Introduction and History of The Internet of Things (IoT)

4th stage/ المرحلة الرابعة

Dr. Marwan Al-Dabbagh

Dr. Zena Al-Irhayim

2024-2025

* What is IoT

- The Internet of Things (IoT) refers to a network of physical devices that are connected to the internet and can collect, exchange, and act on data. These devices, often equipped with sensors, software, and other technologies, can communicate with each other and with central systems, enabling automation and improved efficiency across various applications.

- IoT, is a network of interrelated devices that connect and exchange data with other IoT devices and the cloud.

- IoT refers to a wireless network between objects, usually the network will be wireless and self-configuring, such as household appliances.

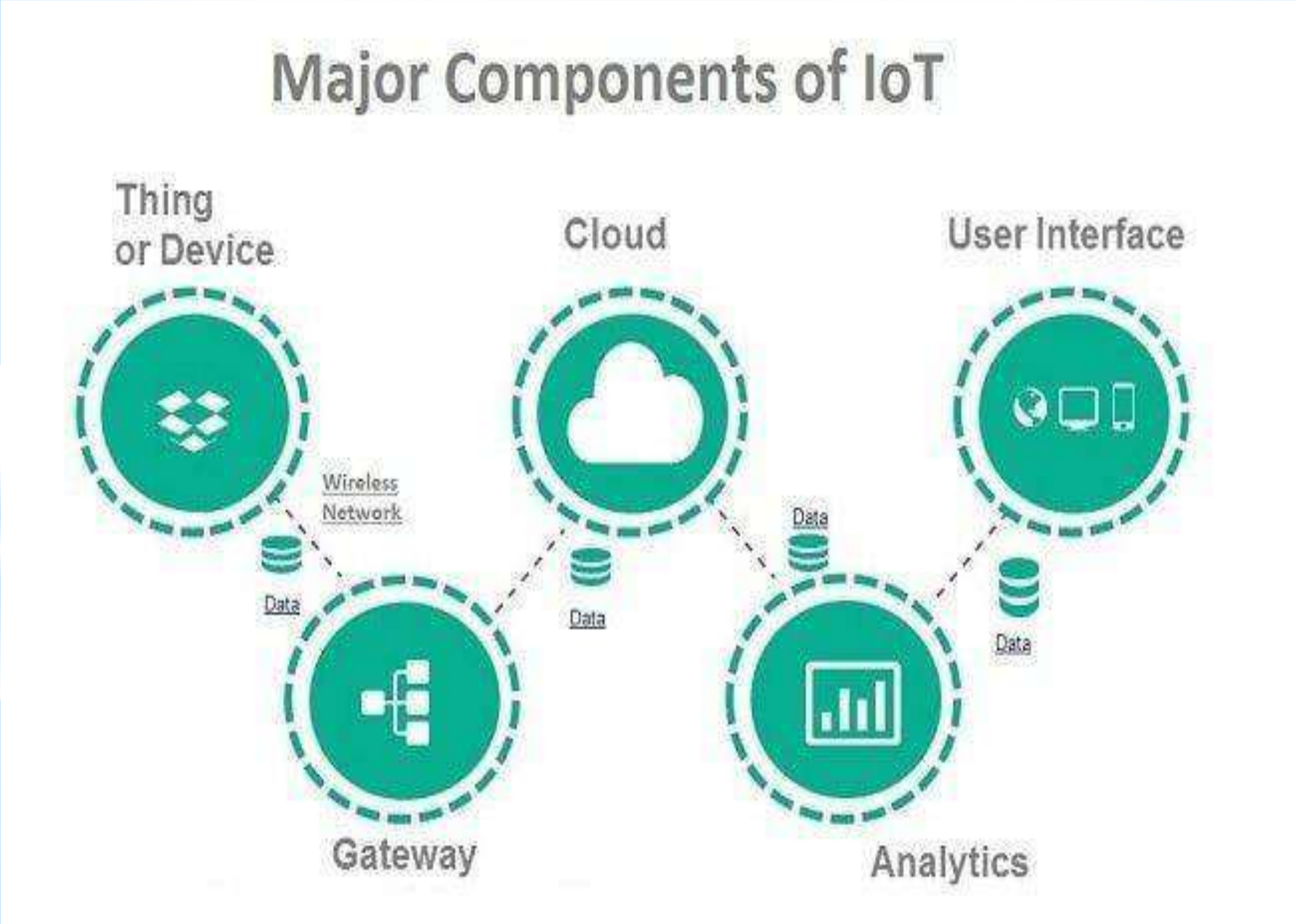
* Goal IoT

The main goal of IoT is to configure, control and network the devices or things, to internet (which are traditionally not associated with the internet), enabling them to collect, exchange, and analyze data autonomously or with minimal human intervention (e.g. thermostats, utility meters, a Bluetooth connected headset, control circuits for an electric car's engine, industrial and agriculture control, and many other domain smarter).

This interconnected network aims to enhance efficiency, convenience, and decision-making in various domains by creating smart environments.

