Emerging Trends In Computer & Information Technology
22618

ESE – 70M *# (90 Min)

PA - 30

Chap 2: Internet of Things(IoT)

Marks: 18

What is IoT?

- The **Internet of Things (IoT)** is the network of physical objects—devices, vehicles, buildings and other items embedded with electronics, software, sensors, and network connectivity—that enables these objects to collect and exchange data.

Allows object to be sensed and controlled remotely.

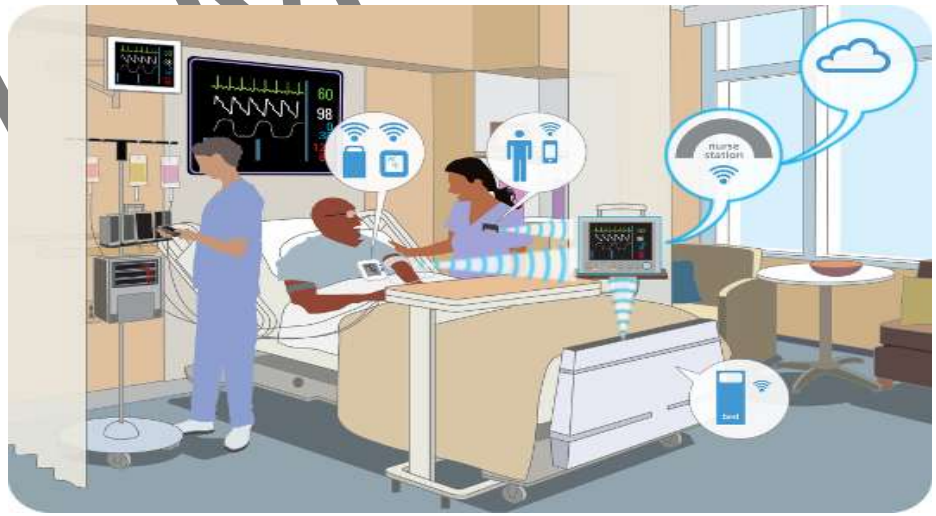
Creates more opportunities between the **physical world** and **computer based system.**

Where is IoT?

It’s everywhere!

Wearable Tech

Smart Appliances

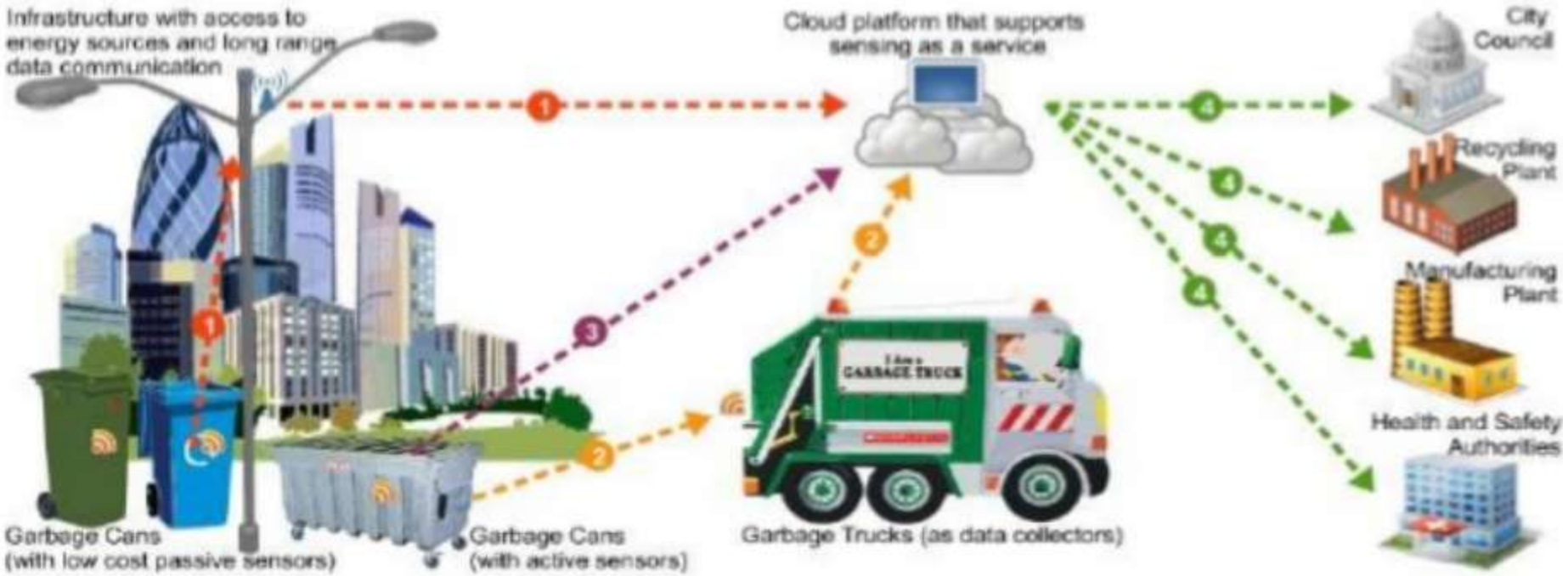


Healthcare

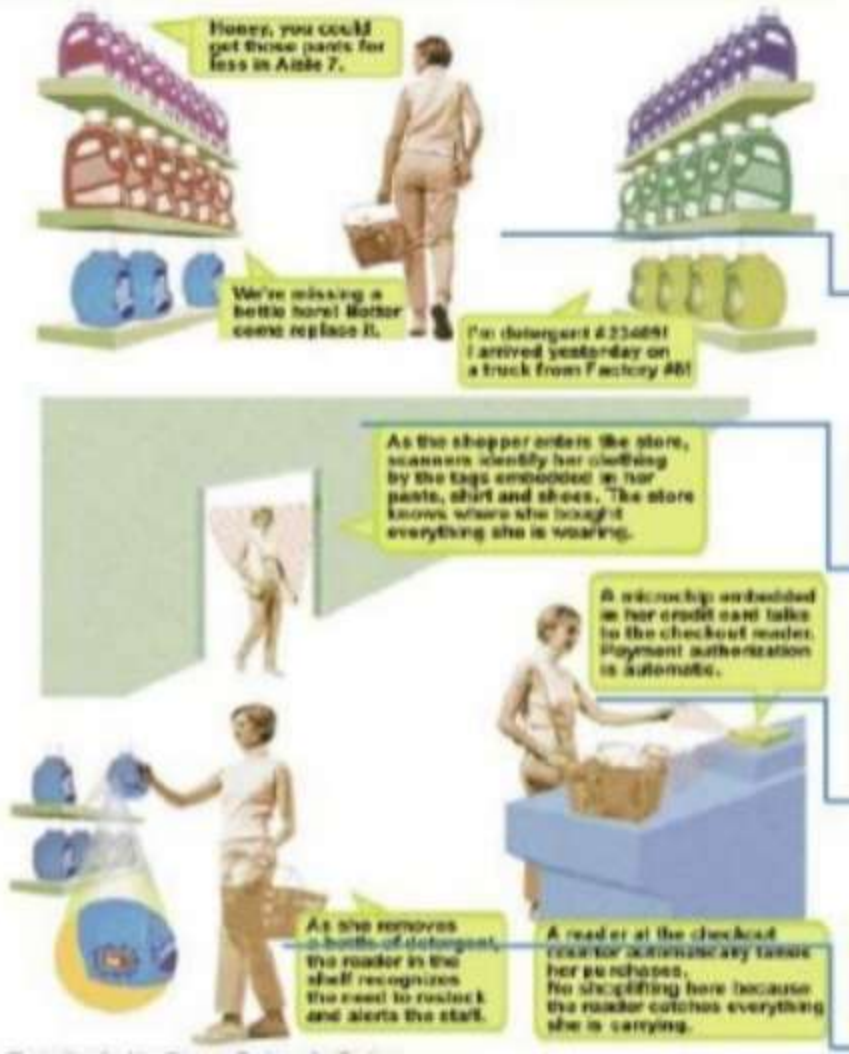
Smart Parking

Residents can identify and reserve the closest available space, traffic wardens can identify non-compliant usage, and municipalities can introduce demand-based pricing.

Efficient Waste Management in Smart Cities Supported by the Sensing-as-a-Service



In the world of IoT, even the cows will be connected and monitored. Sensors are implanted in the ears of cattle. This allows farmers to monitor cows' health and track their movements, ensuring a healthier, more plentiful supply of milk for people to consume. On average, each cow generates about 200 MB of information per year.



(2) When shopping in the market, the goods will introduce themselves.

(1) When entering the doors, scanners will identify the tags on her clothing.

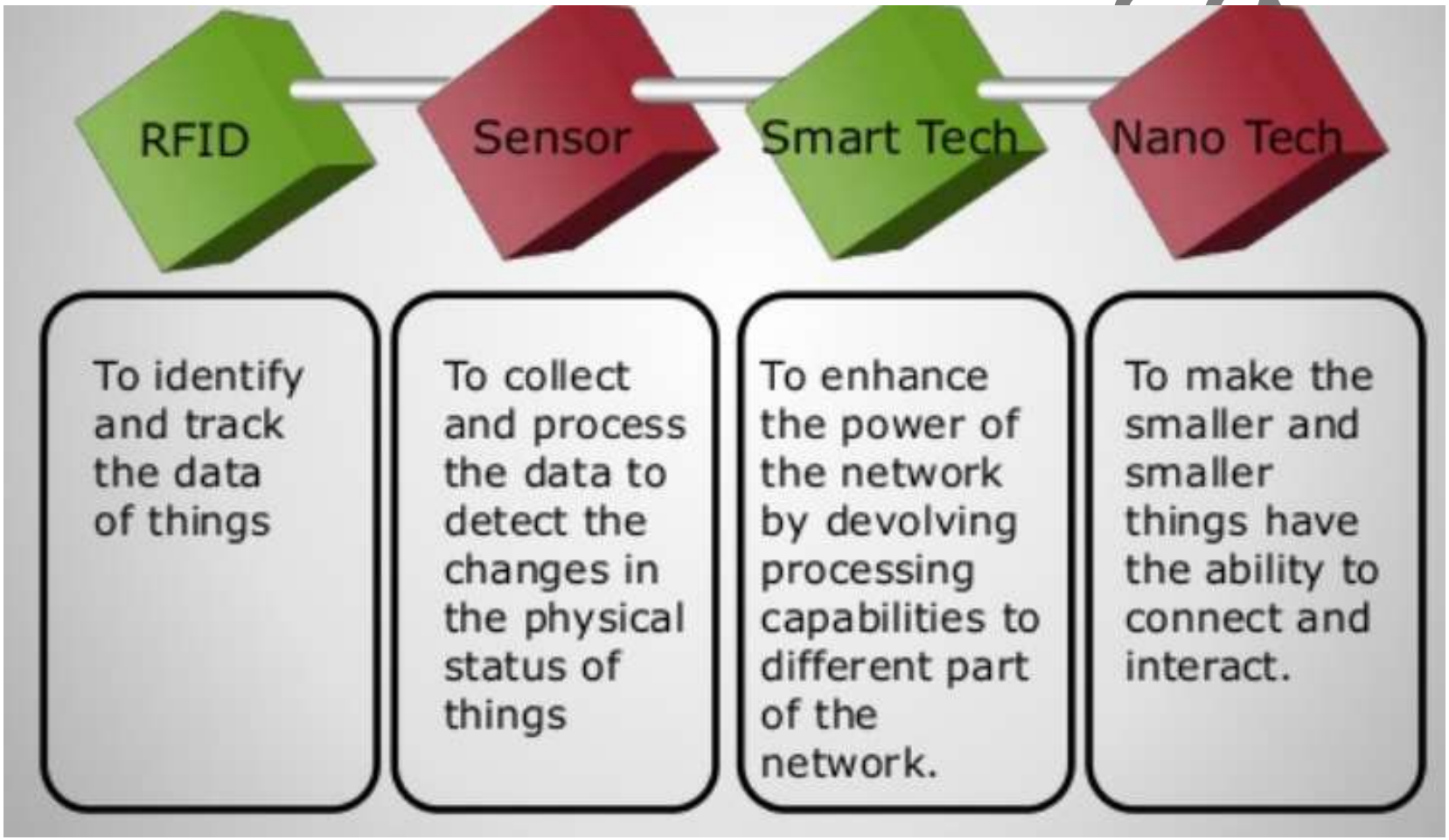
(4) When paying for the goods, the microchip of the credit card will communicate with checkout reader.

(3) When moving the goods, the reader will tell the staff to put a new one.


HOW MANY STEPS
HAVE YOU
WALKED TODAY?

- Concept of IoT became popular in 1999 through the AutoID center at MIT(US).
- RFID was seen as the prerequisites for the IoT at that point.
- If all objects and people in daily life were equipped with identifiers computer could manage and inventory them.
- Besides RFID, the tagging of things may be achieved through such technologies as near field communication such as barcodes, QRCodes and bluetooth etc

How IoT Works



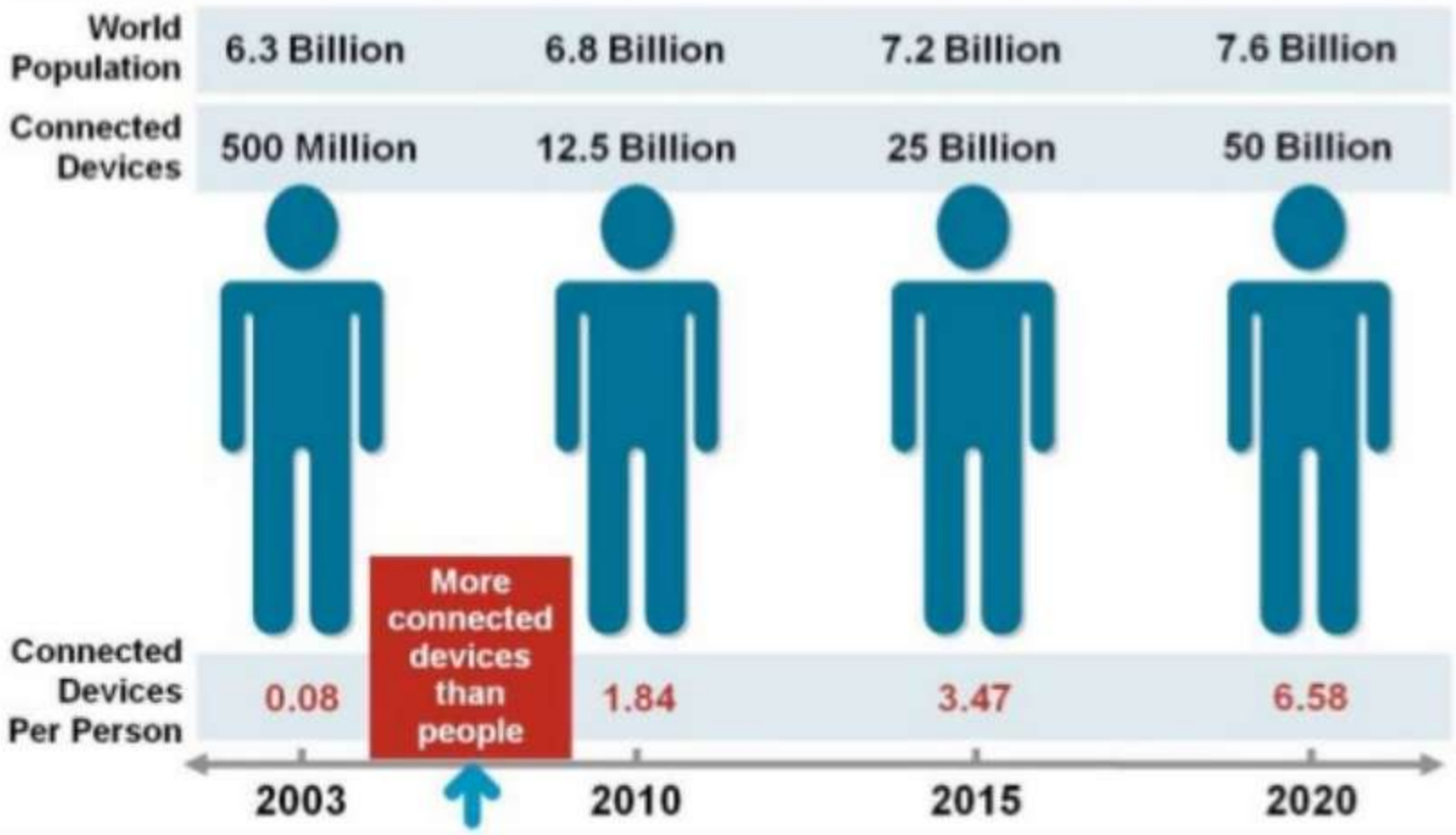
- Tagging Things
- Feeling Things
- Shrinking Things
- Thinking Things

Tagging Things : Real-time item traceability and addressability by ***RFID***s.

Feeling Things : ***Sensors*** act as primary devices to collect data from the environment.

Shrinking Things : *M*iniaturization and ***Nanotechnology*** has provoked the ability of smaller things to interact and connect within the "things" or "smart devices."

Thinking Things : ***Embedded intelligence*** in devices through sensors has formed the network connection to the Internet. It can make the "things" realizing the intelligent control.



"Change is the only thing permanent in this world"

Various Names

- M2M (Machine to Machine)
- “Internet of Everything” (Cisco Systems)
- “World Size Web” (Bruce Schneier)
- “Skynet” (Terminator movie)

2.1.Embedded System

- An embedded system is one that has computer hardware with software embedded in it as one of its important components.

Its s/w embeds in ROM
Does not need secondary memory as in a computer
Designed to perform a few dedicated functions

- A convenient definition for an embedded system is :
  - **An embedded system is any computer system contained within a product that is not described as a computer.**
- Typical characteristics of an embedded system are:
  - **Single-functioned**: designed for a specific task
  - **Reactive and Real time:** gives the o/p within the time limits(pager always functions as pager)
  - **Microprocessors based** (must be microprocessor or microcontroller based)
  - **Memory** (must have memory as its software usually embeds in ROM ,doesn't need nay secondary memories in the Computer)
  - **Connected** (must have connected peripherals to connect input and output device)
  - **The software for embedded systems is normally referred to as firmware.**

