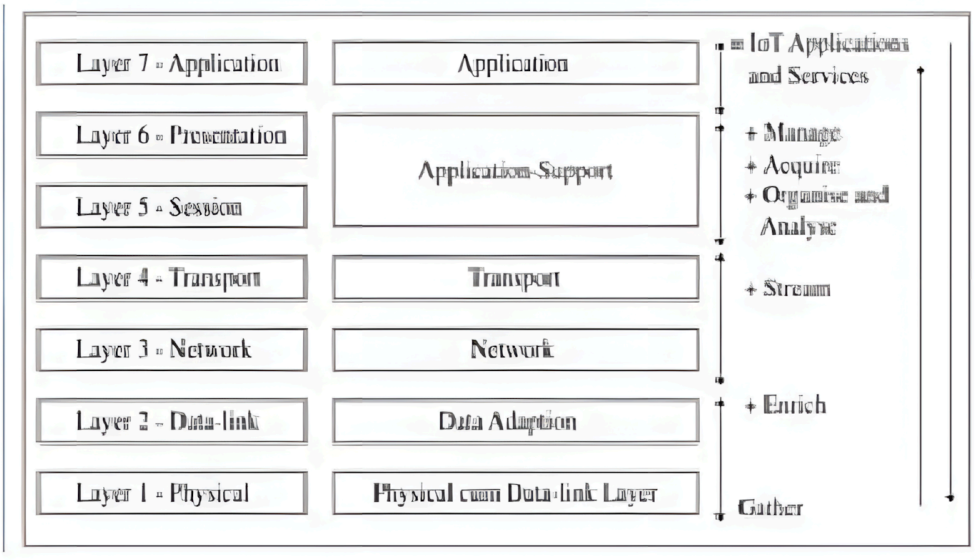
- The design process for IoT systems can be complex due to the integration of various hardware and software components.
- However, advancements in technology and frameworks have simplified this process significantly.
- **Here are some factors:**
  - **Modular Components:** Simplifies integration and customization.
  - **Development Platforms and Tools:** Streamlines development and deployment.
  - **Standard Protocols:** Ensures device interoperability.
  - **Libraries and Frameworks:** Accelerates prototyping.
  - **Community Support and Resources:** Aids troubleshooting and learning.



- Considers availability of SDKs, low-cost prototype boards, open-source components/protocols.
- Hardware should embed minimum components, use ready solutions for local networks and secure internet connectivity.
- Affordances like RFID/card with embedded microcontroller, memory, OS, NFC, RF module at low cost.

26. IoT M2M Systems Layers and Design Standardization

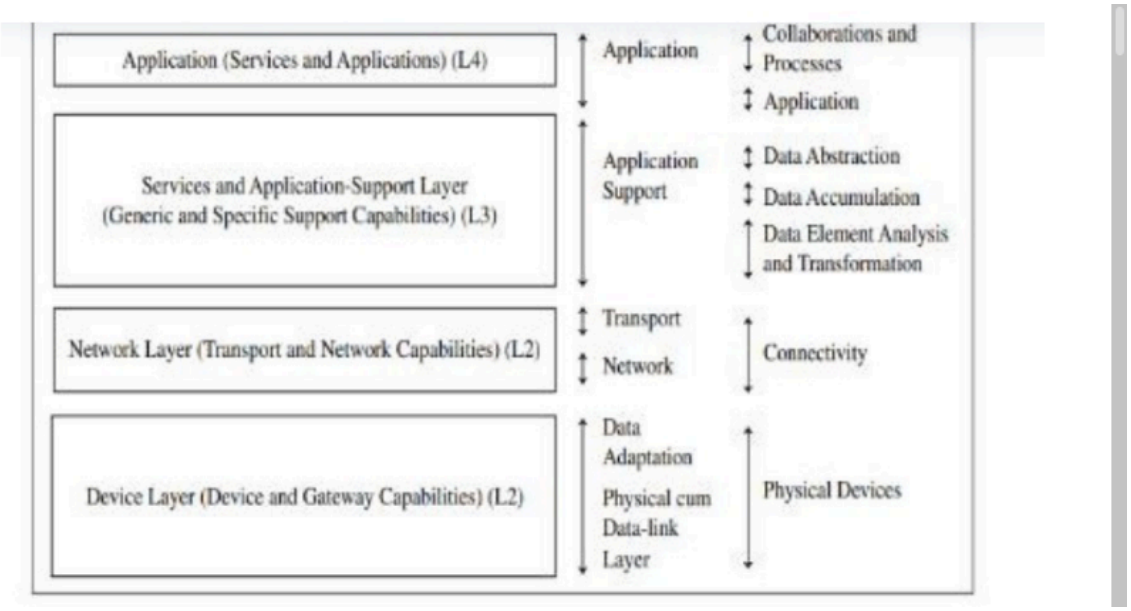
- **Standardization Bodies:**
  - **IETF (Internet Engineering Task Force):** Defines standard internet protocols (e.g., TCP/IP). Aims to make the internet work better.
  - **ETSI (European Telecommunications Standards Institute):** Develops global ICT standards (telecom, IoT, 5G, smart cities).
  - **OGC (Open Geospatial Consortium):** Develops standards for geospatial and location-based services (maps, satellite imagery).
- **IETF Six-Layer Modified OSI Model for IoT/M2M:** [PYQ]



- Classical 7-layer OSI model modified for IoT/M2M.