* History of IoT

- 1999- The term “Internet of Things” was used by Kevin Ashton during his work at P&G which became widely accepted
- 2004- The term was mentioned in famous publications like the Guardian, Boston Globe
- 2005- International Telecommunications Union (ITU) published its first report on this topic
- 2008- the Internet of Things was born
- 2011- the market research company, include the Internet of Things technology in their research


Major Components of IoT
Thing or Device
Cloud
User Interface
Wireless Network
Data
Data
Data
Data
Gateway
Analytics

* Main Components of IoT

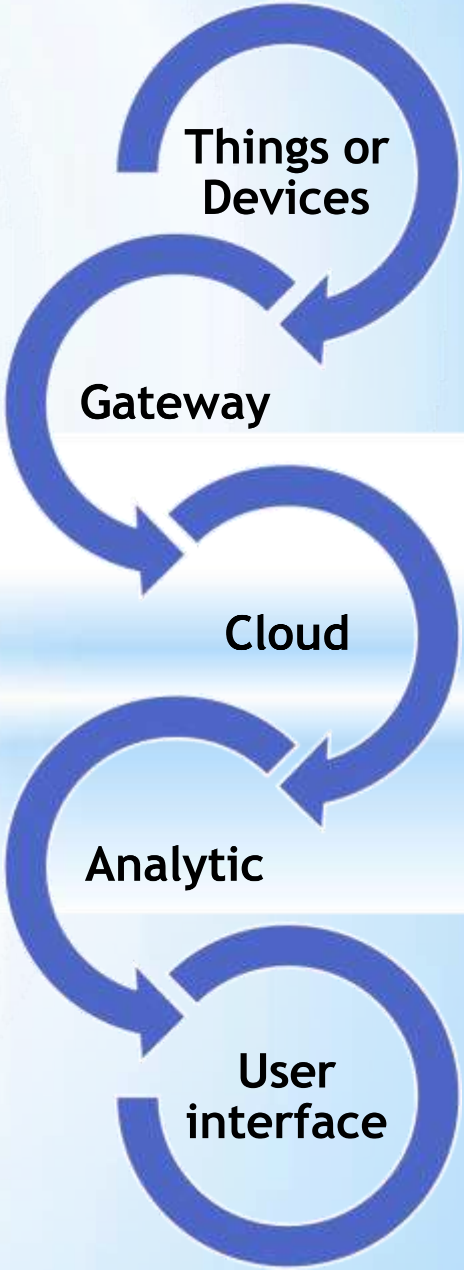
Things or Device: These are fitted with sensors. Sensors collect data from the environment and give to gateway.

Gateway: The sensors give data to Gateway and here some kind of pre-processing of data is even done. It also acts as a level of security for the network and for the transmitted data.

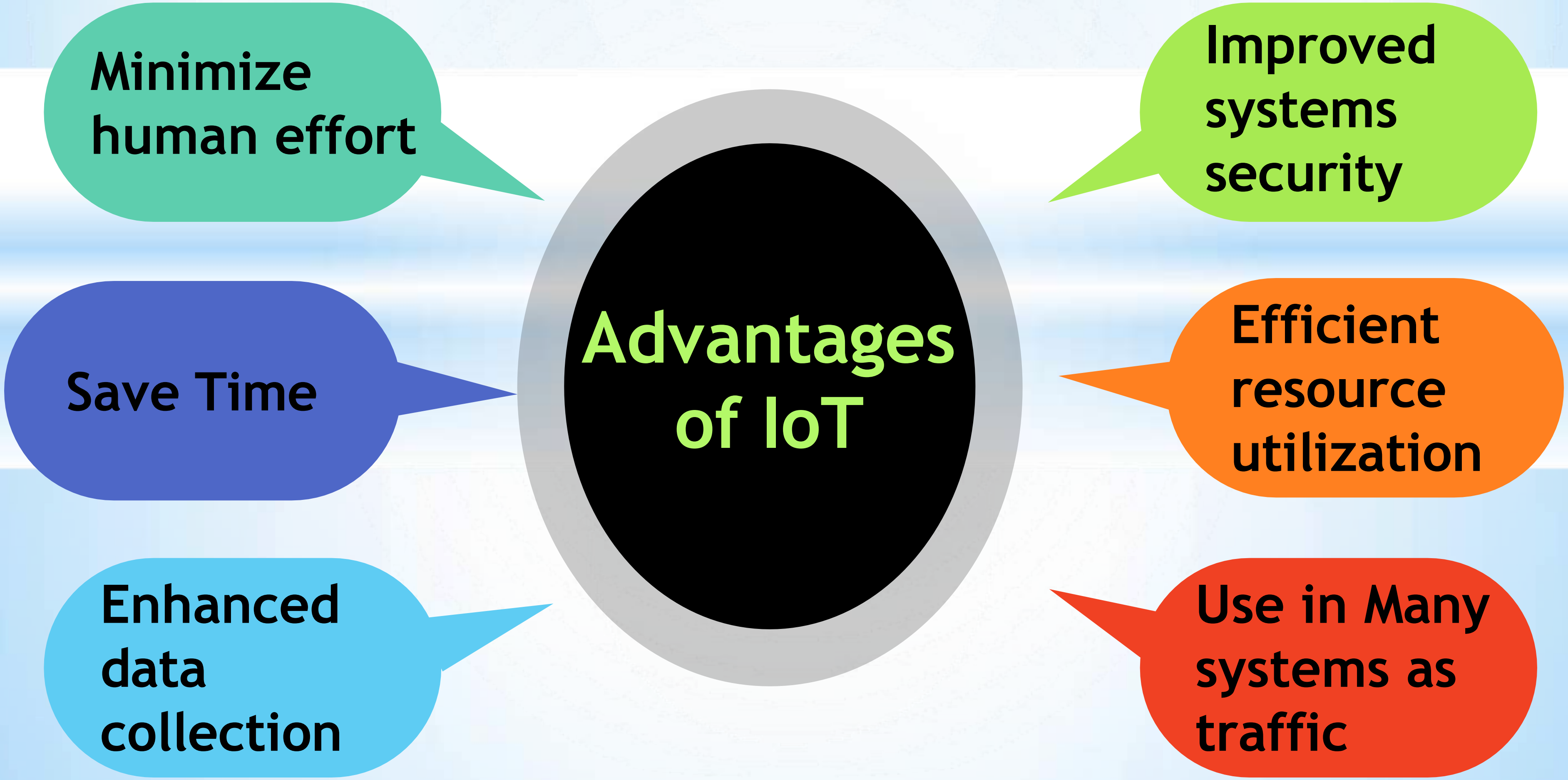
Cloud: The data after being collected is uploaded to cloud. Cloud in simple terms is basically a set of servers connected to internet 24*7.