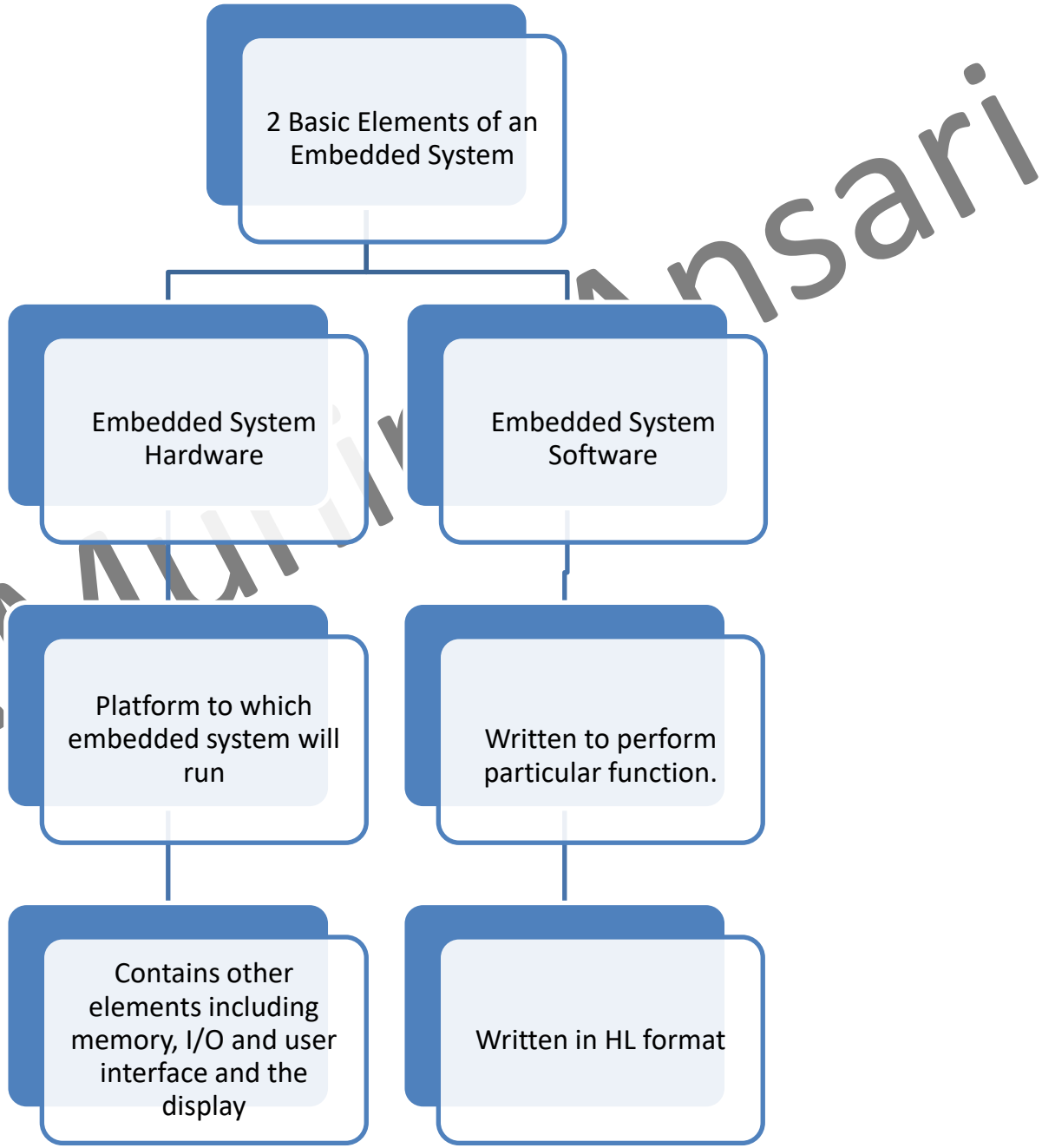
Types of embedded systems

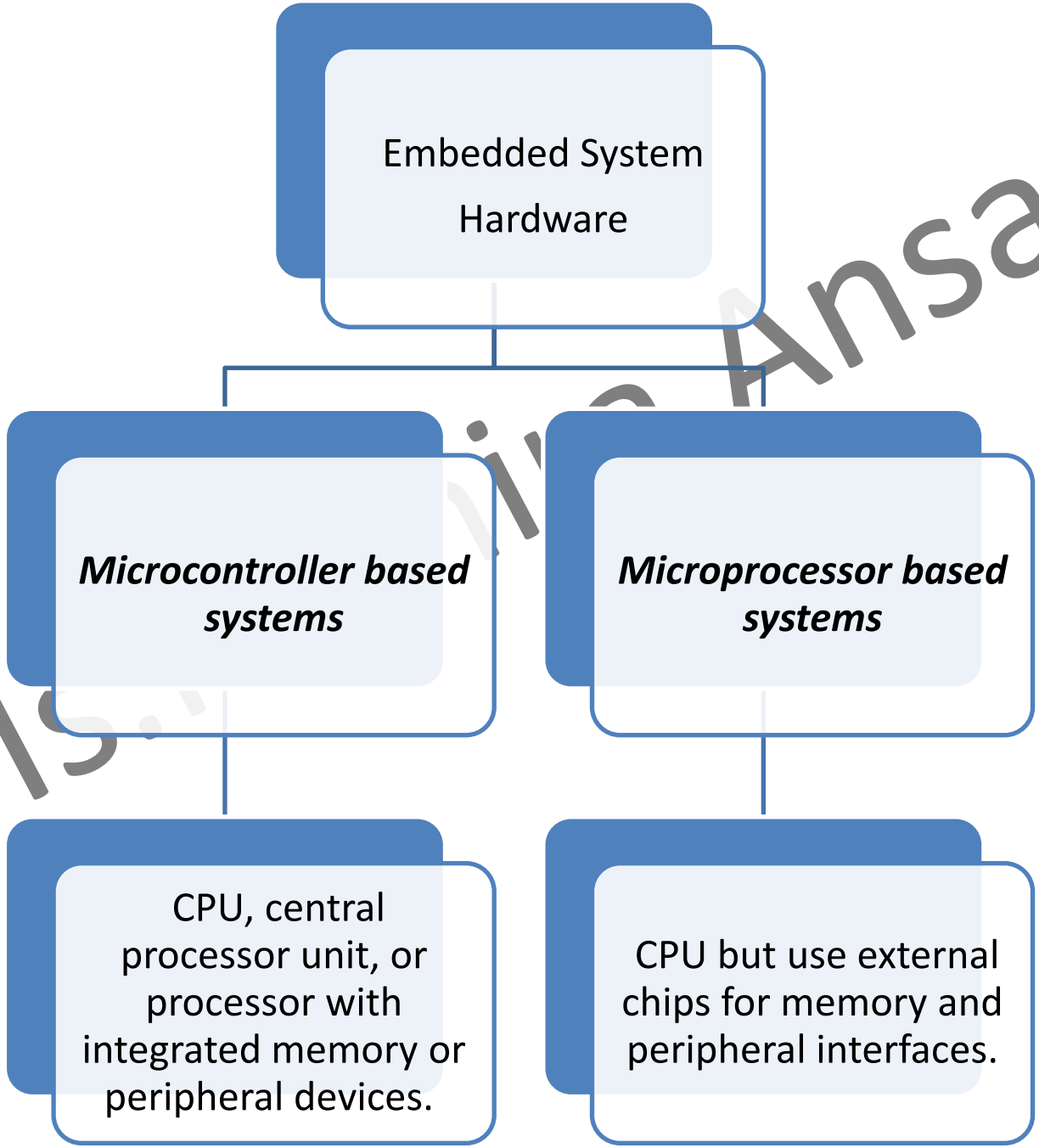
- **Mobile embedded systems** are small-sized systems that are designed to be portable. **Digital cameras are an example of this.**

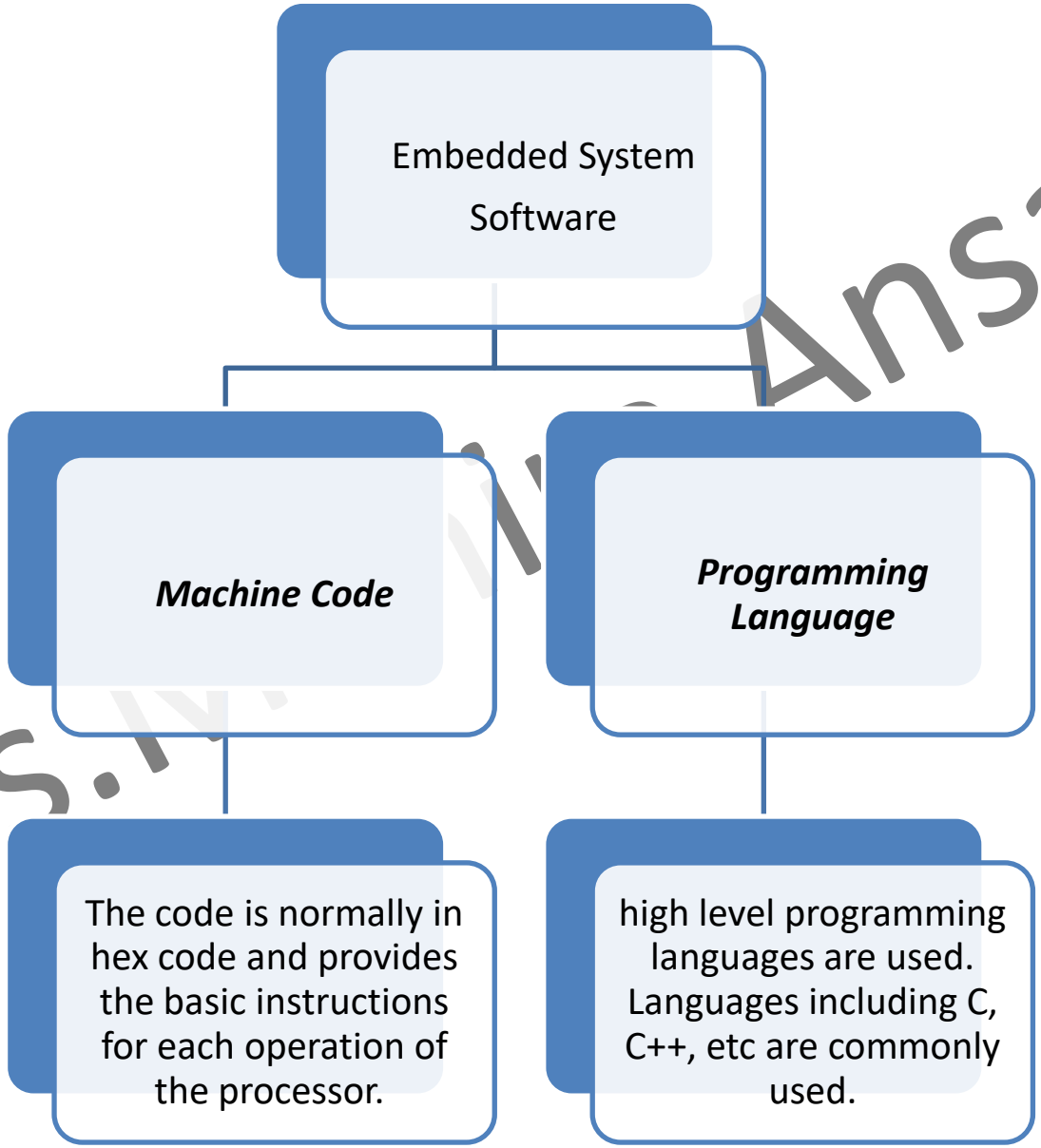
- **Networked embedded systems** are connected to a network to provide output to other systems. Examples include **home security systems and point of sale (POS) systems.**

- **Standalone embedded systems** are not reliant on a host system. Like any embedded system, they perform a specialized task. However, they do not necessarily belong to a host system, unlike other embedded systems. **A calculator or MP3 player is an example of this**.

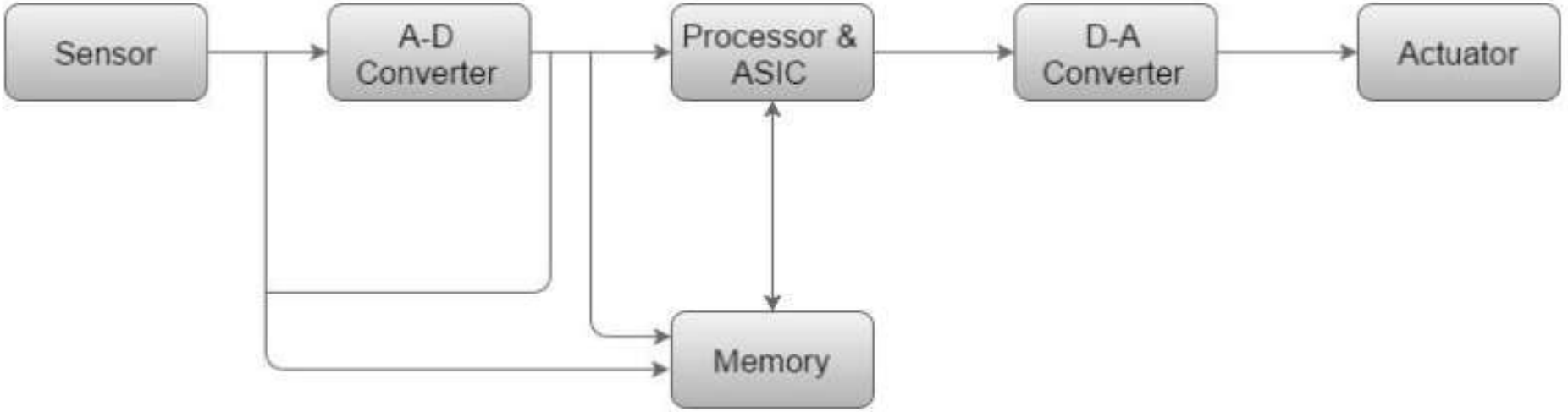
- **Real-time embedded systems** give the required output in a defined time interval. They are often used in medical, industrial and military sectors because they are responsible for time-critical tasks. **A traffic control system is an example of this.**


2 Basic Elements of an Embedded System
Embedded System Hardware
Embedded System Software
Platform to which embedded system will run
Written to perform particular function.
Contains other elements including memory, I/O and user interface and the display
Written in HL format


Embedded System Hardware
Microcontroller based systems
Microprocessor based systems
CPU, central processor unit, or processor with integrated memory or peripheral devices.
CPU but use external chips for memory and peripheral interfaces.


Embedded System Software
Machine Code
Programming Language
The code is normally in hex code and provides the basic instructions for each operation of the processor.
high level programming languages are used. Languages including C, C++, etc are commonly used.

Basic Structure of an Embedded System



- **Sensor** – It measures the physical quantity and converts it to an electrical signal which can be read by an observer or by any electronic instrument like an A2D converter. A sensor stores the measured quantity to the memory.
- **A-D Converter** – An analog-to-digital converter converts the analog signal sent by the sensor into a digital signal.
- **Processor & ASICs** – Processors process the data to measure the output and store it to the memory.
- **D-A Converter** – A digital-to-analog converter converts the digital data fed by the processor to analog data
- **Actuator** – An actuator compares the output given by the D-A Converter to the actual (expected) output stored in it and stores the approved output.

Purpose of Embedded System

- Each Embedded system is designed to serve the purpose of any one or a combination of the following tasks.
  - 1. Data collection/Storage/Representation
  - 2. Data communication
  - 3. Data (Signal) processing
  - 4. Monitoring
  - 5. Control
  - 6. Application specific user interface

Data collection/Storage/Representation

- usually done for **storage, analysis, manipulation and transmission** of data where data can be either analog (continues) or Digital (discrete).

collect data directly in the form of analog and converts the analog to digital signal by using A/D converters and then collect the binary equivalent of the analog data.

can be directly captured without any additional interface by digital embedded system.

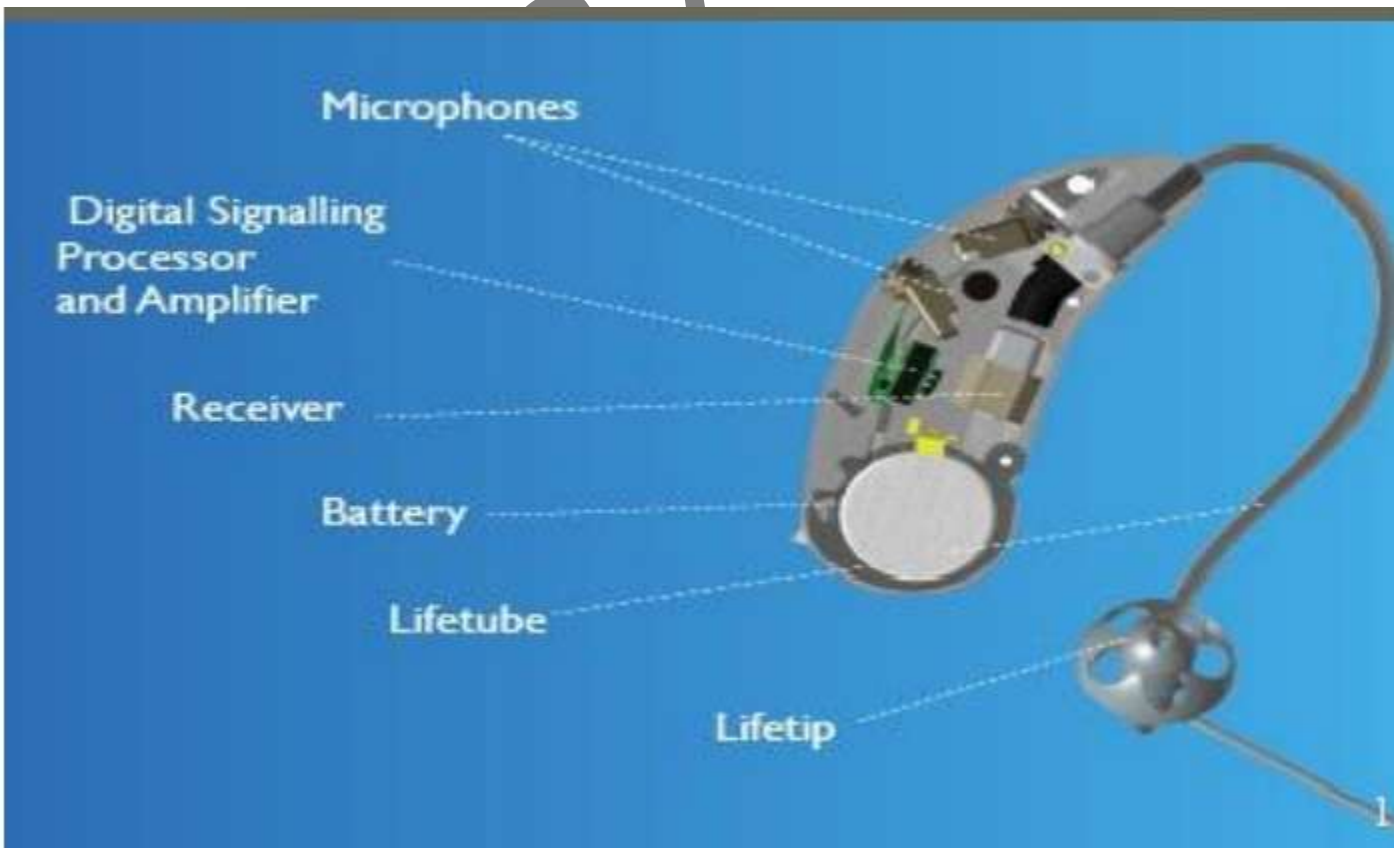
collected data may be stored directly in the system or may be transmitted to other systems or it may be processed by the system or it may be deleted instantly after giving a meaningful representation

Data Communication

- Embedded data communication systems are developed in applications ranging from **complex satellite communication systems to simple home networking systems**

Data (Signal) Processing

- The data collected by embedded system may be used for various kinds of signal processing.
- A **digital hearing aid** is a typical example of an embedded system employing data processing.



Monitoring

- All embedded products coming under the medical domain are with **monitoring functions only**. They are used for determining the state of some variables using input sensors.
- A very good example is the electro cardiogram (ECG) machine for monitoring the heartbeat of patient.

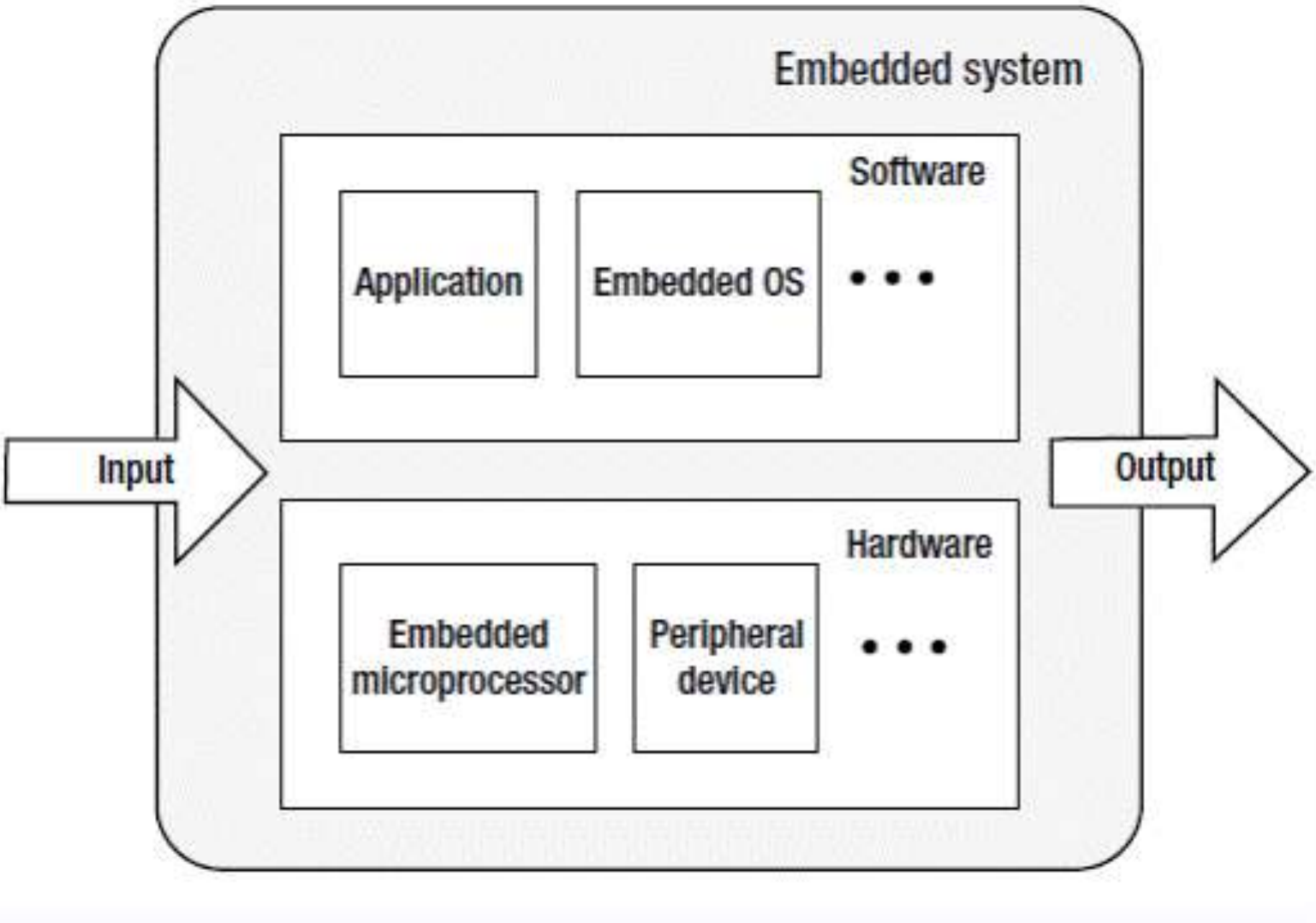
Control

- Embedded system with control functionalities impose **control over some variables according to the input variables**.
- A system with control functionality contains **both sensors and actuators**.
- **Sensors** are **inputs ports** for capturing the changes in environment variables or measuring variable.
- Actuators are output ports are controlled according to the changes in input variable.

Application specific user interface

- These are embedded systems with **buttons, switches, keypad, lights, bells, display units**, etc..
- **Example :** Mobile Phones

Architecture of embedded system



- Typical embedded system consisting of two main parts:
  - embedded hardware(processor, memory, bus, peripheral devices, I/O ports, and various controllers)
  - embedded software(embedded operating system and various applications)

- In embedded systems, the microprocessor's role and function are usually the same as those of the CPU in a general-purpose computer: **control computer operation, execute instructions, and process data**.
- In many cases, the microprocessor in an embedded system is also called the CPU. Memory is used to store instructions and data. I/O modules are responsible for the data exchange between the processor, memory, and external devices. External devices include secondary storage devices (such as flash and hard disk), communications equipment, and terminal equipment.
- The system bus provides data and controls signal communication and transmission for the processor, memory, and I/O modules.