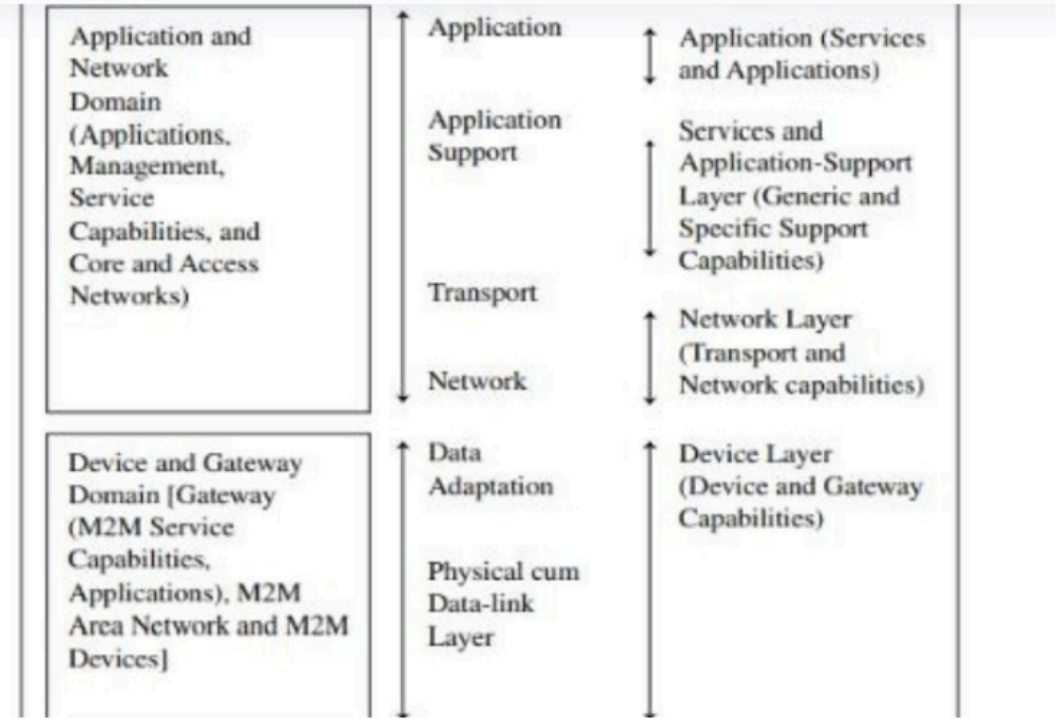
- *L1: Physical cum Data-Link Layer:* Senses data, transfers to L2. May integrate wireless transceiver.
- *L2: Data-Adaptation Layer:* Includes gateway for communication between device network and web. (e.g., group controller for streetlights using Bluetooth/ZigBee).
- *L3: Network Layer:* Communicates network stream on the Internet.
- *L4: Transport Layer:* Device identity management, registry, data routing.
- *L5: Application-Support Layer:* Data managing, acquiring, organizing, analyzing. Uses protocols like CoAP, UDP, IP.
- *L6: Application Layer:* New applications/services. Enables remote programming, commands.

- **ITU-T Reference Model (RM1):**



  - Recommends 4 layers:
    - L1: Device Layer (device/gateway capabilities).
    - L2: Network Layer (transport/network capabilities).
    - L3: Services and Application-Support Layer (generic/specific capabilities).
    - L4: Application Layer (applications/services).
  - *Comparison with 6-layer OSI:* RM1 Device Layer ~ OSI L1/L2. RM1 Network Layer ~ OSI L3/L4. RM1 Upper two layers ~ OSI L5/L6.

- **ETSI M2M Domains and High-level Capabilities:**



  - Proposes high-level architecture for M2M applications/service capabilities.
  - **Network Domain Capabilities (6):** M2M applications, M2M service capabilities, M2M management functions, Network management functions, CoRE network, Access network.
  - **Device and Gateway Domain Functional Units:** Gateway (between M2M area network & CoRE/access network), M2M area network (Bluetooth, ZigBee etc.), M2M devices.

27. What is a Smart Sensor vs. Sensor Node? [PYQ]

- **Sensor Node:** A fundamental component of a Wireless Sensor Network (WSN). It typically consists of:
  - One or more sensors to detect physical phenomena.
  - A microcontroller for processing.
  - A transceiver for wireless communication.
  - A power source (usually a battery).
  - Its primary role is to collect data and transmit it, often with minimal local processing.
- **Smart Sensor:** A sensor that includes integrated electronics for:
  - Signal conditioning (amplification, filtering).
  - Data conversion (analog-to-digital).
  - Some level of data processing or decision-making capabilities (e.g., calibration, self-diagnosis, simple algorithms).
  - Communication interface (wired or wireless).
- **Difference:**

- A sensor node *contains* sensors; it's a networked entity. A smart sensor *is* a sensor with enhanced capabilities.
- All smart sensors can be part of a sensor node. Not all sensors in a sensor node are necessarily "smart" (though modern nodes often use smart sensors).
- Smart sensors offer more processed/refined data, reducing the load on the microcontroller of the sensor node or the central gateway/server.
- A sensor node is more about the networking aspect; a smart sensor is about the capability of the sensing element itself.
- In an IoT conceptual framework, a "sensor connected to a gateway, functions as a smart sensor (with computing and communication capacity)". This implies the gateway adds the "smart" processing if the sensor itself isn't inherently smart.