Analytics: The data after being received in the cloud processing is done . Various algorithms are applied here for proper analysis of data.

User Interface: User end application where user can monitor or control the data.



* Advantages of IoT



* Advantages of IoT

Minimize human effort: As IoT devices interact and communicate with each other, they can automate the tasks helping to improve the quality of a business's services and reducing the need for human intervention.

Save time: By reducing the human effort, it saves a lot of our time. Saving time is one of the primary advantages of using the IoT platform.

Enhanced data collection: Information is easily accessible, even if we are far away from our actual location, and it is updated frequently in real-time. Hence these devices can access information from anywhere at any time on any device.

* Advantages of IoT

Improved systems security: If we have an interconnected system, it can assist in the smarter control of homes and cities through mobile phones. It enhances security and offers personal protection.

Efficient resource utilization: We can increase resource utilization and monitor natural resources by knowing the functionality and how each device works.

Use in traffic systems: Asset tracking, delivery, surveillance, traffic or transportation tracking, inventory control, individual order tracking, and customer management can be more cost-effective with the right tracking system using IoT technology.

* Disadvantages of IoT

Disadvantages of IoT

Security issues

The complexity of the system

Privacy concern

High dependency on the internet

Increased unemployment

Reduced mental & physical activity

* Disadvantages of IoT

Security issues: IoT systems are interconnected and communicate over networks. So, the system offers little control despite any security measures, and it can lead to various kinds of network attacks.

Privacy concern: The IoT system provides critical personal data in full detail without the user's active participation.

Increased unemployment: Unskilled workers or even the skilled ones are at a high risk of losing their jobs, leading to high unemployment rates. Smart surveillance cameras, robots, smart washing machines, and other facilities are replacing the humans who would earlier do these works.

* Disadvantages of IoT

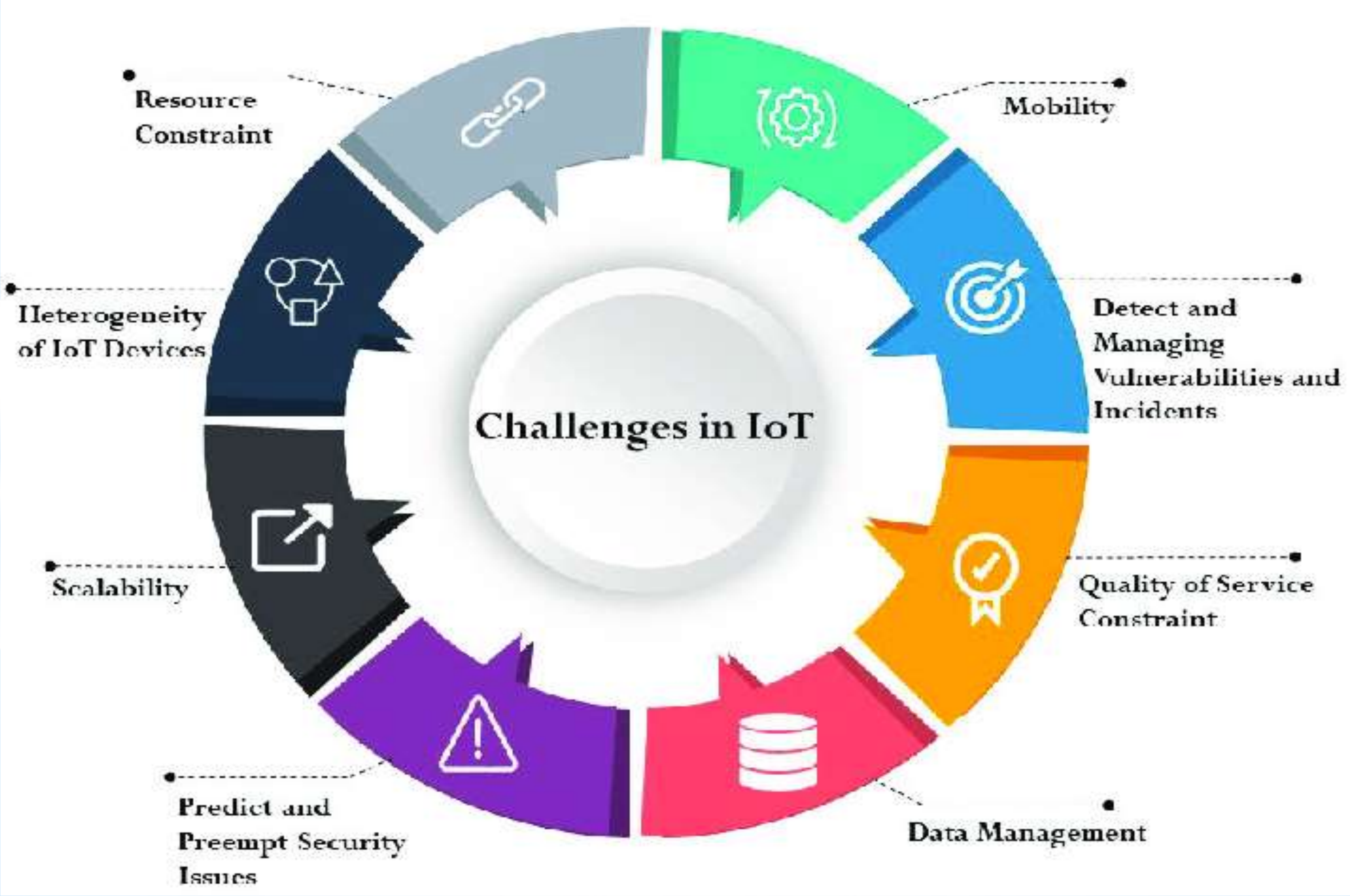
The complexity of the system: The designing, developing, maintaining, and enabling the extensive technology to IoT system is quite complicated.