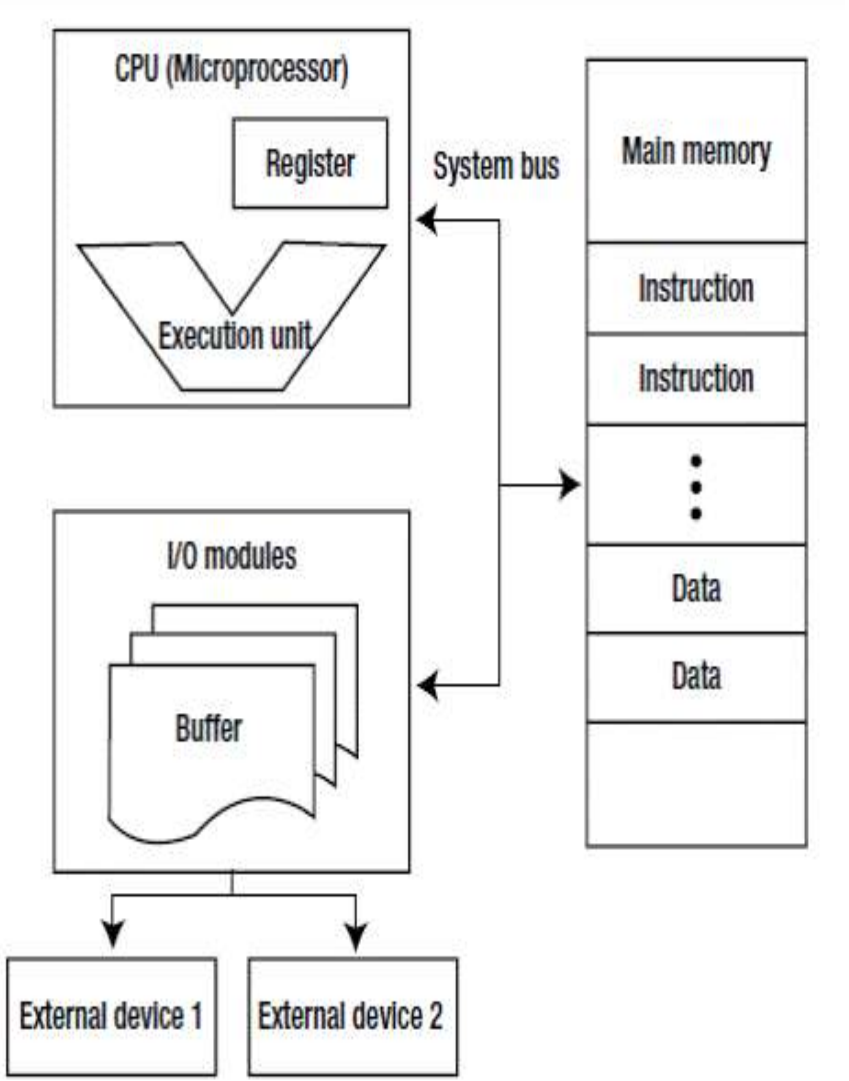
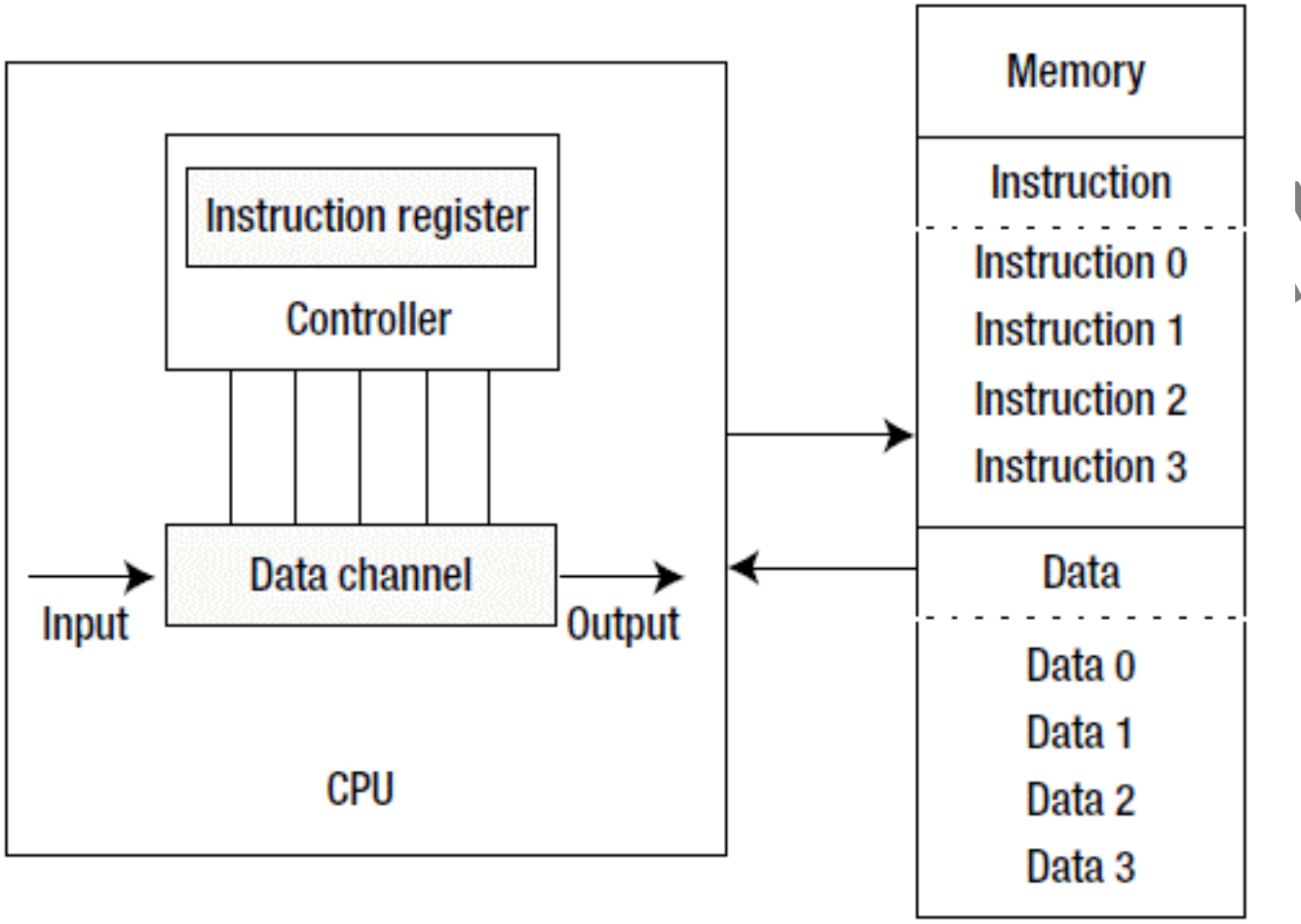


Fig: Computer Architecture

- There are two types of architecture:
  - **Von Neumann Architecture**
  - ***Harvard architecture***
- In general we can say that when data and code lie in the same memory block, then the architecture is referred as **Von Neumann architecture**.
- In case data and code lie in different memory blocks, then the architecture is referred as **Harvard architecture.**

Von Neumann Architecture

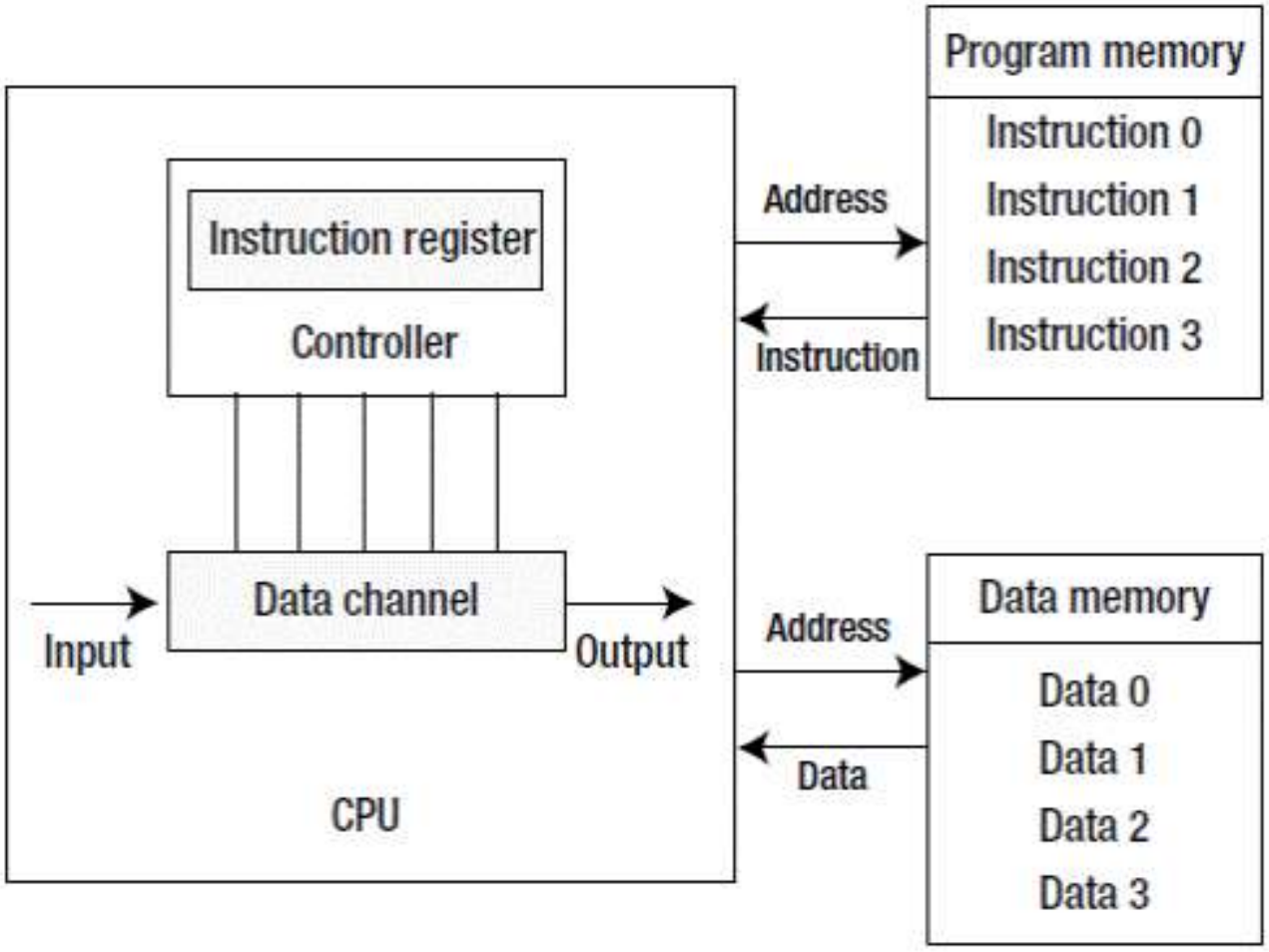
- The Von Neumann architecture was first proposed by a computer scientist John von Neumann.
- **One data path or bus exists for both instruction and data that is the software and data use the same memory**.
- As a result, **the CPU does one operation at a time**. It **either fetches an instruction from memory, or performs read/write operation on data**. So an instruction fetch and a data operation cannot occur simultaneously, sharing a common bus.



The transmission of information becomes the bottleneck of computer performance and affects the speed of data processing; so, it is often called the *Von Neumann bottleneck*.

Harvard Architecture

- The Harvard architecture offers **separate storage and signal buses for instructions and data**.
- This architecture has **data storage entirely contained within the CPU**, and there is no access to the instruction storage as data.
- Computers have **separate memory areas for program instructions and data using internal data buses, allowing simultaneous access to both instructions and data**.



Because the Harvard architecture has **separate program memory and data memory, it can provide greater data-memory bandwidth**, making it the ideal choice for digital signal processing. Most systems designed for digital signal processing (DSP) adopt the Harvard architecture. The Von Neumann architecture features simple hardware design and flexible program and data storage and is usually the one chosen for general-purpose and most embedded systems.

Embedded Processors

- Processor is the heart of an embedded system. It is the basic unit that takes inputs and produces an output after processing the data.
- A processor has two essential units –
  - Program Flow Control Unit (CU)
  - Execution Unit (EU)
- The **CU** includes a **fetch unit for fetching instructions from the memory**. The **EU** has circuits that **implement the instructions pertaining to data transfer operation and data conversion from one form to another**.
- The **EU includes the Arithmetic and Logical Unit (ALU)** and also the **circuits that execute instructions for a program control task such as interrupt, or jump to another set of instructions**.

- **Special microprocessor & microcontrollers often called Embedded processor**.
- An embedded processor is **used when fast processing, fast switching and automatic ALU operations are needed**.
- Where
  - Microprocessor: is a single chip semiconductor device, contains an ALU, program counter, register, timing and interrupt circuit on a single chip. To make micro computer one must add usually RAM ROM ,memory decoder ,number of parallel and serial ports etc.
  - Microprocessors are multitasking in nature. Can perform multiple tasks at a time. For example, on computer we can play music while writing text in text editor.

– Microcontroller: is a functional computer system on a chip. It contains a processor, memory and programmable I/O peripherals.
  - it include an integrated CPU ,memory and peripherals capable of input output.
  - Single task oriented. For example, a washing machine is designed for washing clothes only.

Peripheral Interface Controller (PIC)

- Peripheral Interface Controller (PIC) provide d by Micro-chip Technology in 1993.
- Initially this was developed for **supporting PDP computers to control its peripheral devices, and therefore, named as a peripheral interface device**.
- Based on Harvard architecture.
- Used in many electronic devices such as phones, alarm system , embedded system etc.

- Features:
  - On chip program ROM
  - On chip RAM,EEPROM
  - Include Timers
  - Include ADC
  - Contains IO ports
  - Provides interrupts

- **Application:**
  - Home Appliances
  - Motor Control
  - High Temperature
  - Audio and Speech
  - Touch Sensing Solution
  - Wireless connectivity
  - USB

Advanced Virtual RISC(AVR)

- Four times faster than PIC and consumes less power.
- Executes most of the instruction in single cycle.
- Can operated in different power saving mode.

- Features:
  - Internal 256KB self programmable flash memory is provided.
  - Internal EEPROM upto 4KB and SRAM upto 16KB
  - External 64KB (in some model of AVR)
  - 8 bit and 16 bit timers
  - A/D converter
  - D/A converter
  - Multiple power saving sleep mode
  - Low operating voltage device

- Application:
  - Displays on LCD
  - Interface any type of sensors
  - Interface GSM and GPS
  - Automation of heavy machinery
  - Fire detection and safety device
  - Light sensing Temperature sensing and controlling device

Advanced RISC Machine(ARM)

- 32 bit microcontroller
- Introduced in 1987 by Acron Computers.
- Harvard Architecture
- Supports low level and high level programming languages.

- Features:
  - 32bit instructions can be easily mixed with 16 bit instruction in program
  - Supports multi processing
  - Enhanced power saving design
  - 64 and 32 execution states for scalable high performance
  - Suports DSP algo’s

- Application
  - Smartphones
  - Multimedia Players
  - Digital Cameras
  - Tablet
  - Robotics
  - Digi TV
  - Smart Watches etc.

Application Specific IC(ASIC)

- Designed for particular kind of transmission protocol or a hand held device.
- Can contrast with RAM and microprocessor chips in your PC
- Manufactured for special application

	8051	PIC	AVR	ARM
Bus width	8-bit for standard core	8/16/32-bit	8/32-bit	32-bit mostly also available in 64-bit
Speed	12 Clock/instruction cycle	4 Clock/instruction cycle	1 clock/ instruction cycle	1 clock/ instruction cycle
Memory	ROM, SRAM, FLASH	SRAM, FLASH	Flash, SRAM, EEPROM	Flash, SDRAM, EEPROM
Memory Architecture	Von Neumann architecture	Harvard architecture	Modified	Modified Harvard architecture
Power Consumption	Average	Low	Low	Low
Cost (as compared to features provide)	Very Low	Average	Average	Low

IoT Definition & Characteristics of IoT

- IoT is a computing concept that describes the idea of every day physical objects being connected to the internet and being able to identify themselves to other device.

refers to physical and virtual objects that have unique identities and are connected to internet to facilitate intelligent application

Characteristics:

- Dynamic and self-adapting
- Self-configuring
- Interoperable Communication protocols
- Unique identity
- Integrated into information network

– **Dynamic and Self Adapting:**

Capability to dynamically adapt with the changing contexts and take actions based on their operating conditions or sensed environment.

Eg: Surveillance Camera (adapt their mode automatically based on day and night)

– **Self Configuring**

Allows large number of devices to work together to provide certain functionality, setup the network i.e a new device can be easily added to the existing network.

Eg: Whether Monitoring, Whenever there will be free wifi access one device can be connected easily.

IoT devices can be able to upgrade the software with minimal intervention of user, whenever they are connected to the internet

– **Interoperable Communication Protocol**

Allows communication to other devices and also with the infrastructure.

allows different devices (different in architecture) to communicate with each other as well as with different network.

For ex: MI Phone is able to control the smart AC and smart TV of different manufacturer.

– **Unique Identity**

Unique identity or unique identifier such as IP address or URI.

IoT devices have intelligent interfaces which allow communicating with users. It adapts to the environmental contexts.

It also allows the user to query the devices, monitor their status, and control them remotely, in association with the control, configuration and management infrastructure.

– **Integrated into information network**

Allows IoT device to communicate and exchange data with other devices and systems, share some information with other connected devices.

The devices can be discovered dynamically in the network by other devices.

For ex. If a device has wifi connectivity then that will be shown to other nearby devices having wifi connectivity.

The devices ssid will be visible though out the network. Due to these things the network is also called as information network.

The IoT devices become smarter due to the collective intelligence of the individual devices in collaboration with the information network.

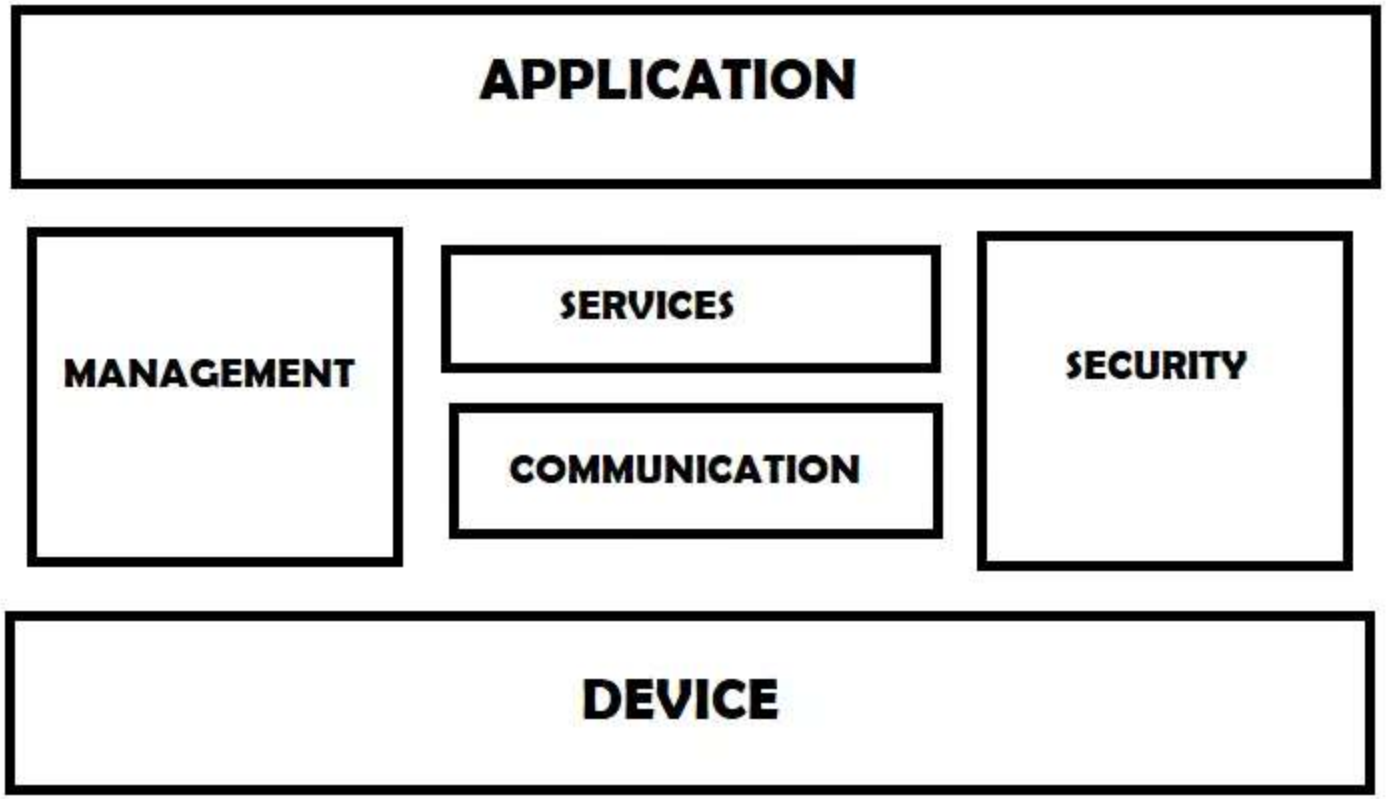
For Ex: weather monitoring system. Here the information collected from different monitoring nodes (sensors, arduino devices) can be aggregated and analysed to predict the weather.