High dependency on the internet: They rely heavily on the internet and cannot function effectively without it.

Reduced mental and physical activity: Overuse of the internet and technology makes people ignorant because they rely on smart devices instead of doing physical work, causing them to become lethargic and inactive.

* Applications of IoT

- ✓ Building and Home automation
- ✓ Manufacturing
- ✓ Medical and Healthcare systems
- ✓ Media
- ✓ Environmental monitoring
- ✓ Infrastructure management
- ✓ Energy management
- ✓ Transportation
- ✓ Better quality of life for elderly



Resource Constraint
Mobility
Heterogeneity of IoT Devices
Detect and Managing Vulnerabilities and Incidents
Challenges in IoT
Scalability
Quality of Service Constraint
Predict and Preempt Security Issues
Data Management

* Introduction to IoT and Sustainability

- The Internet of Things (IoT) is a network of interconnected devices that communicate and exchange data.

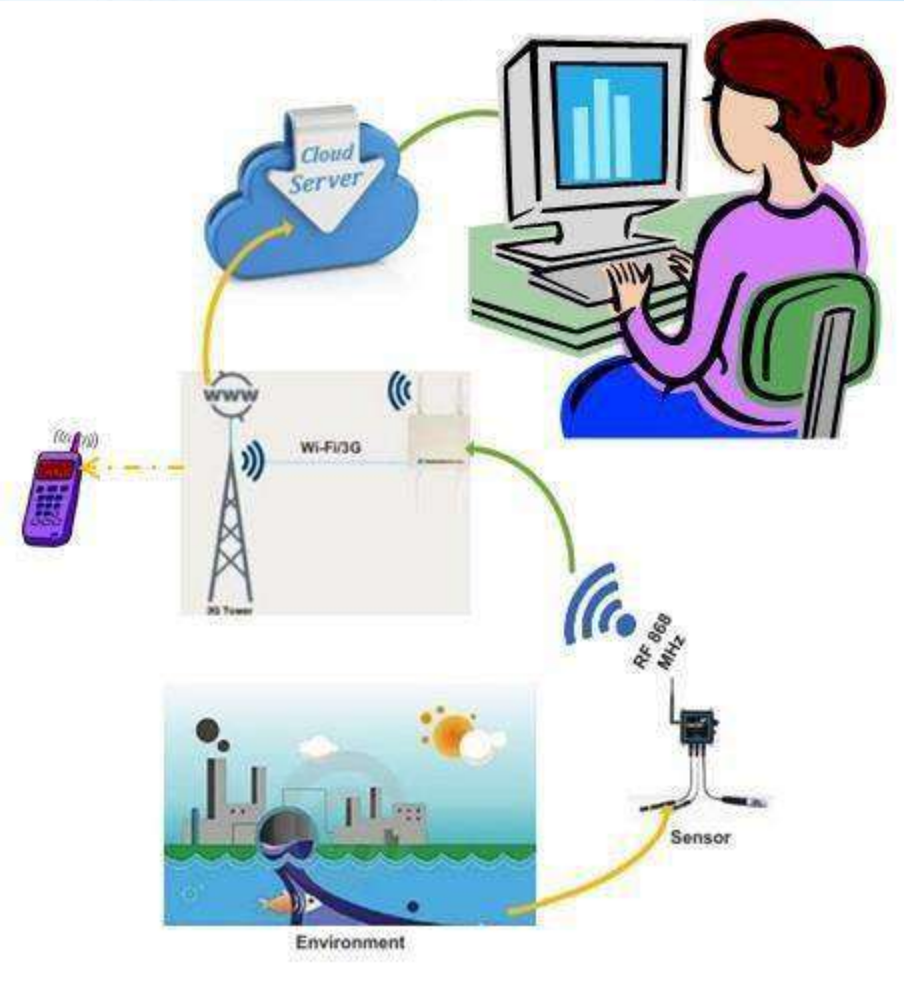
- Sustainability refers to the practice of meeting present needs without compromising the ability of future generations to meet their own needs. It involves balancing environmental, social, and economic considerations to ensure long-term well-being and resource availability.

- Sustainability goals seek to balance economic development and environmental protection.

* The Role of IoT in Sustainability

5. Smart Water Monitoring

- **Smart Water Management:** IoT devices monitor water levels, quality, and distribution. Cities can use this data to reduce leaks in water pipelines and optimize irrigation systems.
- **Agricultural Irrigation Control:** IoT systems can analyze soil moisture and weather data to optimize irrigation, minimizing water waste while ensuring crops are adequately watered.



6. Smart Agriculture

* **Precision Farming:** IoT sensors track soil conditions, weather, and crop health, helping farmers optimize the use of water, fertilizers, and pesticides. This minimizes chemical runoff and soil depletion.
* **Livestock Monitoring:** IoT wearables for livestock monitor health and behavior, reducing the need for medical interventions and improving the overall health of herds sustainably.

7. Cold-chain Management

- Nearly, one-third of the total food produced globally is wasted. This means that 1.6 billion tons of food is dumped every year which translates into the loss of $1.2 trillion. With food, all the energy and resources that it takes to grow, harvest and transport is also wasted and this is also an environmental concern that should be taken into account. The reason behind this huge loss is the inappropriate recording of temperatures in the food supply chain.