Logical Design of IoT

- Abstract representation of the entities and processes without going into low level specifics of the implementation.
- Comprises
  - IoT Functional Block
  - IoT Communication Model
  - IoT Communication API's

Functional Block

- Provides capabilities for
  - Identification
  - Sensing
  - Actuation
  - Communication and management


APPLICATION
MANAGEMENT
SERVICES
COMMUNICATION
SECURITY
DEVICE

- **Device:**
  - IoT i.e Internet of things, where things refer to the IoT devices which have unique identities and **can perform remote sensing, actuating and monitoring** capabilities (ex: combination of sensors, actuators, Arduino, non IoT devices).
  - Devices can share information with as well as collect information from other connected devices and applications (directly and indirectly).
  - They can process the data locally or in the cloud to find greater insights and put them into action based on temporal and space constraints (i.e space memory, processing capabilities, communication latencies and speeds and deadlines)
  - IoT devices can be of varied types. For ex: wearable sensors, smart watches, LED lights, automobiles and industrial machines.

- **Communication:**

Handles the communication for the IoT system.

various communication protocols which allows different devices to communicate with each other by sharing some information. It also allows interoperability among different devices

- **Services:**

  services for

  - device monitoring
  - device control service
  - data publishing services and
  - services for device discovery.

- **Management:** this blocks provides various functions to govern the IoT system.

- **Security:** this block secures the IoT system and by providing functions such as authentication , authorization, message and content integrity, and data security.

- **Application:** This is an interface that the users can use to control and monitor various aspects of the IoT system. Application also allow users to view the system status and view or analyze the processed data.

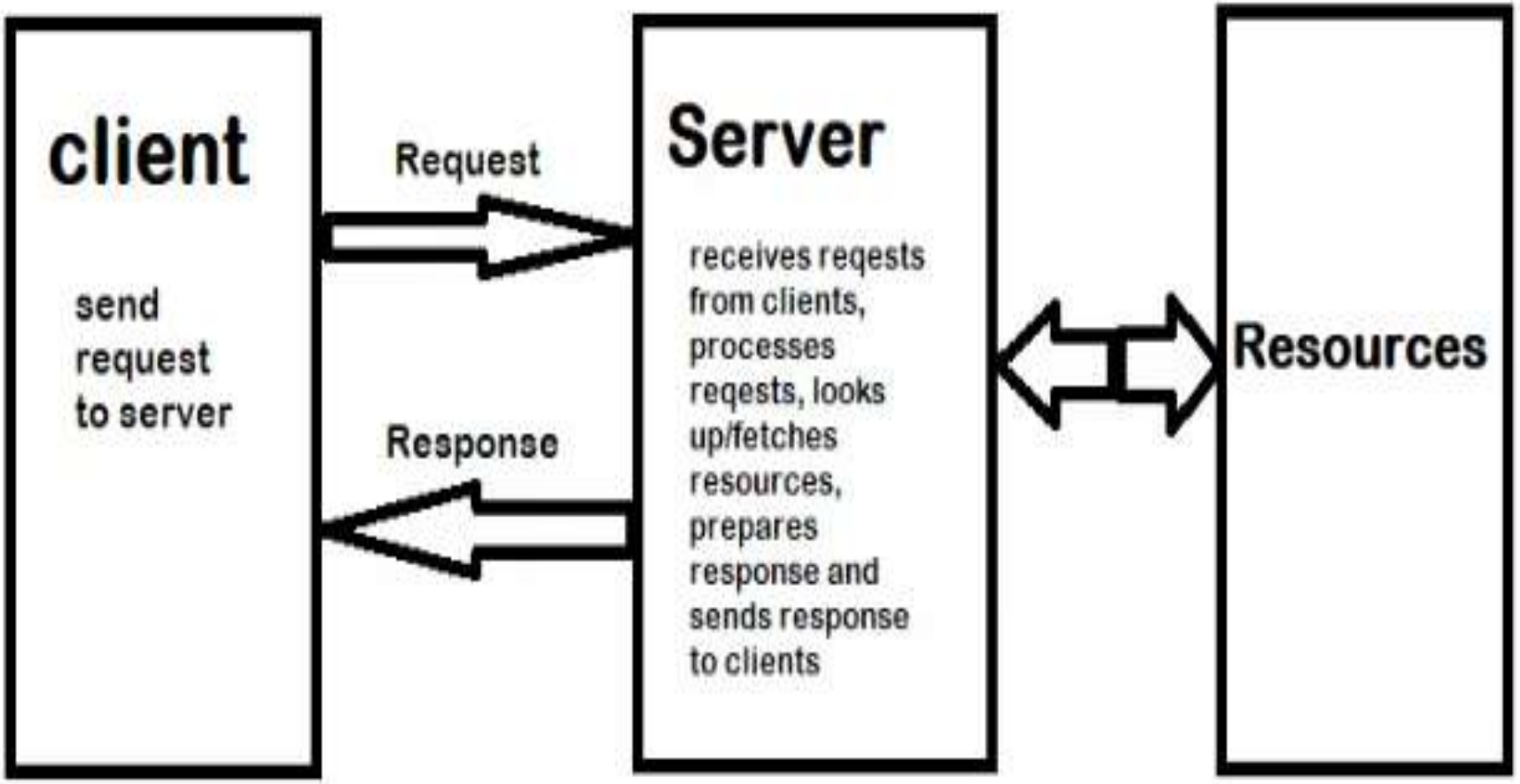
IoT Communication Model

- Request-Response Model
- Publish-Subscribe Model
- Push-Pull Model
- Exclusive Pair Model

Request Response Model

- In this model, the **client sends requests to the server and the server responds to the requests**.
- When the **server** receives a request, it **decides how to respond, fetches the data, retrieves resource representation, prepares the response, and then sends the response to the client.**
- Request-response is a **stateless communication model** and each request-response pair is independent of others.
- **HTTP works as a** request-response **protocol between a client and server.**
- A web browser may be the client, and an application on a computer that hosts a web site may be the server.

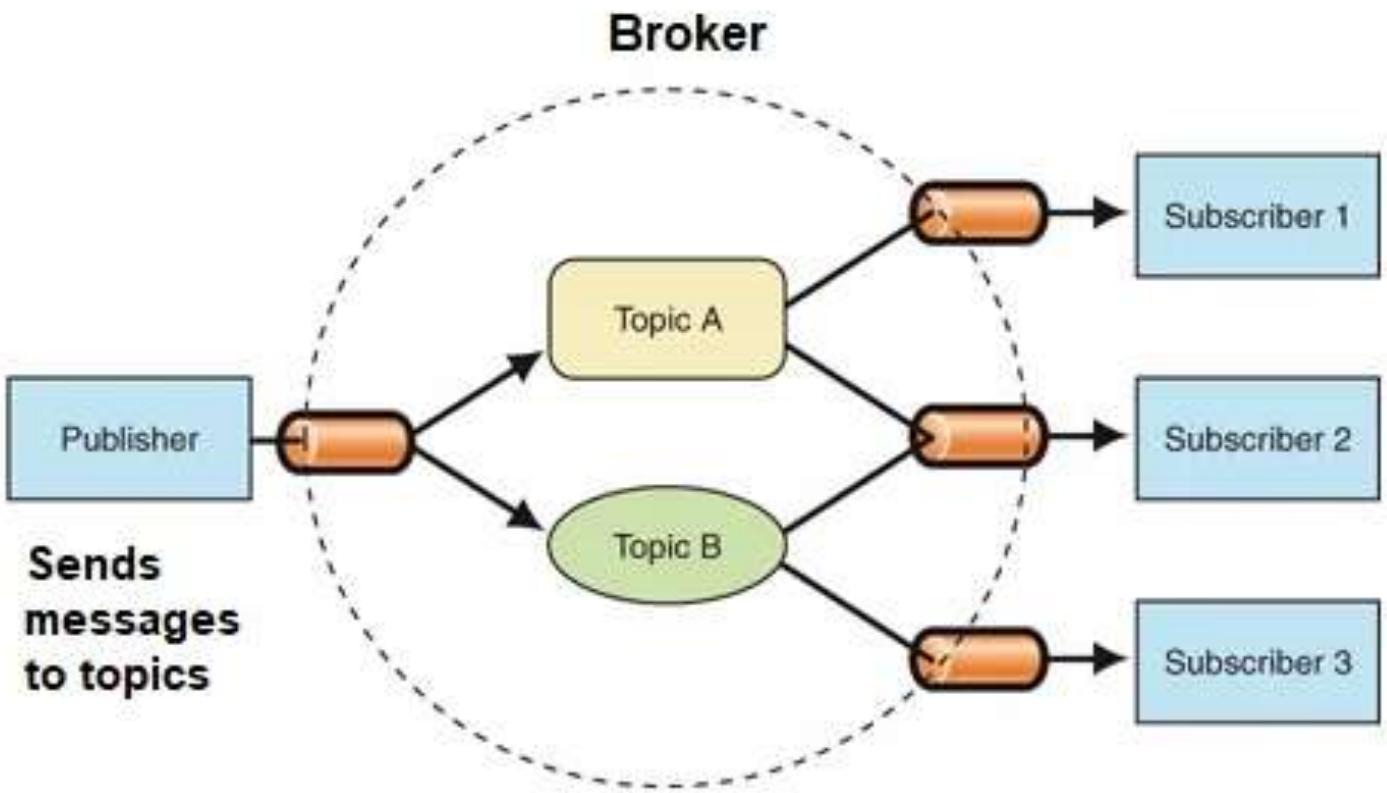
- Example: A client (browser) submits an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content.



Request-Response Communication Model

- Ex: HTTPs requests to log in some secure sites like IRCTC. Here only client can request to the server. i.e request is unidirectional but data transfer is bi-directional.

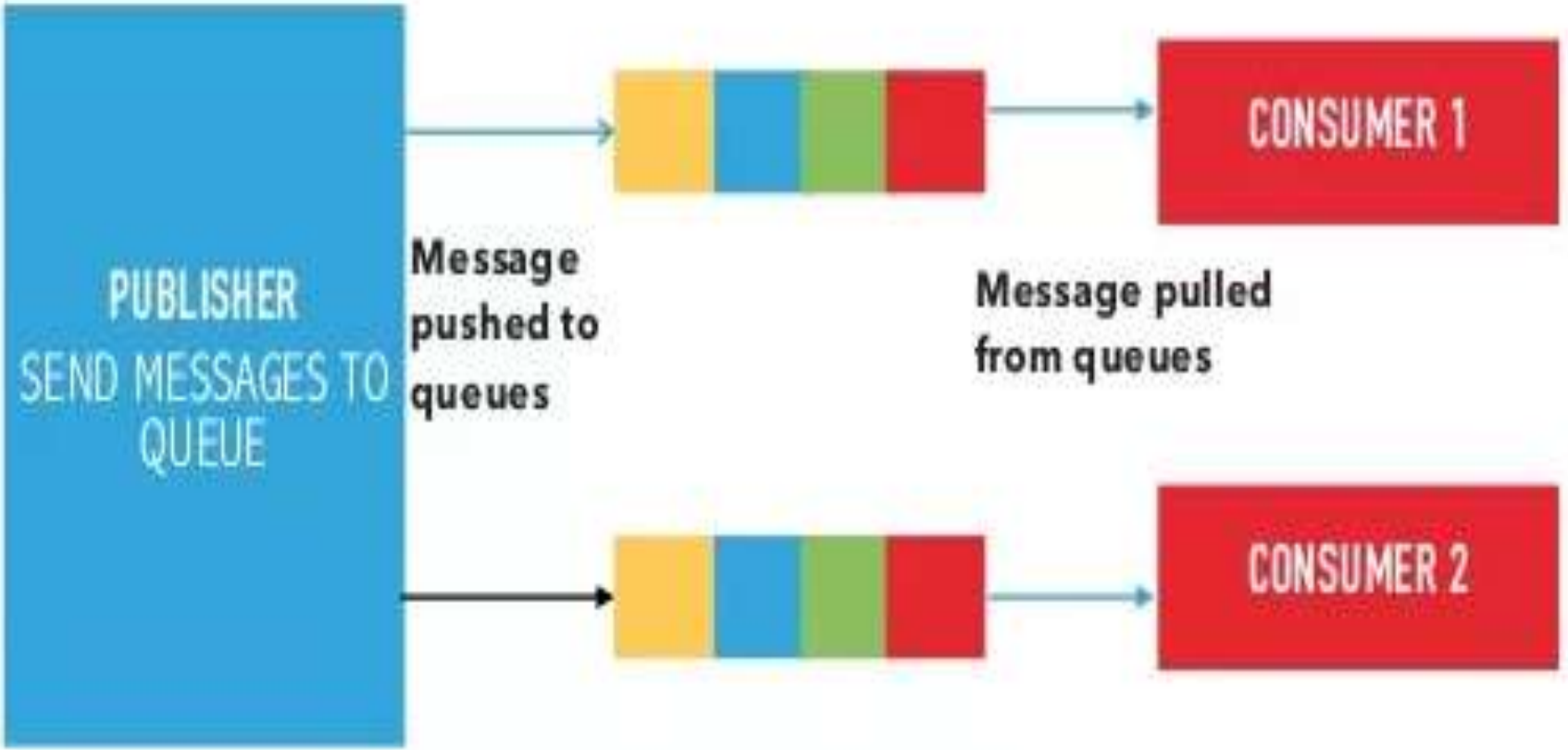
Publish Subscribe Model



- Involves publishers, brokers and consumers.
- **Publishers are the source of data.**
  - Publishers send the data to the topics which are managed by the broker.
  - Publishers are not aware of the consumers.
- Consumers subscribe to the topics which are managed by the broker.
- When the broker receive data for a topic from the publisher, it sends the data to all the subscribed consumers.

It also follows client-server architecture.

Push Pull Model



- In this model, the **data producers push the data to queues** and **the consumers Pull the data from the Queues**.
- **Producers do not need to be aware of the consumers.**
- Queues help in decoupling the messaging between the Producers and Consumers.
- **Queues also act as a buffer** which helps in situations when there is a mismatch between the rate at which the producers push data and the rate at which the consumer pull data.

Exclusive Pair Model

- It is a bidirectional, fully duplex communication model that uses a persistent connection between the client and server.
- Connection is setup it remains open until the client sends a request to close the connection.
- Client and server can send messages to each other after connection setup.
- Exclusive pair is **stateful communication model** and the server is aware of all the open connections.

It also supports client-server architecture

IoT Communication APIs

Application programming interface is a set of requirements that govern how one application can talk to another. API's do all these things by exposing some of program's internal functions to the outside world in a limited fashion.

- These IoT Communication APIs are:
  - REST-based Communication APIs
  - WebSocket-based Communication APIs

REST-based Communication APIs

- Representational state transfer (REST)

These are the set of architectural **principles** by which **you can design Web services , the Web APIs** that **focus on system's resources and how resource states are addressed and transferred.**

- Follows Request Response Model
- Unidirectional
- Its constraints are applied to
  - The components
  - Connector
  - Data Elements

- REST Architecture Constraints are:
  - **Client-server**
  - **Stateless**
  - **Cache able**
  - **Layered System**
  - **Uniform Interface**
  - **Code on demand**

- **Client Server:**

  – Principle behind this constraint is the separation.

  Example: clients should not be concerned with the storage of data which is concern of the server. Similarly the server should not be concerned about the user interface, which is concern of the client. Separation allows client and server to be independently developed and updated.

- **Stateless**:

Each request from client to server must contain all the information necessary to understand the request, and cannot take advantage of any stored context on the server. The session state is kept entirely on the client.

- **Cache-able**

Cache constraints requires that the data within a response to a request to be implicitly or explicitly lebelled as cache-able or non cache-able.

If a response is **cache-able, then a client cache is given the right to reuse that respsonse data for later**, equivalent requests.

The data can be cached in client side so that it can be reused when requested for the next time in order to minimize the time.

Caching can partially or completely eliminate some instructions and **improve efficiency and scalability.**

- **Layered system**

Layered system constrains, constrains the behavior of components such that **each component cannot see beyond the immediate layer with they are interacting.**

For example, the client cannot tell whether it is connected directly to the end server or two an intermediary along the way.

- **Uniform interface** :

Uniform interface constraints requires that the **method of communication between client and server must be uniform.**

Resources are identified in the requests (by URIs in web based systems) and are themselves is separate from the representations of the resources data returned to the client.

When a client holds a representation of resources it has all the information required to update or delete the resource you (provided the client has required permissions).

Each message includes enough information to describe how to process the message.

- **Code on demand** :

Servers can provide executable code or scripts for clients to execute in their context. this constraint is the only one that is optional.

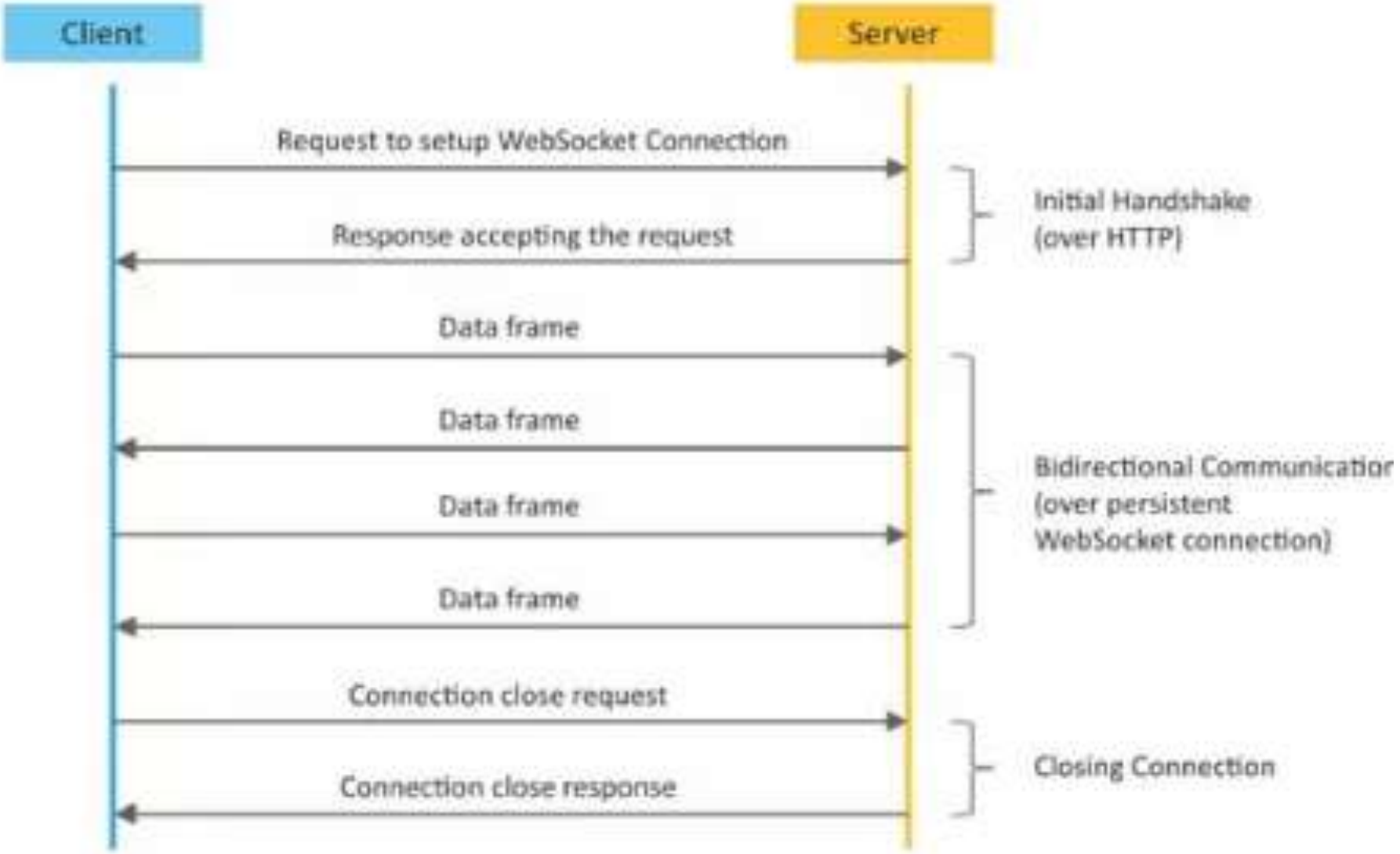
- **Scalability:**

It supports both horizontal and vertical scalability. As it is stateless so scalability is easier to implement.

Web Socket Based Communication API

- Websocket APIs allow **bi-directional, full duplex** communication between clients and servers.
- Follow the **exclusive pair communication model**.
- Therefore it **does not require new connection** to be setup for each message to be sent

- Communication **begins with a connection setup request sent by the client to the server.**
- The **request** (called websocket handshake) is **sent over HTTP** and the server interprets it.
- **If the server supports websocket protocol**, the **server responds** to the websocket handshake response.
- **After the connection setup client and server can send data/messages to each other in full duplex mode.**


Client
Server
Request to setup WebSocket Connection
Response accepting the request
Initial Handshake
(over HTTP)
Data frame
Data frame
Data frame
Data frame
Bidirectional Communicatior
(over persistent
WebSocket connection)
Connection close request
Connection close response
Closing Connection

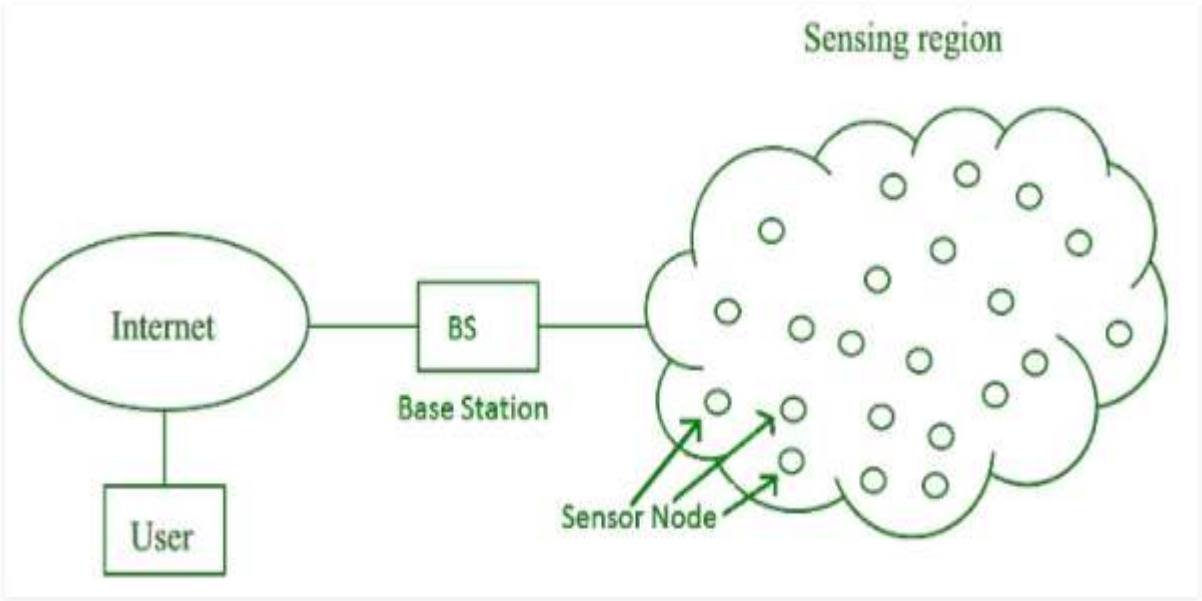
- **Reduce the network traffic and latency** as there is **no overhead for connection setup and termination requests for each message**.
- Websocket **suitable for IoT applications that have low latency or high throughput requirements**.
- So Web socket is most suitable IoT Communication APIs for IoT System.

IoT Enabling Technologies

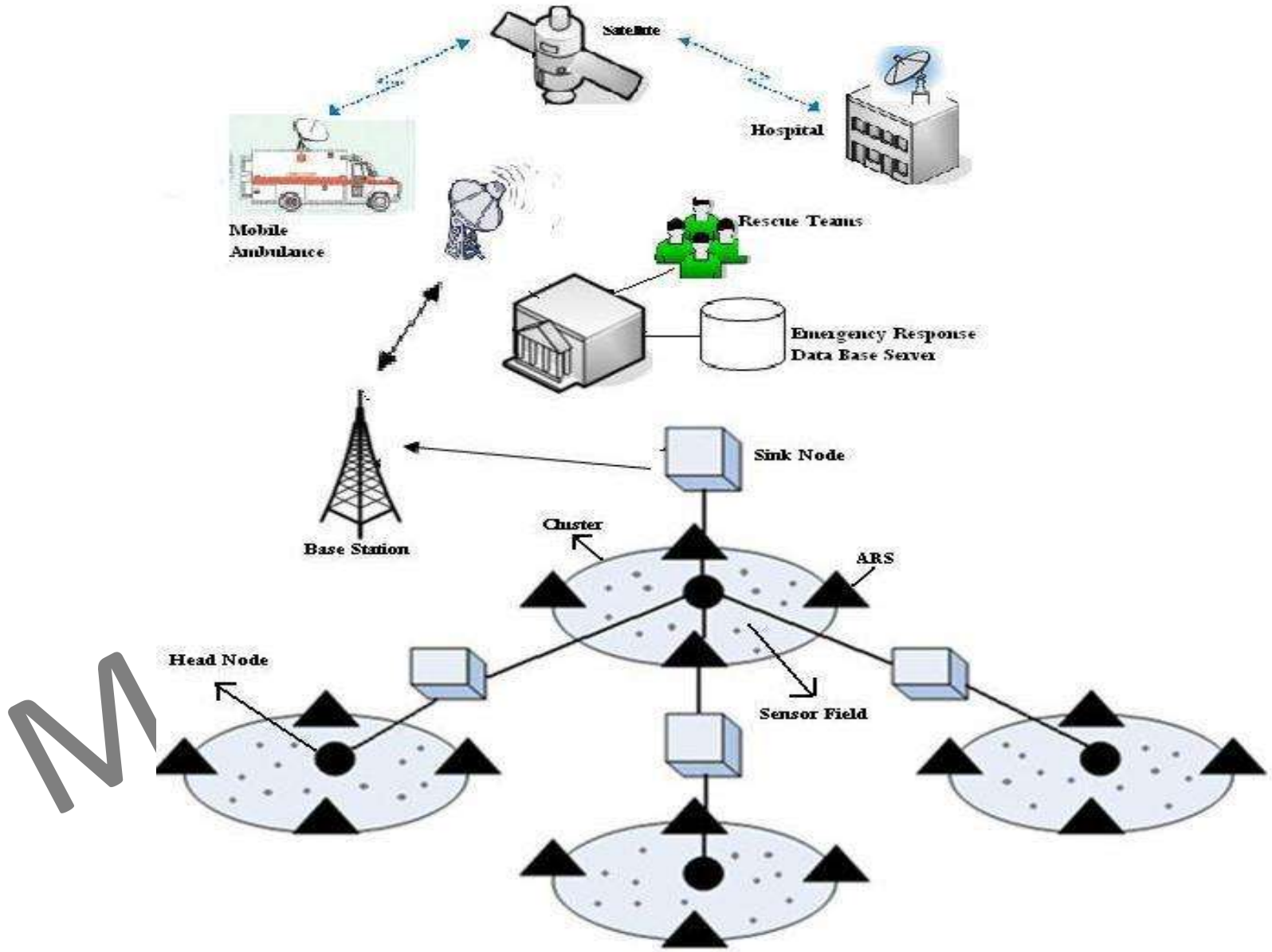
- IoT is enabled by several technologies which plays key role in IoT such as
  - WSN
  - CC
  - Big Data Analytics

- **Wireless Sensor Network:**
  - Comprises of **distributed device with sensor** which are used **to monitor the environmental and physical conditions**.
  - Consists of
    - A number of end-nodes,
    - Routers and
    - A coordinator.

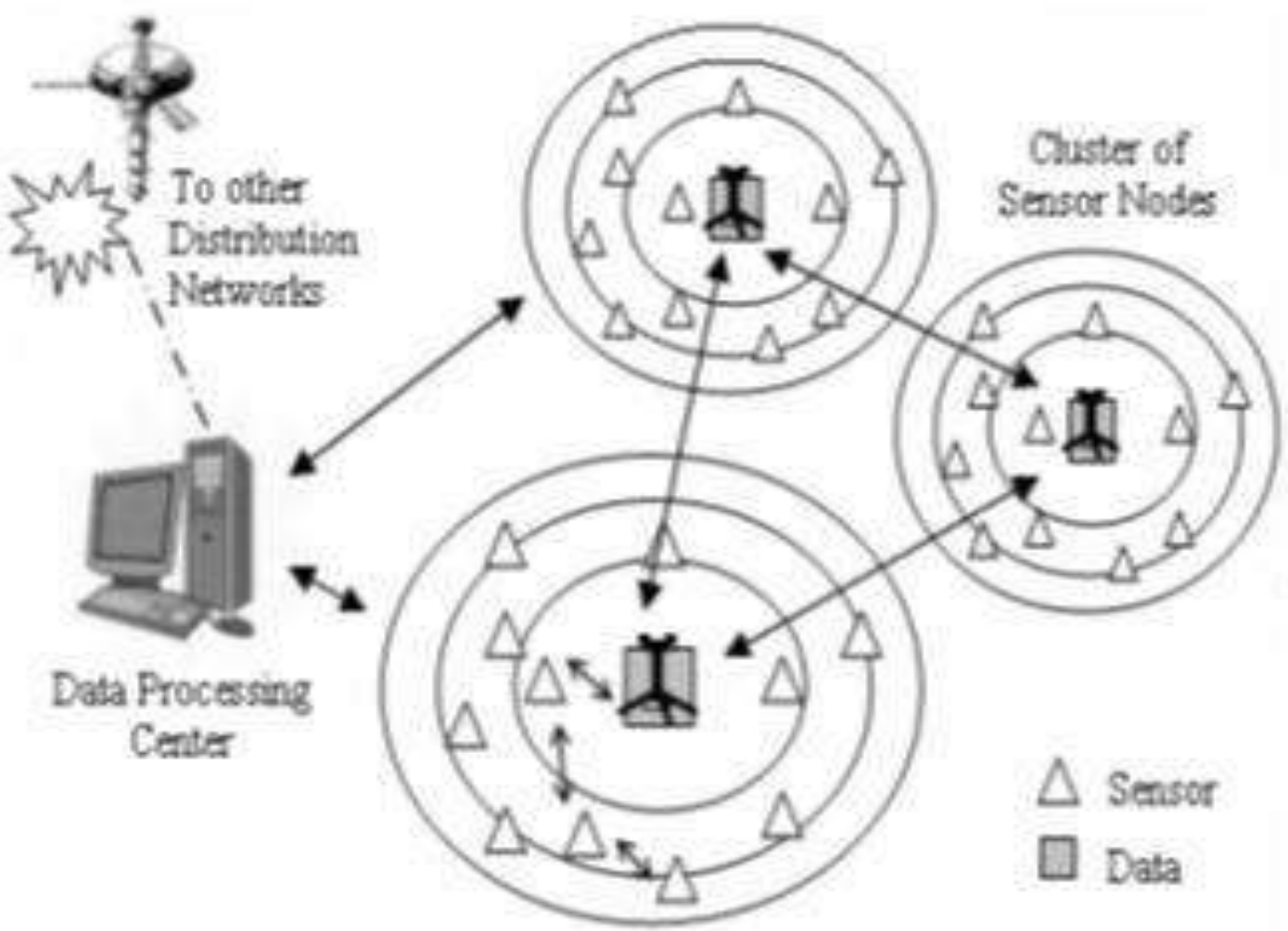
- End Nodes **have several sensors** attached to them in node can also act as routers.
- Routers **are responsible for routing the data packets from end-nodes to the coordinator**.
- The **coordinator collects the data from all the nodes.**
- Coordinator also **act as a gateway that connects the WSN to the internet.**



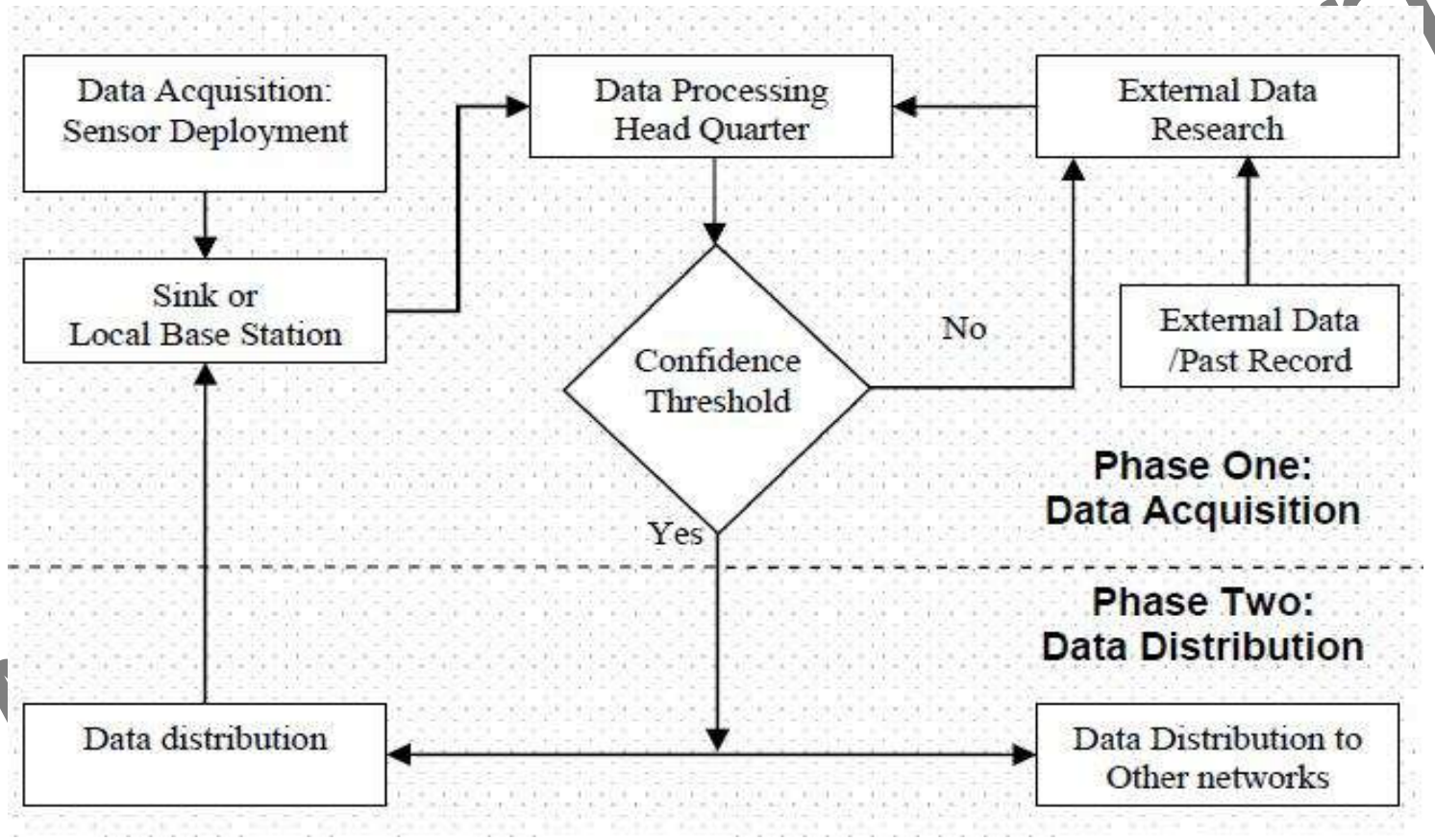
- **Sensors:**
  Sensors in WSN are used to capture the environmental variables and which is used for data acquisition. Sensor signals are converted into electrical signals.
- **Radio Nodes/router**
  It is used to receive the data produced by the Sensors and sends it to the WLAN access point. It consists of a microcontroller, transceiver, external memory, and power source.
- **WLAN Access Point/Coordinator:**
  It receives the data which is sent by the Radio nodes wirelessly, generally through the internet.
- **Evaluation Software:**
  The data received by the WLAN Access Point is processed by a software called as Evaluation Software for presenting the report to the users for further processing of the data which can be used for processing, analysis, storage, and mining of the data.



Wireless Sensor Network Architecture for disaster survivor detection



Data collection and aggregation in wireless sensor network


Data Acquisition:
Sensor Deployment
Data Processing
Head Quarter
External Data
Research
Sink or
Local Base Station
Confidence
Threshold
No
External Data
/Past Record
Phase One:
Data Acquisition
Yes
Phase Two:
Data Distribution
Data distribution
Data Distribution to
Other networks

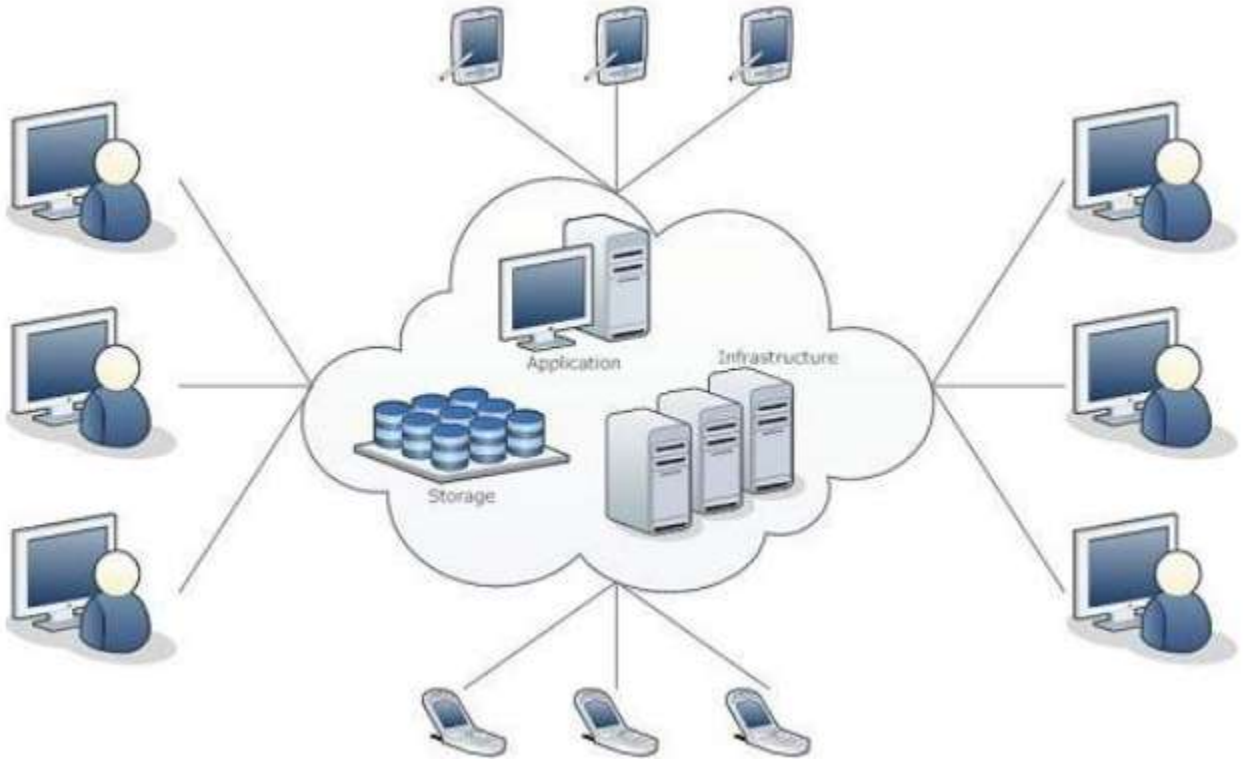
- Weather monitoring system
- Indoor air quality monitoring systems
- Soil moisture monitoring system
- Surveillance system
- Structural health monitoring system

- Weather monitoring system use WSNs in which the nodes collect **temperature humidity and other data** which is aggregated and analysed.
- Indoor air quality monitoring systems use WSNs to collect data on the **indoor air quality and concentration of various gases**
- Soil moisture monitoring system use WSNs to monitor **soil moisture at various locations**.
- Surveillance system use WSNs for collecting **Surveillance data (such as motion detection data)**
- Structural health monitoring system use WSNs to **monitor the health of structures (buildings, bridges) by collecting vibration data from sensor nodes de deployed at various points in the structure**.

- **Cloud Computing**
  - What is Cloud?
    - The term **Cloud** refers to a **Network** or **Internet.**
    - Cloud is something, which is **present at remote location**.
    - Provide services over public and private networks, i.e., WAN, LAN or VPN.

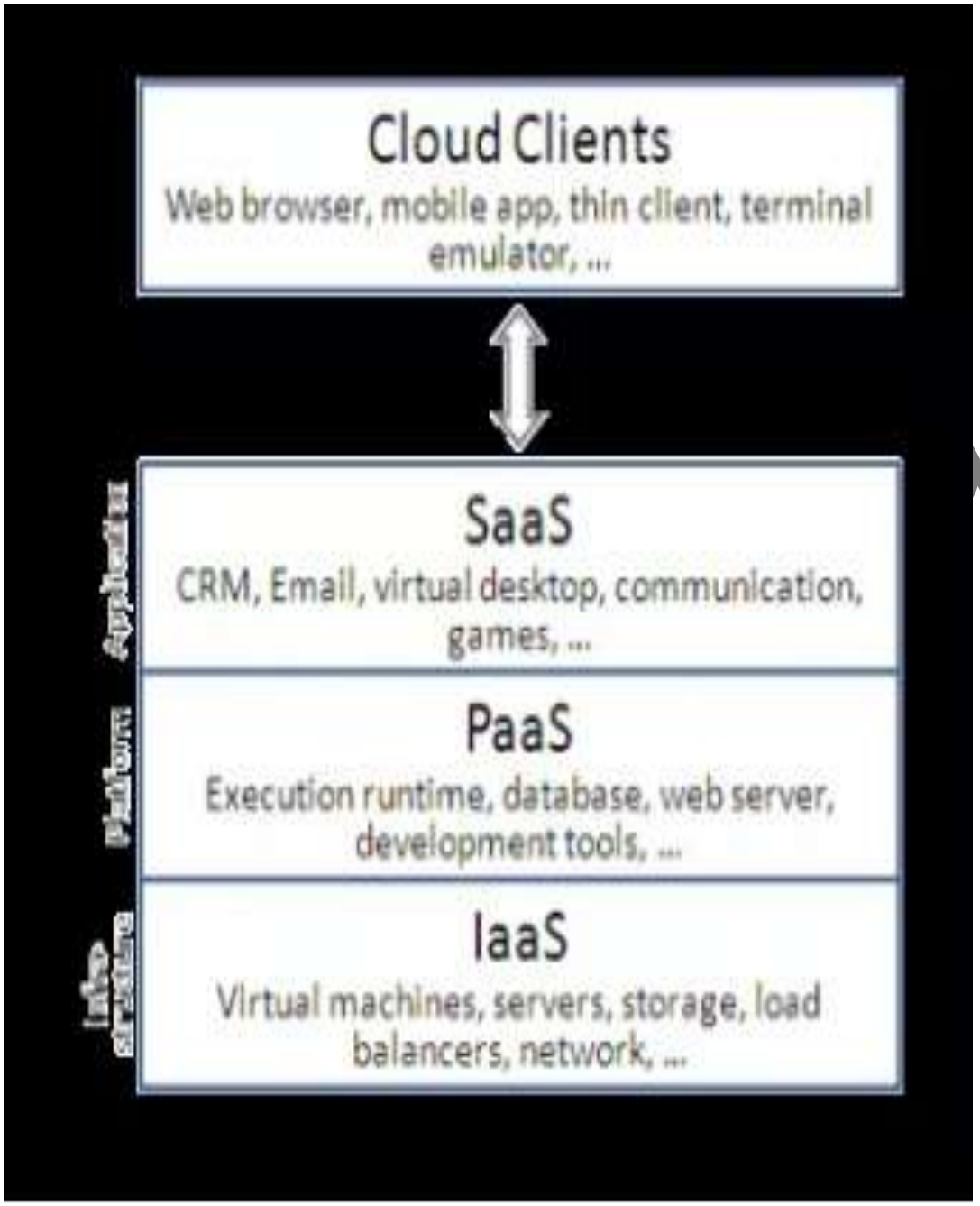
– What is Cloud Computing?