What is a GSM

GSM (Global System for Mobile communication) is a digital mobile network that is widely used by mobile phone users in Europe and other parts of the world. It is communication at a distance through electrical signals or electromagnetic waves. It includes mechanical and electrical communication.

10th March, 1876 *– The day when Alexander Graham Bell made the first successful transmission of speech in the form of electromagnetic waves using a custom built transmitter and receiver.*

Zero Generation(0G)

O G - Mobile radio telephone (also known as "oG"). oG refers to pre-cellphone mobile telephony. Being the predecessors of the first generation of cellular telephones, these systems are called oG (zero generation) systems. Mobile telephones systems were always usually vehicle mounted in the vehicle boot/trunk.

First Generation (1G)

There never was something called as 1G at first. The Mobile Wireless communication was possible only by the introduction of 1G Technology. It basically was a network with only voice call capabilities and only got the name 1G after 2G was put to use.

Drawbacks of 1G:

Poor voice quality
Poor battery life
Large phone size
No security
Limited capacity
Poor reliability

Second Generation (2G)

2G or second generation of wireless technology was first launched commercially in Finland in 1991. It utilized digital signals for voice transmission.

Drawbacks of 2G:

- ❖ Required strong digital signals to help mobile phones work. If there was no network coverage, digital signals weakened.
- ❖ Systems were unable to handle complex data like videos.

Third Generation (3G)

3G mobile telecommunication was first introduced in May 2001. The speed of 3G is about 4 times quicker than the old 2G standards.

Drawbacks of 3G:

- Power consumption is high
- Require closer base station and are expensive
- A 3G cell phone is more costly compared to 2G cell

Fourth Generation (4G)

4G is the 4th generation of broadband mobile communication technology, which was released in 2008. On December 14, 2009, 4G was launched commercially in the world. It provides a more reliable and fastest connection and higher speeds to mobile Internet experience-10 times faster than 3G.

Drawbacks of 4G:

- The battery uses is more
- Hard to implement
- Power consumption is high
- Require closer base station and are expensive

Fifth Generation (5G)

5G is the 5th generation wireless communication technology for digital cellular networks. It is expected that 5G networks can be smarter, faster and more efficient than 4G networks.
5G technology may use a variety of spectrum bands and can provide mobile data speeds up to 100 gigabits per second.

Drawbacks of 5G:

- Limited Coverage.
- Cyber security Risk.
- Battery Damages.

IoT Design Methodology Steps

1. **Purpose & Requirements**
   Define Purpose & Requirements of IoT system
2. **Process Model Specification**
   Define the use cases
3. **Domain Model Specification**
   Define Physical Entities, Virtual Entities, Devices, Resources and Services in the IoT system
4. **Information Model Specification**
   Define the structure (e.g. relations, attributes) of all the information in the IoT system
5. **Service Specifications**
   Map Process and Information Model to services and define service specifications
6. **IoT Level Specification**
   Define the IoT level for the system
7. **Functional View Specification**
   Map IoT Level to functional groups
8. **Operational View Specification**
   Define communication options, service hosting options, storage options, device options
9. **Device & Component Integration**
   Integrate devices, develop and integrate the components
10. **Application Development**
    Develop Applications

Step 1: Purpose & Requirements Specification

The first step in IoT system design methodology is to define the purpose and requirements of the system. In this step, the system purpose, behavior and requirements (such as data collection requirements, data analysis requirements, system management requirements, data privacy and security requirements, user interface requirements, ...) are captured.

Step 2: Process Specification

The second step in the IoT design methodology is to define the process specification. In this step, the use cases of the IoT system are formally described based on and derived from the purpose and requirement specifications.

Step 3: Domain Model Specification

The third step in the IoT design methodology is to define the Domain Model. The domain model describes the main concepts, entities and objects in the domain of IoT system to be designed. Domain model defines the attributes of the objects and relationships between objects. Domain model provides an abstract representation of the concepts, objects and entities in the IoT domain, independent of any specific technology or platform. With the domain model, the IoT system designers can get an understanding of the IoT domain for which the system is to be designed.

Step 4: Information Model Specification

The fourth step in the IoT design methodology is to define the Information Model. Information Model defines the structure of all the information in the IoT system, for example, attributes of Virtual Entities, relations, etc. Information model does not describe the specifics of how the information is represented or stored. To define the information model, we first list the Virtual Entities defined in the Domain Model. Information model adds more details to the Virtual Entities by defining their attributes and relations.

Step 5: Service Specifications

The fifth step in the IoT design methodology is to define the service specifications. Service specifications define the services in the IoT system, service types, service inputs/output, service endpoints, service schedules, service preconditions and service effects.

Step 6: IoT Level Specification

The sixth step in the IoT design methodology is to define the IoT level for the system.

Step 7: Functional View Specification

The seventh step in the IoT design methodology is to define the Functional View. The Functional View (FV) defines the functions of the IoT systems grouped into various Functional Groups (FGs). Each Functional Group either provides functionalities for interacting with instances of concepts defined in the Domain Model or provides information related to these concepts.

Step 8: Operational View Specification

The eighth step in the IoT design methodology is to define the Operational View Specifications. In this step, various options pertaining to the IoT system deployment and operation are defined, such as, service hosting options, storage options, device options, application hosting options, etc

Step 9: Device & Component Integration

The ninth step in the IoT design methodology is the integration of the devices and components.

Step 10: Application Development

The final step in the IoT design methodology is to develop the IoT application.