- Cloud Computing refers to **manipulating, configuring,** and **accessing** the hardware and software resources remotely. It offers online data storage, infrastructure, and application.



Offers **platform independency,** a s the software is not required to be installed locally on the PC

- It’s a computing paradigm that **involves delivering applications and services over the Internet.**
- It involves **provisioning of computing, networking and storage resources on demand.**
- **Provisioning resources is automated**.
- Cloud computing **resources can be accessed over the network using standard access mechanisms** that provide **platform independent access** through the use of heterogeneous client platforms such as the workstations, laptops, tablets and smartphones.



- **Infrastructure as a Service (IaaS)**
- **Platform as a Service (PaaS)**
- **Software as a Service (SaaS)**

- **Infrastructure as a Service (IaaS): hardware is provided by an external provider** and managed for you
- **Platform as a Service (PaaS):** in addition to hardware, **your operating system layer is managed for you**
- **Software as a Service (SaaS):** further to the above, an **application layer is provided** and managed for you – you won't see or have to worry about the first two layers.

- **Advantages:**
  - Faster Implementation
  - Anywhere access to application and content
  - High scalability to meet demand
  - Higher utilization of infrastructure investments
  - Enhanced security and protection of information assets

- **Big Data Analytics**
  - Is the process of **collecting, organizing and analysing** large sets of data (*called* Big Data) **to discover patterns and other useful information**.
  - Can **help organizations to better understand the information contained within the data** and **will also help identify the data that is most important to the business and future business decisions**.
  - Analysts working with Big Data typically want the *knowledge* that comes from analysing the data.

- Collection of data whose volume, velocity or variety is too large and difficult to store, manage, process and analyse the data using traditional databases.
  - It involves data cleansing, processing and visualization
  - Lots of data is being collected and warehoused
    - Web data, e-commerce
    - purchases at department/ grocery stores
    - Bank/Credit Card transactions
    - Social Network

- **Velocity Refers to speed at which data is processed**
  - Batch
  - Real-time
  - Streams
- **Variety Includes different types of data**
  - Structured
  - Unstructured
  - Semi Structured

- **Volume refers to the amount of data**
  - Terabyte
  - Records
  - Transactions
  - Files
  - Tables

- Example:
  - **Sensor data generated by IoT** system such as weather monitoring stations.
  - **Machine sensor data** collected from sensors **embedded in industrial and energy systems** for monitoring their health and detecting Failures.
  - **Health and fitness data** generated by IoT devices such as wearable fitness bands
  - Data generated by IoT systems for location and tracking of vehicles
  - Data generated by retail inventory monitoring systems

IoT Levels and Deployment Templates

An IoT system comprises the following components:

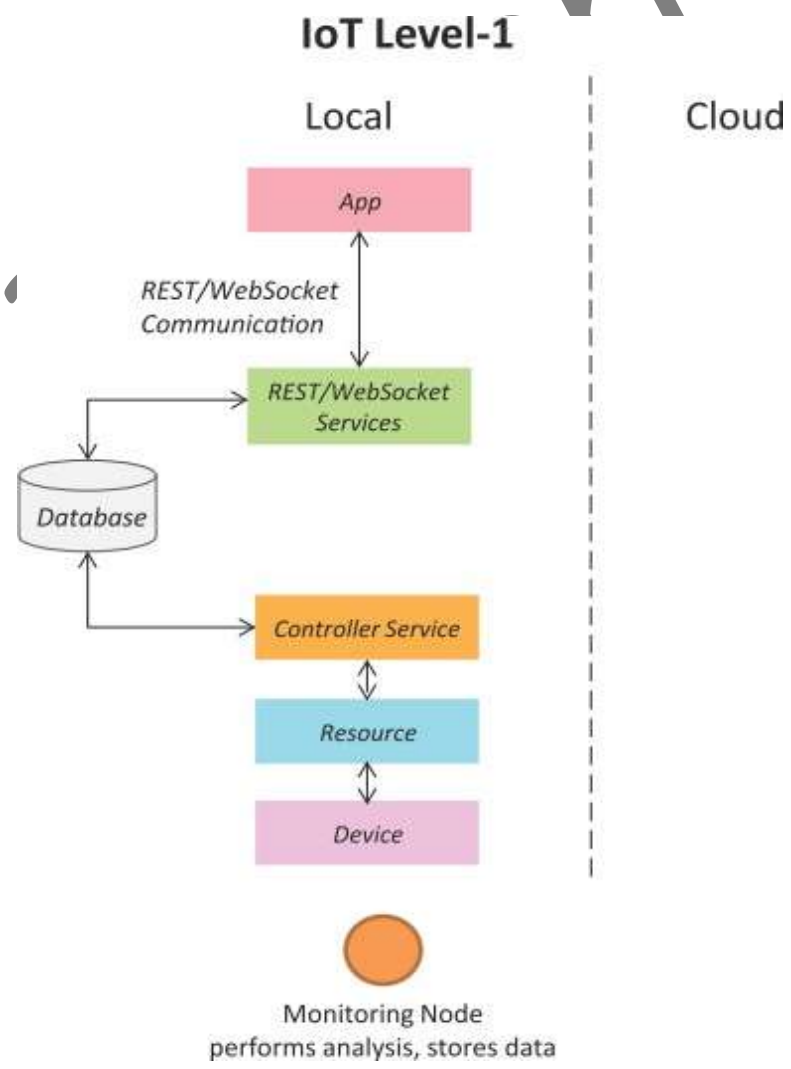
- **Device**: An IoT device allows **identification**, remote **sensing**, **actuating** and remote **monitoring** capabilities.
- **Resource**: Resources are **software components** on the IoT device for **accessing**, **processing** and **storing sensor information**, or for **controlling actuators connected to the device**. Resources also include the software **components that enable network access for the device.**
- **Controller Service**: Controller service **is a native service** that runs **on the device** and **interacts with the web services**. Controller service **sends data from the device to the web service** and **receives commands from the application (via web services) for controlling the device.**

- **Database**: Database can be **either local or in the cloud** and stores the data generated by the IoT device.
- **Web Service**: Web services serve as a **link between the IoT device, application, database and analysis components**.Web service can be implemented using HTTP and REST principles (REST service) or using the WebSocket protocol (WebSocket service).
- **Analysis Component**: This is responsible for **analyzing the IoT data** and generating results in a form that is easy for the user to understand.
- **Application**: IoT applications **provide an interface that the users can use to control and monitor various aspects of the IoT system**. Applications also allow users to view the system status and the processed data.

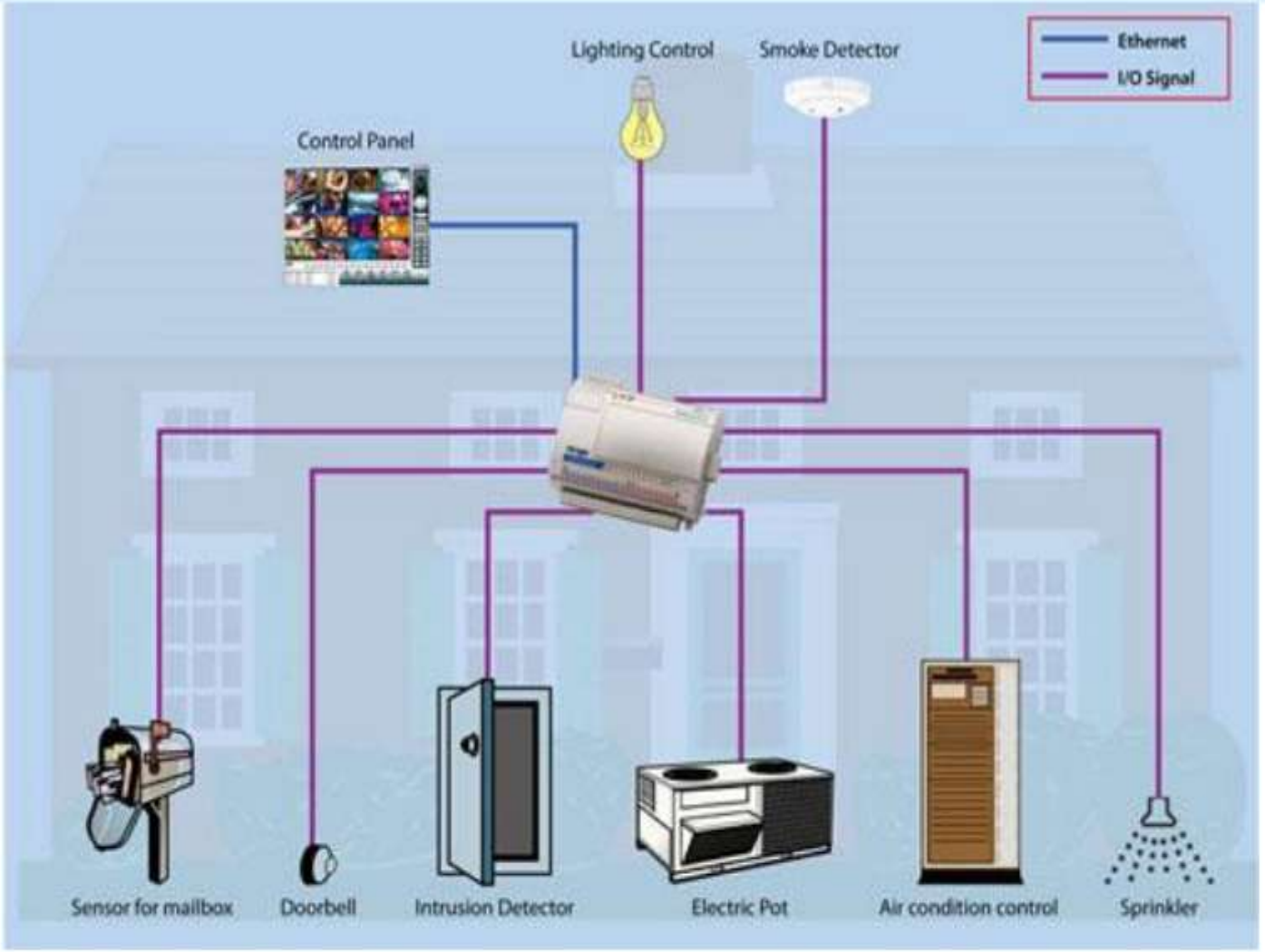
IoT Level 1

- A level - 1 IoT system **has a single node/device** that performs **sensing and/or actuation, stores data, performs analysis** and **hosts the application.**

- Level - 1 IoT systems are **suitable for modelling low - cost and low - complexity solutions** where the data involved is not big and the **analysis requirements are not computationally intensive** .

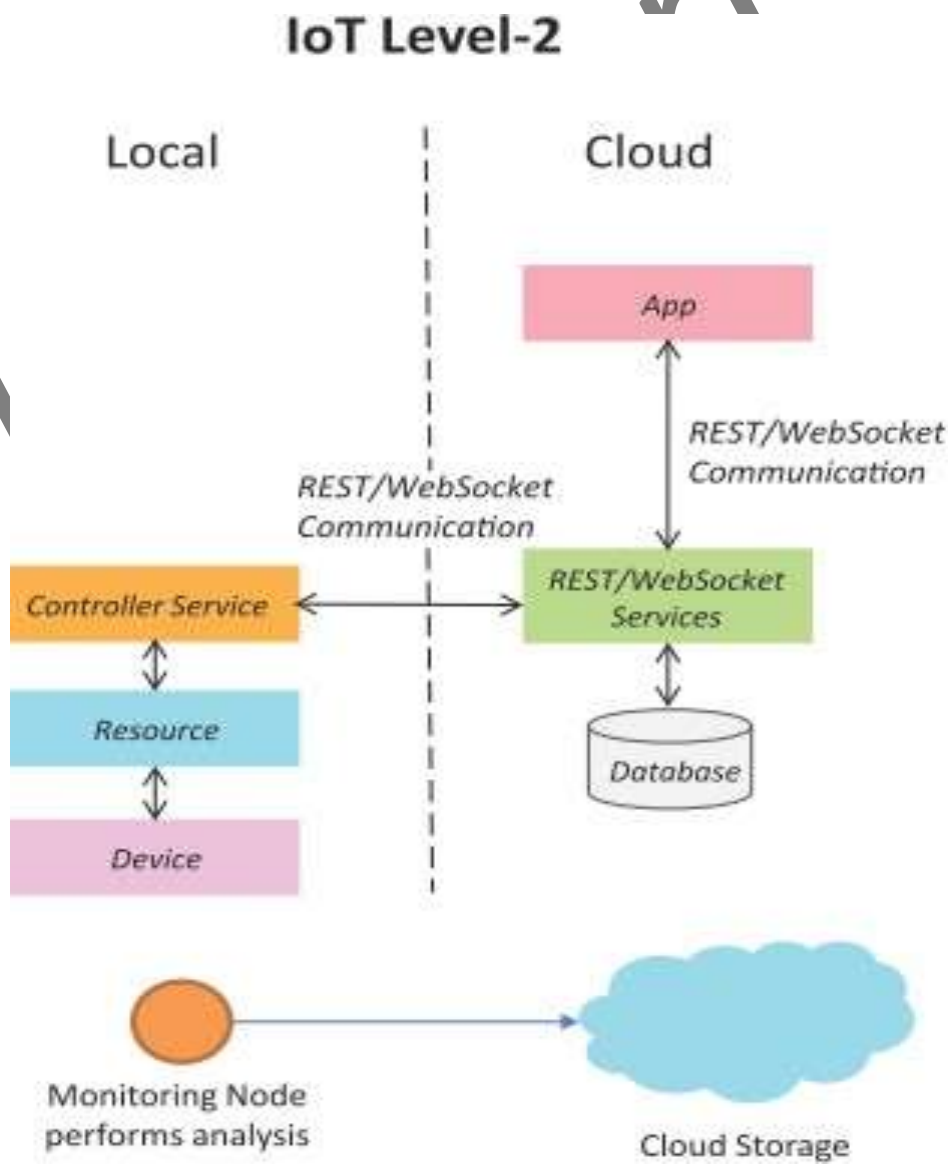


IoT – Level 1 Example …Home Automation System



IoT Level-2

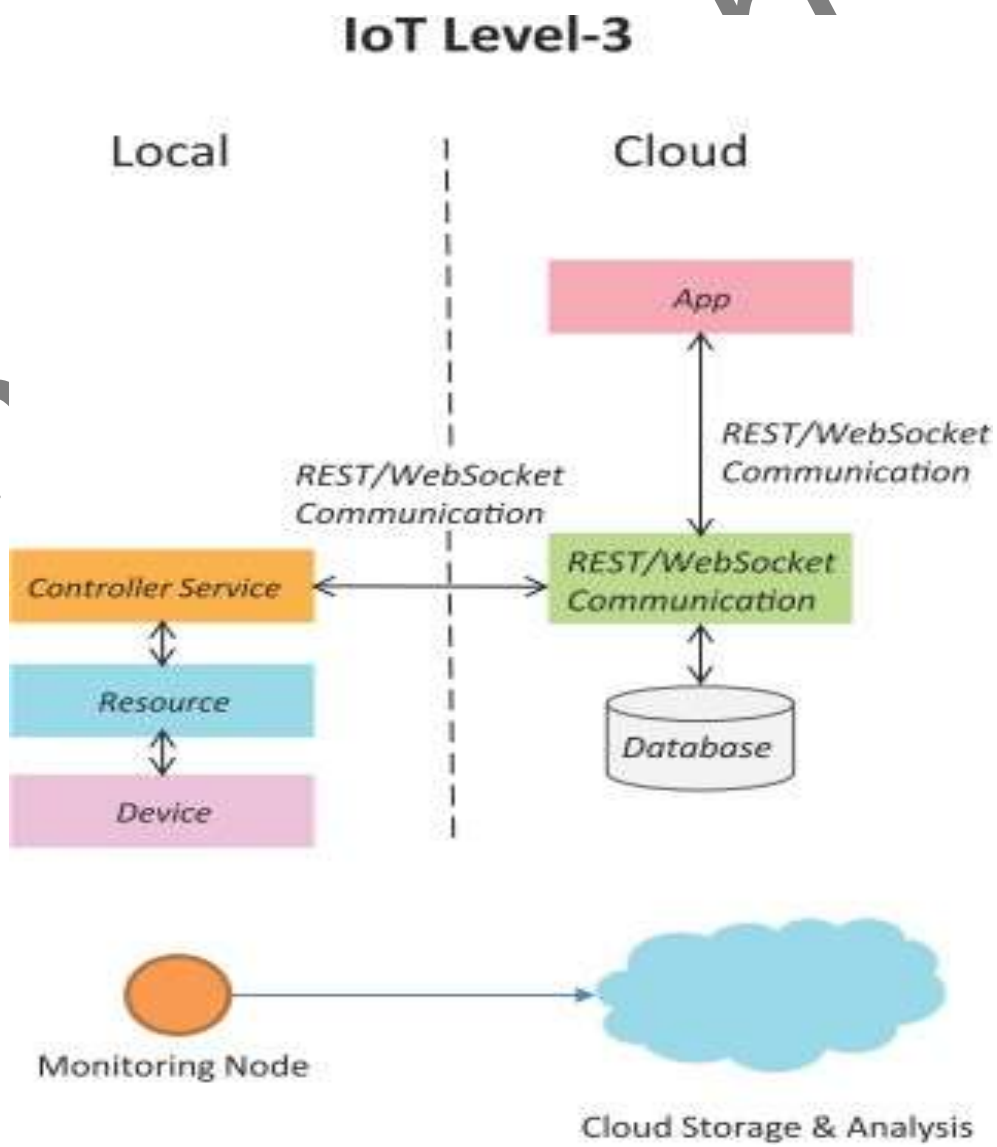
- A level-2 IoT system has a **single node that performs sensing and/or actuation and local analysis.**
- **Data is stored in the cloud** and the application is usually cloud-based.
- Level-2 IoT systems are **suitable for solutions where the data involved is big;** however, the primary **analysis requirement is not computationally intensive** and can be done locally.



IoT – Level 2 Example …Smart Irrigation

IoT Level-3

- A level -3 IoT system **has a single node. Data is stored and analysed in the cloud and the application is cloud - based.**
- Level - 3 IoT systems are **suitable for solutions where the data involved is big and the analysis requirements are computationally intensive .**

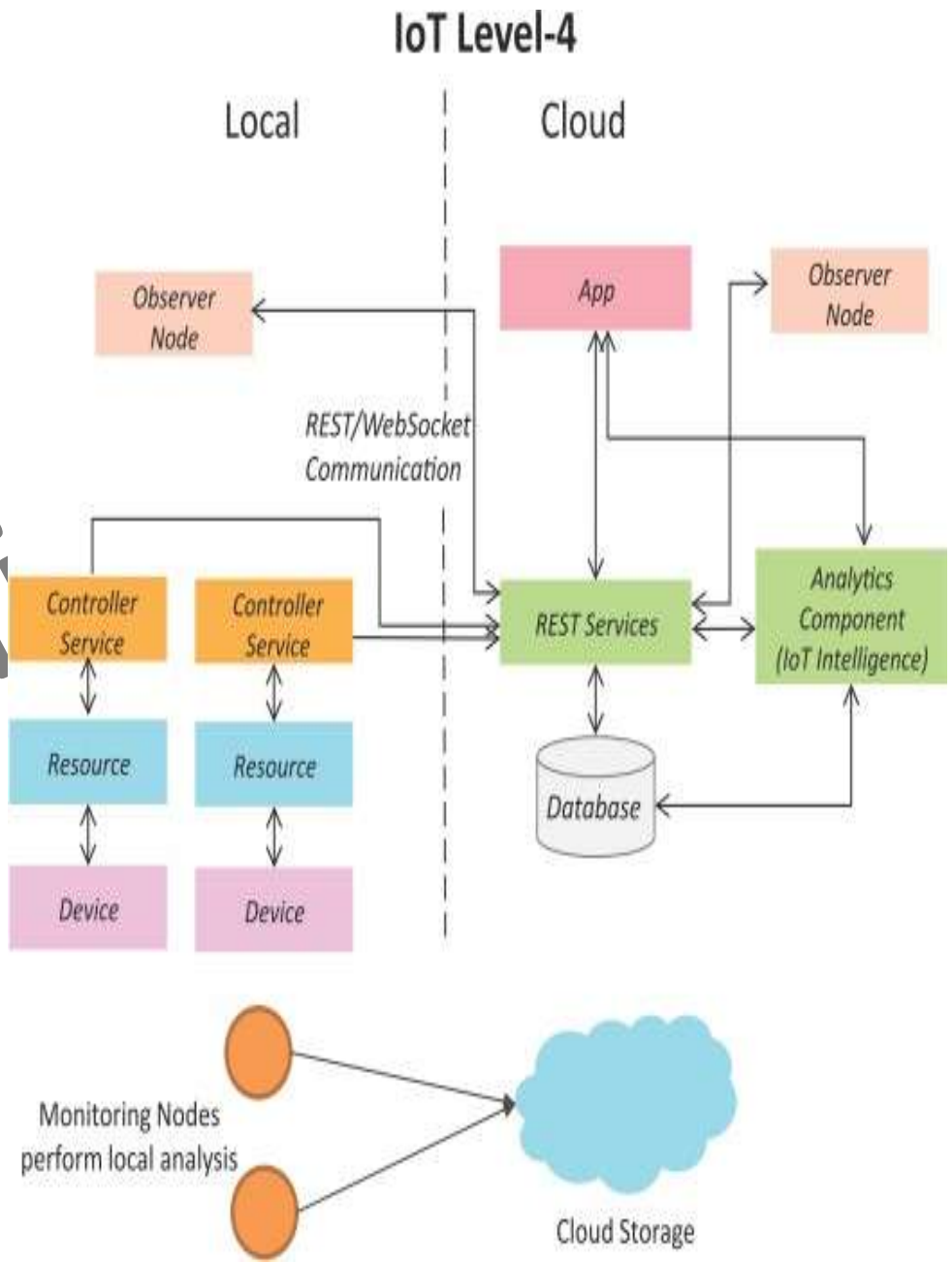


IoT – Level 3 Example …Tracking Package Handling

Sensors used accelrometer and gyroscope

IoT Level-4

- A level-4 IoT system **has multiple nodes that perform local analysis.** **Data is stored in the cloud and the application is cloud-based.**
- Level-4 contains **local and cloud- based observer nodes** which can subscribe to and receive information collected in the cloud from IoT devices.
- Level-4 IoT systems are **suitable for solutions where multiple nodes are required, the data involved is big and the analysis requirements are computationally intensive**.

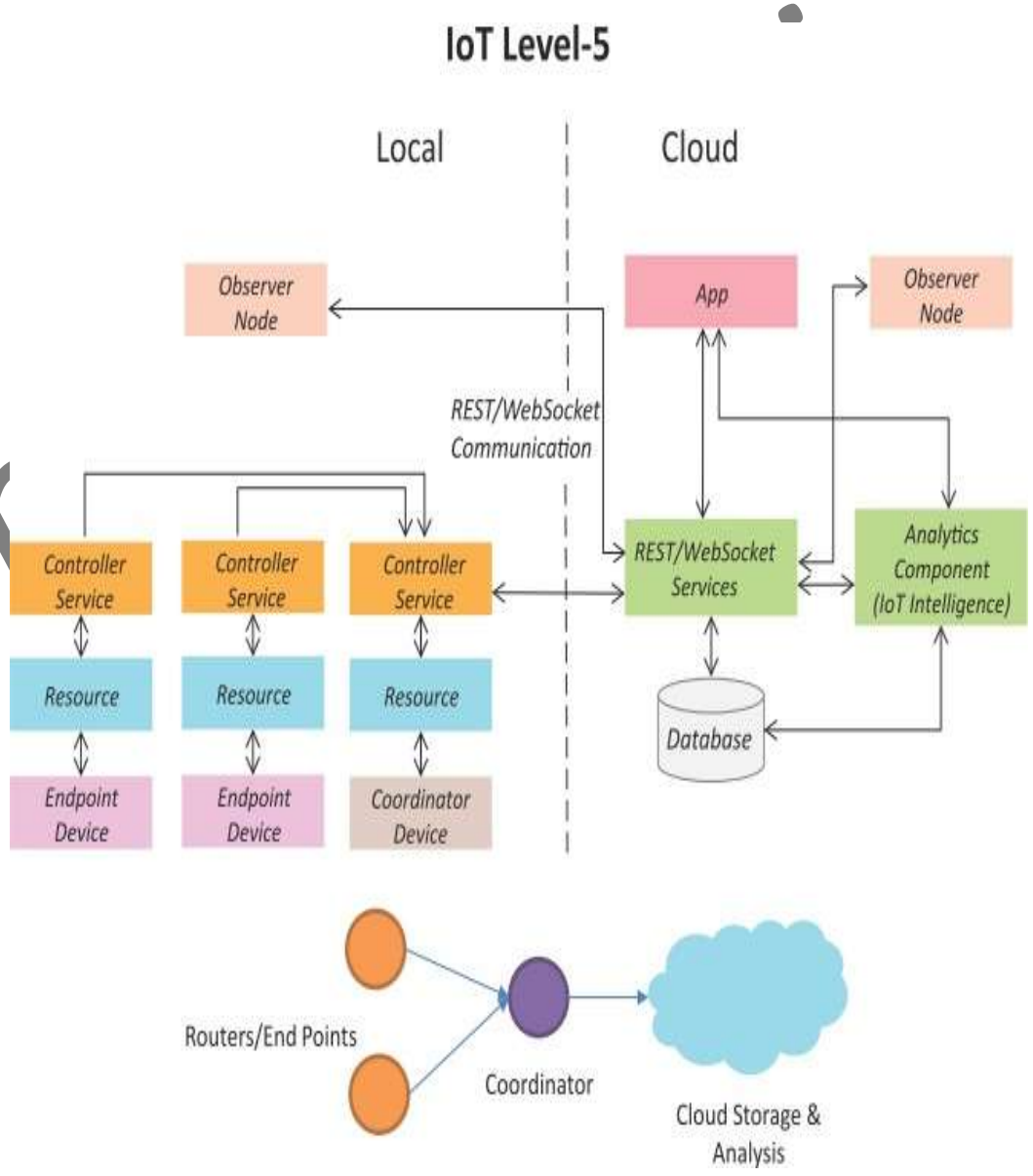


IoT – Level 4 Example …Noise Monitoring

Sound Sensors are used

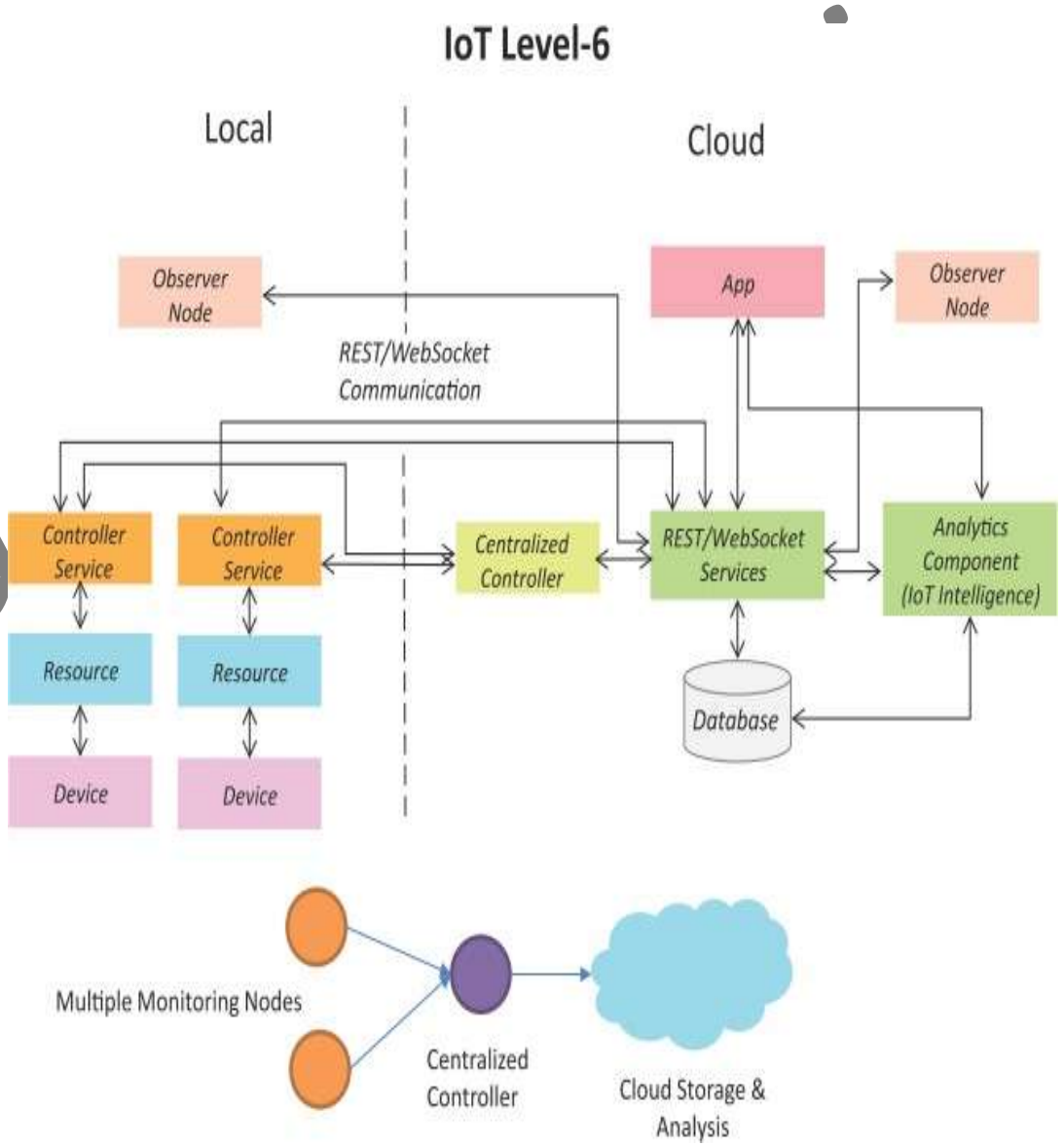
IoT Level-5

- A level-5 IoT system **has multiple end nodes and one coordinator node.**
- The **end nodes perform sensing and/or actuation**.
- The **coordinator node collects data from the end nodes and sends it to the cloud**.
- Data **is stored and analyzed in the cloud and the application is cloud- based**.
- Level-5 **IoT systems are suitable for solutions based on wireless sensor networks**, in which the data involved is big and the analysis requirements are computationally intensive.



IoT Level-6

- A level-6 IoT system has multiple independent end nodes that perform sensing and/or actuation and send data to the cloud.
- Data is stored in the cloud and the application is cloud-based.
- The analytics component analyzes the data and stores the results in the cloud database.
- The results are visualized with the cloud-based application.
- The **centralized controller is aware of the status of all the end nodes and sends control commands to the nodes**.



IoT Issues, Challenges and Application

- **IoT Challenges and Issues:**
  - Data Privacy
  - Data Security
  - Insurance Concern
  - Lack of Common Standard
  - Technical Concern
  - Security attacks and System Vulnerabilities
  - System Security
  - Application Security
  - Network Security

- **Application :**
  - **Home Automation**
    - **Smart Lightening**
    - **Smart Appliances**
    - **Intrusion Detection**
    - **Smoke/Gas Detector**

- **Cities**
  - **Smart Parking**
  - **Smart Lightening**
  - **Smart Roads**
  - **Surveillance**
  - **Emergency Response**

- **Environmental**
  - **Weather Monitoring**
  - **Air Pollution Monitoring**
  - **Noise Pollution Monitoring**
  - **Forest Fire Detection**
  - **River Flood Detection**

- **Retail:**
  - **Inventory Management**
  - **Smart Payments**
  - **Smart Vending Machines**

Physical Design of IoT

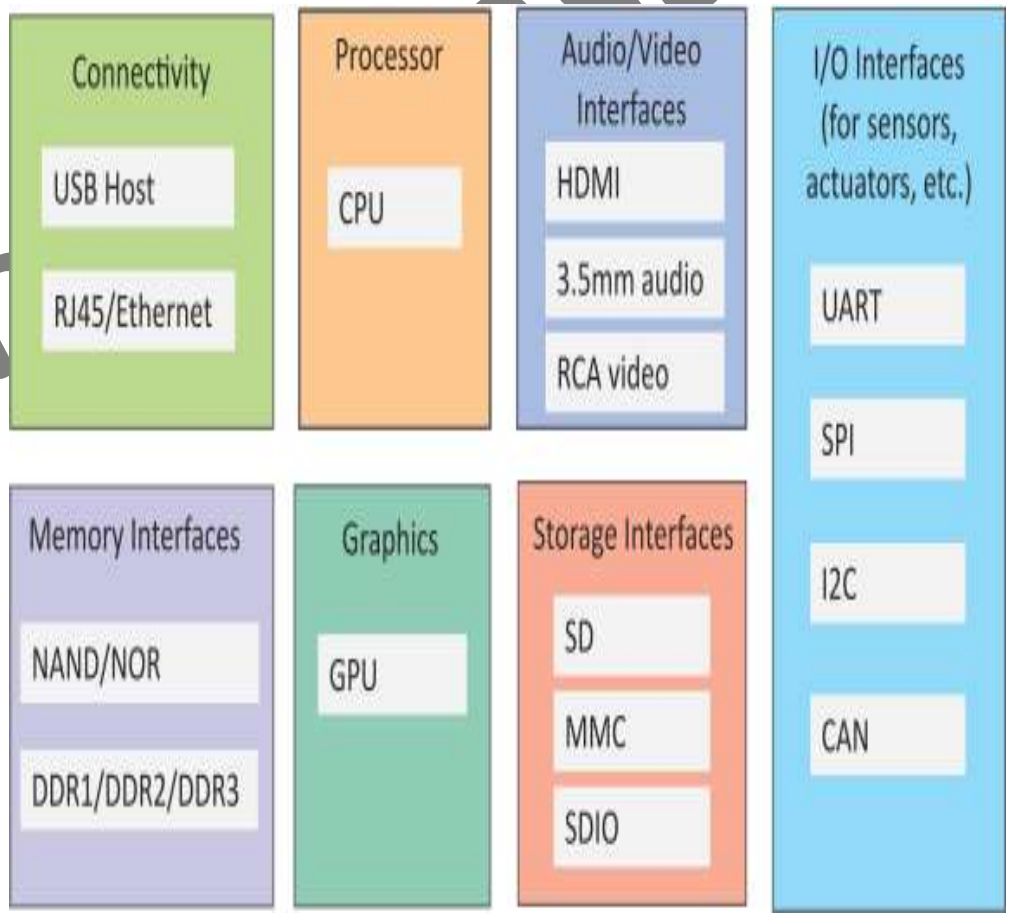
- Things in IoT
- IoT Protocols

Things in IoT

- Refers to IoT devices **which have unique identities that can perform sensing, actuating and monitoring capabilities.**
- IoT devices **can exchange** data with other connected devices or **collect data** from other devices and **process the data** either locally or send the data to centralized servers or cloud – based application back-ends for processing the data.

Generic Block Diagram of an IoT Device

- An IoT device may consist of several interfaces for connections to other devices, both wired and wireless.
  - I/O interfaces for sensors
  - Interfaces for internet connectivity
  - Memory and storage interfaces
  - Audio/video interfaces



- **Connectivity (RJ 45, USB):** This component is used to **connect the IoT device to the internet**.
- **Processing unit (CPU):** This is used to **control** all the other components as well as it **processes** the instruction present in the algorithm.
- **Audio/ video unit:** This is used to **connect the monitor or speaker.**
- **HDMI:** High definition multimedia interface. It supports uncompressed all digital audio video interfaces which **provides all digital audio and video via a single cable.**
  - HDMI provides an interface between any audio/video source such as **set-top box, DVD player or audio/ video receiver and audio and or video monitor such as a digital television over a single cable.** HDMI supports high definition video, plus multi-channel digital audio on a single cable

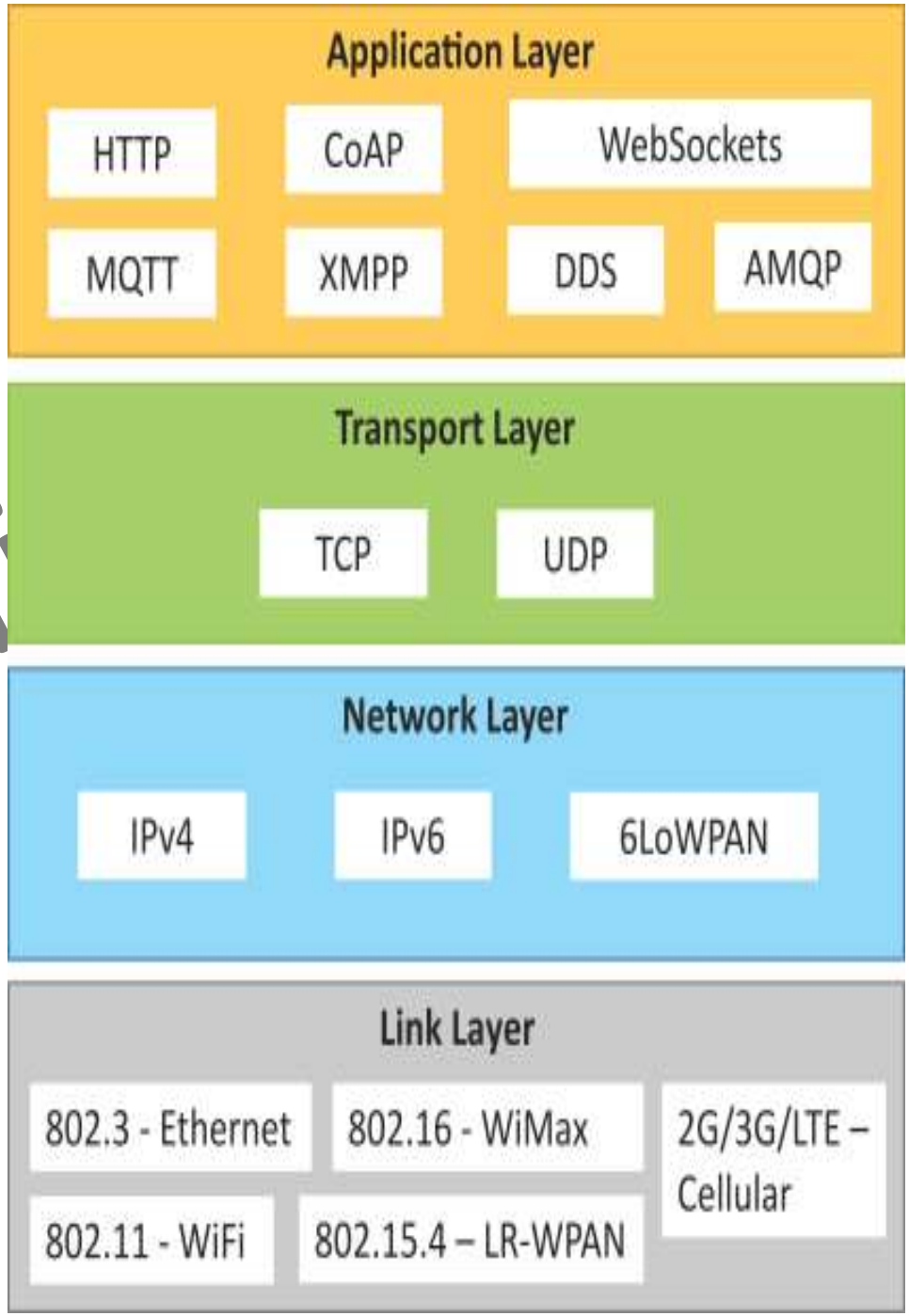
- **2.5/3.5 mm audio:** This is used **for audio**.
- **RCA:** Radio Corporation of America. It is sometimes called as **phono connector**. It is a type of **electrical connector commonly used to carry audio and video signals.**
- **I/O interfaces:** This is used **to connect sensors, actuators or any external devices**.
- **UART:** universal Asynchronous Receiver Transmitter. It is used to **provide serial communication.**

- **SPI:** Serial peripheral interface. It is used to **provide serial communication between different devices or you can say between monitoring node and controlling node or coordinator node.**
- **I2C:** Inter integrated circuit. Here **multiple masters can communicate with multiple slaves.**

- **CAN:** Controller area network. It is designed to **allow micro controller and devices to communicate with each other in applications without a host computer.**
- **Storage:** This is used to store or **record the sensed data**.
- **SD:** Secure digital. This is used **to hold the SD card which** is a non-volatile memory card format. SD card is a popular storage media for digital cameras and other mobile devices.
- **SDIO:** It is used to **connect external HDD**.
- **MMC:** Multi Media card. It is a **flash memory based memory card standard.** It is a popular storage media for digital cameras and other mobile devices.
- **GPU:** Graphics processing unit. It is useful **for the resolution of the screen and used for high end games**.
- **Memory:** There are used for temporary storage and for execution of instruction.

IoT Protocols

- Link Layer
  - 802.3 – Ethernet
  - 802.11 – WiFi
  - 802.16 – WiMax
  - 802.15.4 – LR-WPAN
  - 2G/3G/4G
- Network/Internet Layer
  - IPv4
  - IPv6
  - 6LoWPAN
- Transport Layer
  - TCP
  - UDP
- Application Layer
  - HTTP
  - CoAP
  - WebSocket
  - MQTT
  - XMPP
  - DDS
  - AMQP



Link Layer Protocol

- Determine **how the data is physically sent over the network physical layer or medium**
- HOW THE PACKETS ARE CODEDE AND SIGNALED

IoT Protocols…Link Layer…Ethernet

Sr.No	Standard	Shared medium
1	802.3	Coaxial Cable…10BASE5
2	802.3.i	Copper Twisted pair …..10BASE-T
3	802.3.j	Fiber Optic……10BASE-F
4	802.3.ae	Fiber…..10Gbits/s

Collection of Wired Ethernet Standard

Data Rates are provided from **10Gbit/s to 40Gb/s and higher**

- 10BASE5 is derived from several characteristics of the physical medium.
- The 10 refers to its transmission speed of 10 Mbit/s.
- The BASE is short for baseband signaling (as opposed to broadband), and the 5 stands for the maximum segment length of 500 meters (1,600 ft).