* Requirements of IoT

- Connectivity
- Sensors
- Intelligence

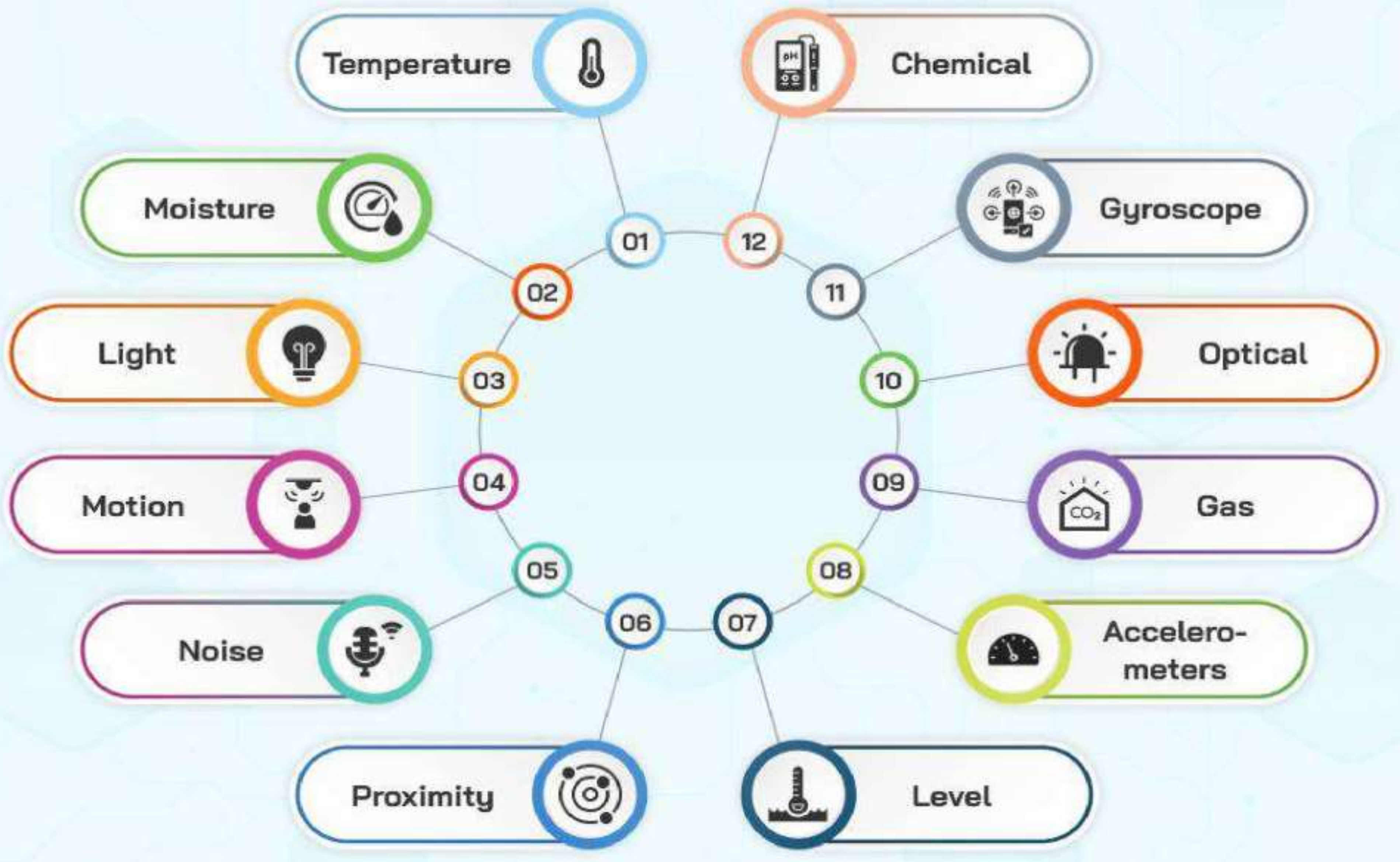
Connectivity

- IoT devices must be able to connect to the internet through a wired or wireless connection. Connectivity is essential for the Internet of Things (IoT), as it allows IoT devices to communicate with each other and with a central server or cloud platform.
- The IoT connectivity is defined as the type of technology that helps in providing the communication platform and network infrastructure used by several IoT devices for sending, and exchange the data.
- Various technologies and protocols can enable internet connectivity in IoT devices, including Wi-Fi, Bluetooth, and cellular networks.
- The choice of technology will depend on factors such as the device's location, power requirements, and data transfer needs.

Sensors

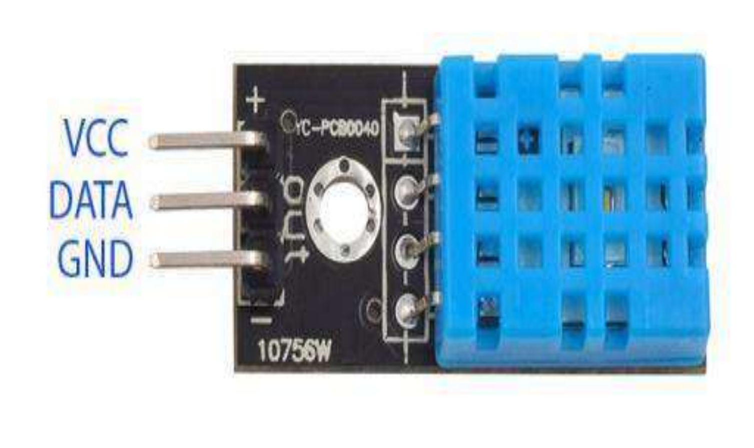
IoT devices must have sensors that can gather data from their environment, such as temperature, humidity, motion, or sound. This data is then used to trigger actions or provide user information. Sensors can be integrated into the device or connected through external means, such as a sensor module or a smart hub.

* Types of Sensors in IoT



* Temperature Sensor

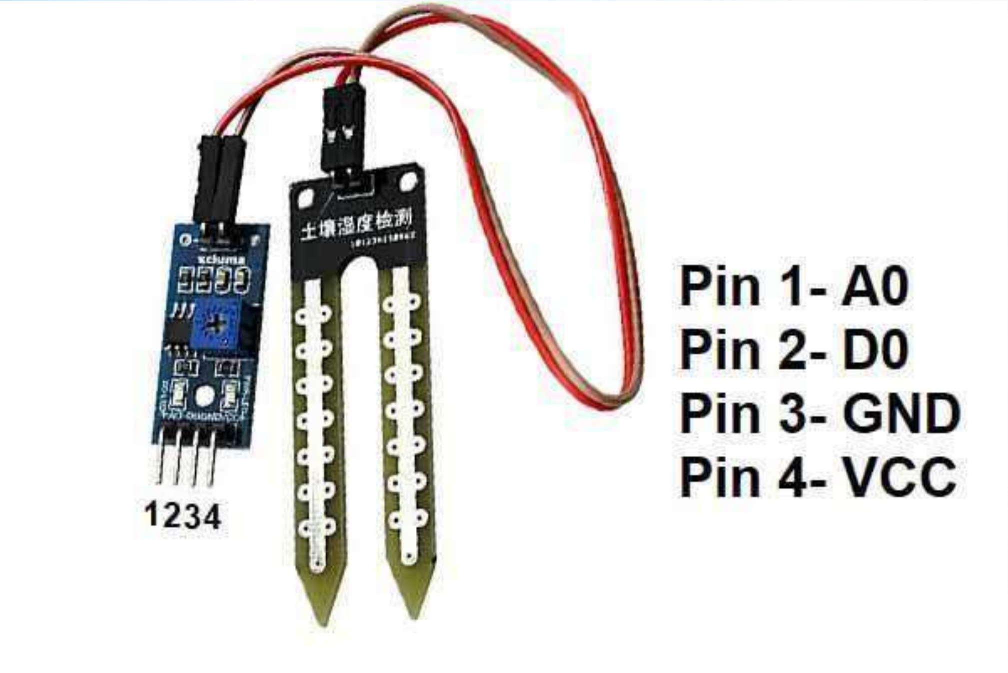
DHT11



The most common type of sensor is the temperate sensor, this sensor records the amount of heat in a given setting. It can be a machine, a room, a car, a lab, etc. This information can be used to take the desired action, like changing the temperature to optimal settings.

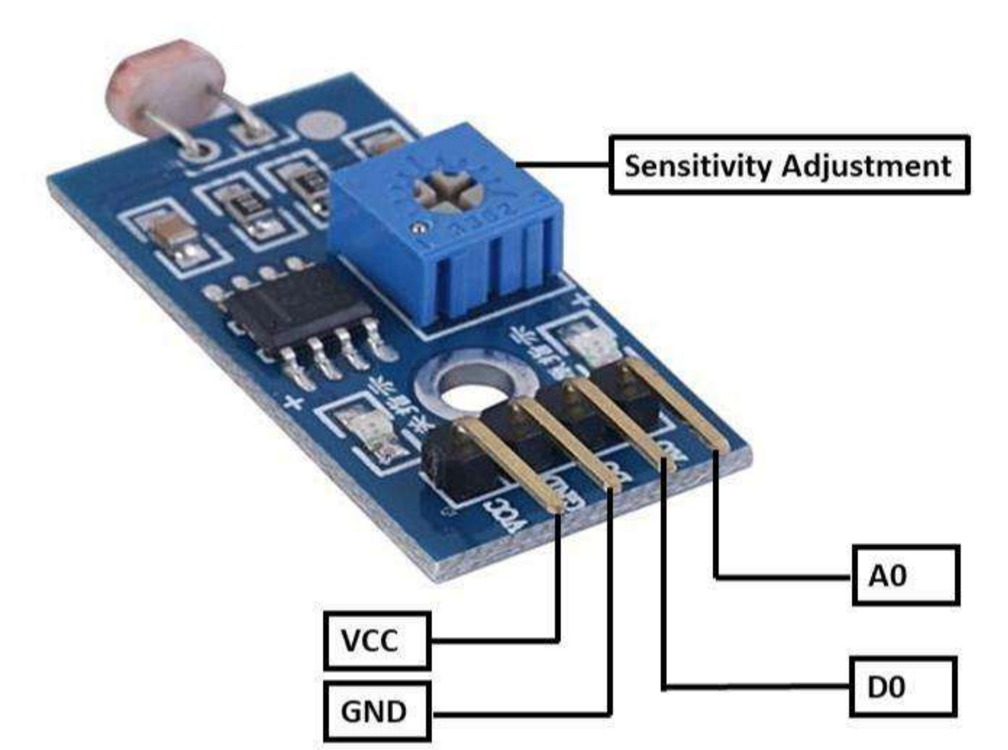
* Moisture Sensor

Soil Moisture



moisture sensors record the amount of humidity. They have a wide array of applications in the environment, medicinal labs, agriculture, etc.