IoT Protocols...Link Layer...WiFi

Sr.No	Standard	Operates in
1	802.11a	5 GHz band
2	802.11b and 802.11g	2.4GHz band
3	802.11.n	2.4/5 GHz bands
4	802.11.ac	5GHz band
5	802.11.ad	60Hz band

- Collection of Wireless LAN standard
- Data Rates from **1Mb/s to 6.75 Gb/s**
- An **idle wifi** based network reaches around **100 m**, as it maximum range.
- Generally, Can transfer data at speeds up to 54mbps.

IoT Protocols…Link Layer…WiMax.. Worldwide interoperability for microwave access

Sr.No	Standard	Data Rate
1	802.16m	100Mb/s for mobile stations 1Gb/s for fixed stations

- Collection of Wireless Broadband standards
- Data Rates from **1.5Mb/s to 1 Gb/s**
- An idle wi-max network **can reach about 80-90 kms**
- Generally, Can **exchange data at speed upto 40 mbps.**

IoT Protocols...Link Layer...LR(Low Rate)-WPAN

- Collection of standards for low-rate wireless personal area networks
- It is **applicable for power constrained device providing low cost and low speed communication.**
- Data Rates from 40Kb/s to 250Kb/s

IoT Protocols…Link Layer…2G/3G/4G – Mobile Communication

Sr.No	Standard	Operates in
1	2G	GSM-CDMA
2	3G	UMTS and CDMA 2000
3	4G	LTE

- Data Rates from 9.6Kb/s (for 2G) to up to 100Mb/s (for 4G)

IoT Protocols…Network/Internet Layer

- Responsible **for sending of IP datagrams from source to destination network**
- Performs the **host addressing and packet routing**
- **Host identification** is **done** using hierarchical IP addressing schemes such as **IPV4 or IPV6**

IoT Protocols…Network Layer

- **IPV4**
  - ✓ Used to identify the devices on a network
  - ✓ Uses 32-bit address scheme
- **IPV6**
  - ✓ Uses 128-bit address scheme

IPv4	IPv6
Internet protocol version 4	Internet protocol version 6
It is 32 bit dotted decimal	It is 128 bit colon hexa decimal
It supports 2^{32} number of addresses	It supports 2^{128} number of addresses
It supports point to multipoint connection.	It supports point to point connection
All the applications have same priority.	Different applications have different priority.
It is connection less.	It is connection oriented.

- **6LoWPAN (IPV6 over Low power Wireless Personal Area Network)**
  - ✓ **Used for devices with limited processing capacity**
  - ✓ It brings IP protocol to the low power devices which have limited processing capability.
  - ✓ Operates **in 2.4 Ghz**
  - ✓ Data Rates **of 250Kb/s**

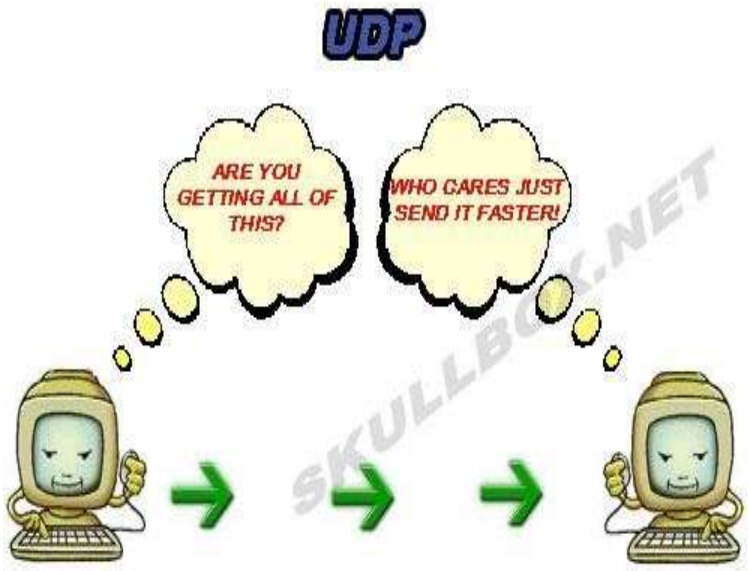
IoT Protocols...Transport Layer

- **Provide end-to-end message transfer capability** independent of the underlying network.
- It provides functions such as **error control, segmentation, flow- control and congestion control**

IoT Protocols…TCP

- Transmission Control Protocol
- **Connection Oriented**
- Ensures Reliable transmission
- Provides **Error Detection Capability** to ensure **no duplicacy of packets and retransmit lost packets**
- **Flow Control capability** to ensure the **sending data rate is not too high for the receiver process**
- **Congestion control** capability helps in avoiding congestion which leads to degradation of n/w performance

IoT Protocols…UDP



- User Datagram Protocol
- Connectionless
- Does not ensures Reliable transmission
- Does not do connection before transmitting
- Does not provide proper ordering of messages
- stateless

IoT Protocols…Application Layer…HTTP…Hyper Text Transfer Protocol

- Forms foundation of World Wide Web(WWW)
- Includes commands such as GET,PUT, POST, HEAD, OPTIONS, TRACE..etc
- Follows a request-response model
- Uses Universal Resource Identifiers(URIs) to identify HTTP resources

- It is a web transfer protocol.
- It suitable for web browsing.
- It is a stateless protocol.
- It uses ort number TCP: 80 in transport layer.
- It uses URI to identify HTTP Resources.
- It uses Request- response model.
- It uses client server architecture.
- It uses PUT, GET, POST, DELETE HEAD, TRACE, OPTIONS method to communicate with the resources.

IoT Protocols…Application Layer…CoAP

- **Constrained Application Protocol**
- Used for **Machine to machine (M2M) applications** meant for constrained devices and n/w's
- Web transfer protocol for IoT and uses request-response model
- Uses client –server architecture
- Supports methods such as GET,POST, PUT and DELETE

I. It is a web transfer protocol.

II. It is suitable for machine to machine communication.

III. It is applicable for constrained environment with constrained device and network. It is a stateful protocol.

IV. It uses UDP 5683 port number in transport layer.

V. It uses request- response model.

VI. It uses client server architecture.

VII. It uses PUT, GET, POST, DELETE, HEAD, TRACE, OPTIONS.

VIII. It is designed to work with HTTP.

IoT Protocols…Application Layer…WebSocket

- Allows full-duplex communication over single socket
- Based on TCP
- Client can be a browser, IoT device or mobile application

IoT Protocols…Application Layer…MQTT

- **Message Queue Telemetry Transport** , light-weight messaging protocol
- Based on publish-subscribe model
- Well suited for **constrained environments where devices have limited processing, low memory and n/w bandwith requirement**

- It is a low weight messaging protocol (small message size).
- It uses TCP as transport layer protocol and uses port number TCP 1883.

IoT Protocols…Application Layer…XMPP

- **Extensible messaging and presence protocol**
- For **Real time communication** and streaming XML data between n/w entities
- Used for Applications such as **Multi-party chat and voice/video calls.**
- It is a decentralised protocol.
- It uses TCP as transport layer protocol.
- Client server architecture allows client to server and server to server communication.

IoT Protocols…Application Layer…DDS

- **Data Distribution service** is a data-centric middleware standard for **device-to-device or machine-to-machine communication**.
- **Publish subscribe model where publishers create topics to which subscribers can use**.

IoT Protocols…Application Layer…AMQP

- **Advanced Messaging Queuing Protocol used for business messaging.**
- Supports both point-to-point and publisher/subscriber models, routing and queuing
- Broker here receives messages from publishers and route them over connections to consumers through messaging queues.
- It uses 5672 TCP.

IoT Devices and its features: Ardino,Uno,Rasberry Pi,Nodeu

- IoT devices connects wirelessly to a network with each other and able to transfer the data.
- Connected over internet, allows data transfer without human intervention
- Generally it includes
  - Computer devices
  - Software
  - Wireless Sensor
  - Actuators

- **Properties of IoT device:**
  - **Sense**
  - **Send and Receive data**
  - **Analyze**
  - **Controlled**

Arduino Uno

- IoT Microcontroller
- Arduino UNO is an **open source prototyping platform and a great place to start with electronics and programming**.
- Used to **build digital device that can be sense and control objects in the physical and the digital world.**

- With I/O pins
- Some has USB to load programs from the PC

Properties:

- **Inexpensive:** least expensive version cost < 50$
- **Cross-platform:** run on Win,Mac,Linux some time limited to Win
- **Simple, clear programming environment:** easy to use, flexible
- **Open source and extensible software:** open source and can be expanded through C++ library etc
- **Open source and extensible hardware**

- **Application:**
  - Arduino based RFID Sensed Device Access
  - Arduino based Industrial Appliances Control
  - Arduino based Underground Cable Fault Detection
  - Arduino based Home Automation

RASPBERRY PI

- The raspberry pi is small size of a credit card computer.
- The raspberry pi can be **easily plugged into monitor, computer or your TV**.
- Also, it **uses standard keyboard and mouse**.
- Even non technical **users depend on it for configuring their digital media systems and surveillance cameras**.
- Raspberry Pi 3 is certainly the **most affordable and powerful computing platform**.

- **Application:**
  - Tablet Computer
  - Home Automation
  - Media Streamer
  - Controlling Robots

NodeU(Node MCU-Node Microcontroller Unit)

- **Arduino and Raspberry Pi do not have built-in support for wireless networks.**
- Developers will have to add a wifi or cellular module to the board and write code to access the wireless module.
- **So open source IoT development board called NodeMCU is used and allows you to code your device.**
- One of its most unique features is **that it has built-in support for wifi connectivity.**
- It is a very inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems,
- ESP8266 contains all crucial elements of the modern computer: CPU, RAM, networking (wifi), and even a modern operating system and SDK.
- This makes NodeMCU a smart choice to play with the IoT

- **Application:**
  - Book Passes/Tickets on-line.
  - Teaching management system.
  - Security Alarm.

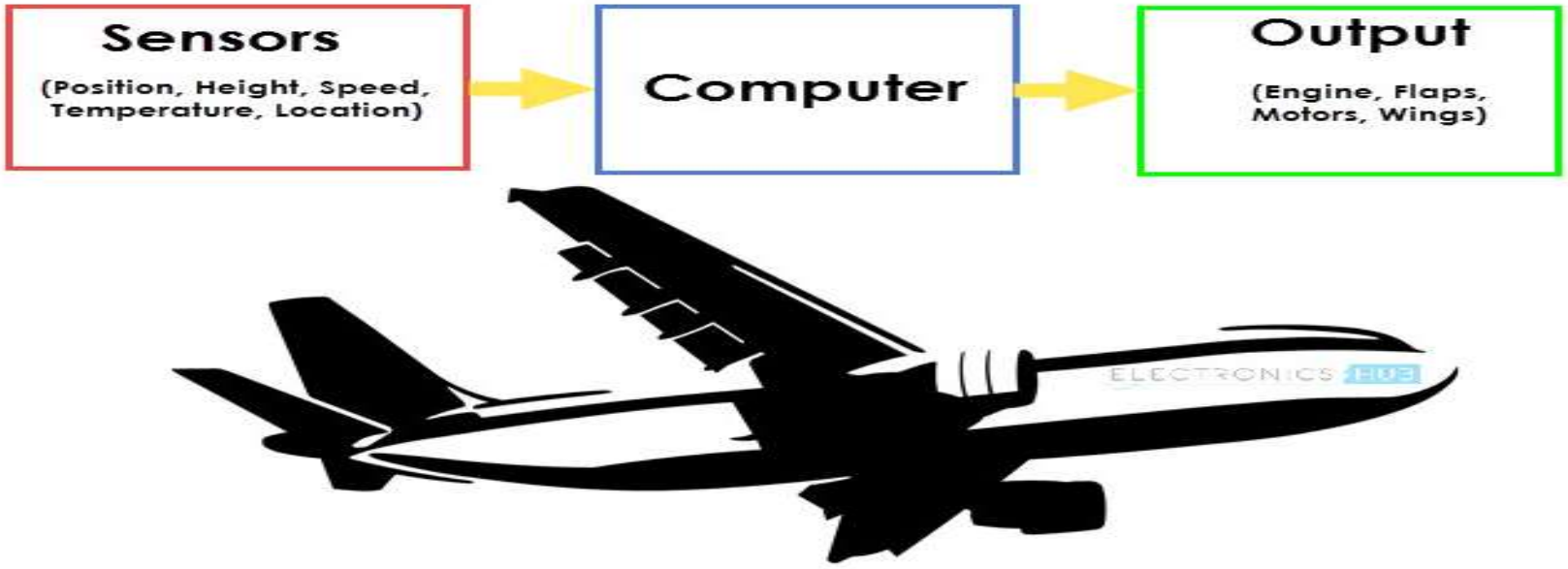
Case study on IoT Applications using various Sensors and Actuators

- **Sensor:**
  - A sensor is a **device that is able to detect changes in an environment.**
  - Able to measure a physical phenomenon (like temperature, pressure, and so on) and transform it into an electric signal.

- There are different types of Sensor but fundamentally they are of two types:
  - Analog : does not have ADC
  - Digital : comes with ADC

Real Time Application of Sensors

The example we are talking about here is the Autopilot System in aircrafts. Almost all civilian and military aircrafts have the feature of Automatic Flight Control system or sometimes called as Autopilot.



Types of Sensors

- **Temperature Sensor:**
  - One of the most common and most popular sensor is the Temperature Sensor. A Temperature Sensor, as the name suggests, **senses the temperature i.e. it measures the changes in the temperature.**
  - They are used everywhere like **computers, mobile phones, automobiles, air conditioning systems, refrigerator, industries etc**.
  - Eg: Thermocouple, thermistors, semiconductor temperature sensors, resistance temperature detectors (RTDs), and so on

- **Humidity Sensors:**
  - These types of sensors **measure the amount of water vapour in the atmosphere of air or other gases**.
  - Humidity sensors are **commonly found in heating, vents and air conditioning (HVAC) systems** in both industrial and residential domains.
  - They can be **found in many other areas including hospitals, and meteorology stations** to report and predict weather.

- **Motion Sensor:**
  - Is used to **detect the physical movement (motion) in a given area and it transforms motion into an electric signal**; motion of any object or motion of human beings.
  - Plays an **important role in the security industry**. Businesses **utilize these sensors in areas where no movement should be detected at all times**, and it is easy to notice anybody's presence with these sensors installed.

- **Gas Sensors:**
  - These types of sensors **monitor and detect changes in air quality, including the presence of toxic, combustible or hazardous gasses.**
  - Industries using gas sensors include mining, oil and gas, chemical research and manufacturing.
  - A common consumer use is the familiar carbon dioxide detectors used in many homes.
- **Following are some common Gas sensors:**
  - Carbon dioxide sensor
  - Carbon monoxide detector
  - Hydrogen sensor
  - Air pollution sensor
  - Nitrogen oxide sensor
  - Oxygen sensor
  - Ozone monitor

- **Smoke sensor**
  - A smoke sensor is a device **that senses smoke (airborne particulates & gases), and it's level**.
  - Smoke sensors are extensively used by manufacturing industry, buildings and accommodation infra to detect fire and gas incidences.
  - This serves to **protect people working in dangerous environments**, as the whole system is much more effective in comparison to the older ones.

- **Pressure sensor:**
  - A pressure sensor **is a device that senses pressure and converts it into an electric signal**. Here, the **amount depends upon the level of pressure applied.**
  - There are plenty of devices that **rely on liquid or other forms of pressure**.
  - Eg: Water systems and heating systems, as it is easy to detect any fluctuation or drops in pressure.

- **Image sensors**
  - Image sensors are instruments which are **used to convert optical images into electronic signals for displaying or storing files electronically**.
  - The major use of image sensor is found in digital camera & modules, medical imaging and night vision equipment,thermal imaging devices, radar, sonar, media house, Biometric & IRIS devices.

- **Accelerometers**
  - Accelerometers **detect an object's acceleration** i.e. the **rate of change of the object's velocity with respect to time**.
  - Accelerometers can **also detect changes to gravity**.
  - Used in other application to detect orientation of an object, shake, tap, position, shock or vibrations.

- **Infrared Sensors**
  - These types of sensors **sense characteristics in their surroundings by either emitting or detecting infrared radiation.**
  - They can **also measure the heat emitted by objects.**
  - Infrared sensors are used in a variety of different IoT projects including **healthcare** as **they simplify the monitoring of blood flow and blood pressure.**
  - **Televisions use infrared sensors to interpret the signals sent from a remote control**.
  - Another interesting application is that of **art historians using infrared sensors to see hidden layers in paintings to help determine whether a work of art is original or fake or has been altered by a restoration process.**

- **Proximity Sensors:**
  - **Detects the presence or absence of a nearby object**, **or properties of that object,** and converts it into signal which can be easily read by user or a simple electronic instrument **without getting in contact with them.**
  - **Eg:** Vehicles. You are reversing your car and are alarmed about an obstacle while taking reverse, that's the work of proximity sensor.

Actuator & its Types

- **Actuator:**
  - It is a **device which converts energy into motion.**
  - Actuators **convert an electrical signal into a corresponding physical quantity such as movement, force, sound etc.**
  - It alters the physical quantity as it can cause a mechanical component to move after getting some input from the sensor.
  - In other words, **it receives control input (generally in the form of the electrical signal) and generates a change in the physical system through producing force, heat, motion, etcetera.**
  - **Example: opening and closing dampers, locking doors etc.**

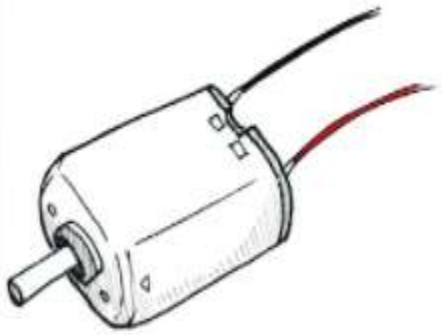
- **Servo Motor:**
  - **A Servo is a small device that incorporates a two wire DC motor, a gear train, a potentiometer, an integrated circuit, and a shaft (output spine).**
  - The shaft can be positioned to specific angular positions by sending the servo a coded signal.
  - Of the three wires that stick out from the servo casing, one is for **power**, one is for **ground**, and one is a **control input line**.
  - **When a control signal is applied to a Servo that represents a desired output position of the servo shaft, it (servo) applies power to its DC motor until its shaft turns to that position.**
  - It uses the position-sensing device to determine the rotational position of the shaft, so it knows which way the motor must turn to move the shaft to the commanded position.

- **Stepper Motor:**
  - Stepper motors are **DC motors that move in discrete steps**.
  - Each time a **pulse given** in the input accordingly **motor rotates in a predefined amount.**
  - A stepper motor is **suitable** for the applications where the **position of the object has to be controlled precisely**, for example, **robotic arm**.
  - They have **multiple coils** that are organized in groups called "**phases**".

- By **energizing each phase** in sequence, **the motor will rotate, one step at a time**.
- With a computer controlled stepping, you can achieve very precise positioning and/or speed control.
- A **servomotor** consumes power as it rotates to the commanded position but then the servomotor **rests**.
- **Stepper motors continue** to consume power to lock in and hold the commanded position.

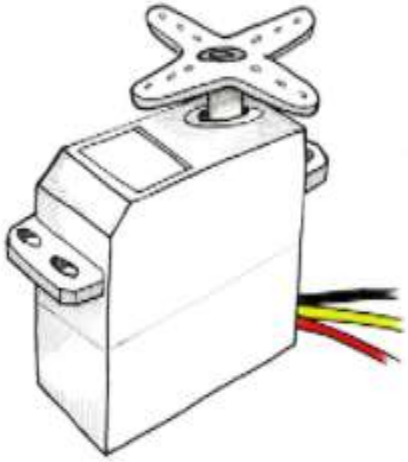
- **DC Motors:**
  - **DC Motor is a two wire continuous rotation motor** and the two wires are **power and ground**.
  - When the **supply is applied**, a DC motor will **start rotating** until that power is **detached**.
  - Most of the DC motors **run at high revolutions per minute (RPM)**, examples are**; fans** being used in **computers** for cooling etc.
  - DC motors **convert electrical into mechanical energy**.
  - They consist of **permanent magnets and loops of wire inside. When current is applied, the wire loops generate a magnetic field, which reacts against the outside field of the static magnets**.

DC Motor



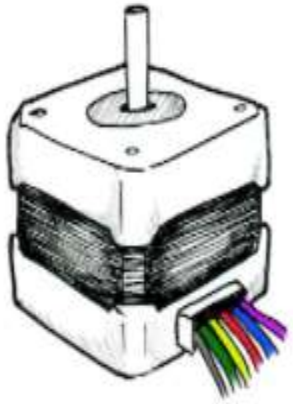
2 Wires

Servo Motor



3 Wires

Stepper Motor



Lots of wires

- **Linear Actuator:**
  - A linear actuator is an **actuator that creates motion in a straight line,** in contrast to the circular motion of a conventional electric motor.
  - Linear actuators are used in machine tools and industrial machinery, in computer peripherals such as **disk drives and printers, in dampers**, and in many other places where linear motion is required.

- **Relay:**
  - **Relay** is an switch, **which opens and closes the circuit electronically.**
  - It uses **electromagnetism** from **small voltage to provide higher voltages.** It has two **basic contacts** i.e. **NO (Normally Open) and NC (Normally Closed).**
  - When a relay is in **Normally Open (NO)** contact, there is actually **an open circuit** until the relay is energized.
  - If a relay is in **Normally Close (NC) contact**, there is a **closed circuit** until a relay is energized.
  - If **we apply current** to the relay contact in any of the above cases (NO, NC), **they will change their states** i.e. NC will become NO and vice versa.

- **Solenoid:**
  - A solenoid is simply a specially designed **electromagnet.** Solenoids are inexpensive, and their use is primarily limited to on-off applications such as latching, locking, and triggering.
  - They are frequently used **in home appliances (e.g. washing machine valves), office equipment (e.g. copy machines), automobiles (e.g. door latches and the starter solenoid) and factory automation.**