* Light Sensor



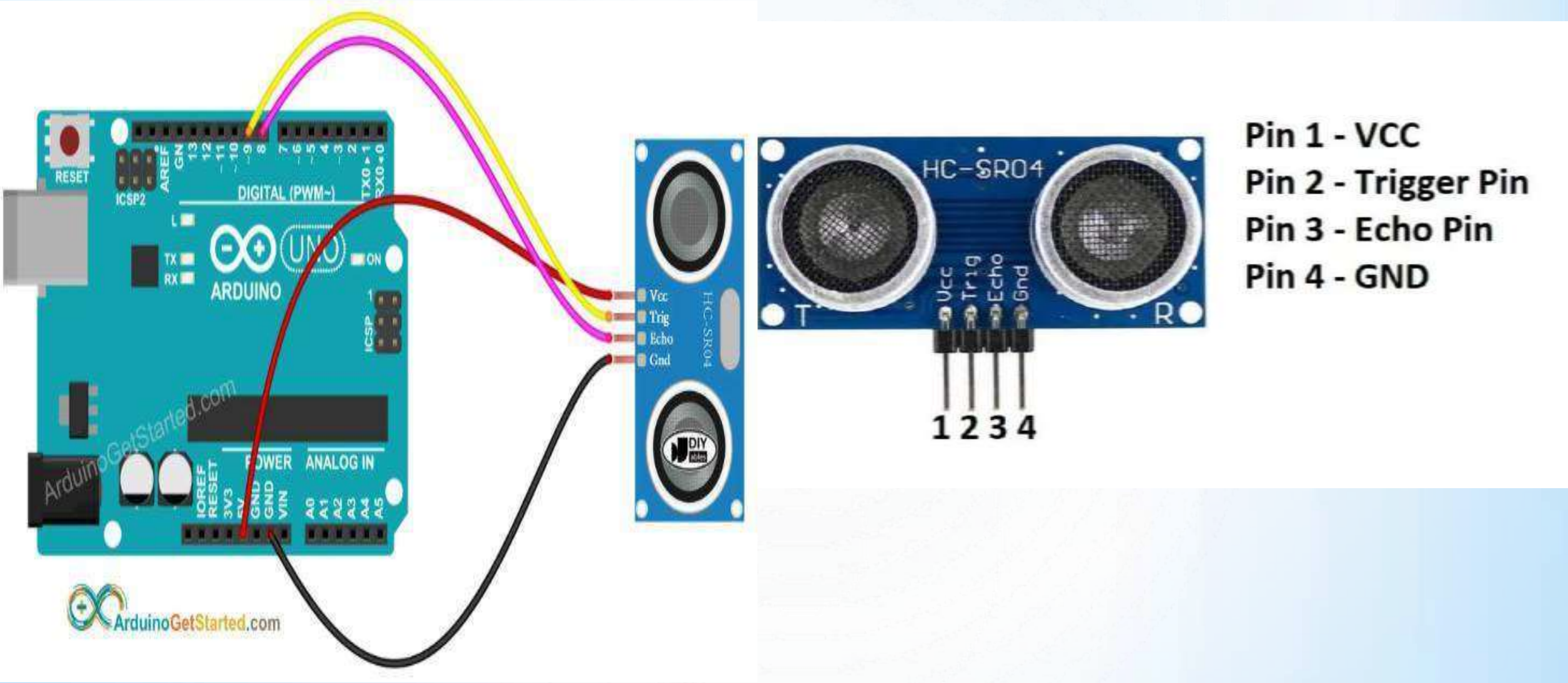
Light sensors record and assess the ambient light settings in a defined area and recommend actions to change the same.

* Motion Sensor

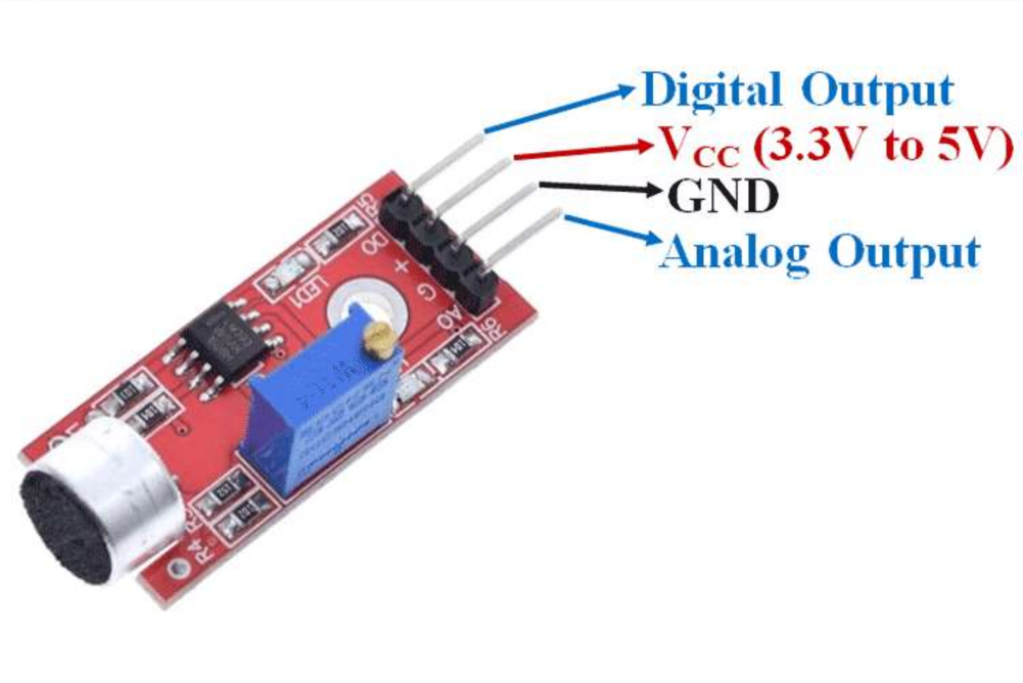
Ultrasonic

Motion sensors are usually installed in security systems and help detect unauthorized activity. Upon sensing activity either by changes in the heat or weight, the sensor activates an alarm system sending notifications to the right people. Motion IoT sensors use radar, infrared, or ultrasonic waves to detect activity in their vicinity.

* Motion Sensor



* Noise Sensor



Noise sensors, as the name suggests, record the noise levels in the given environment. It can be an entire city, a room, a car, etc. In IoT, these sensors are used to build safe working and living environments for people.

Intelligence

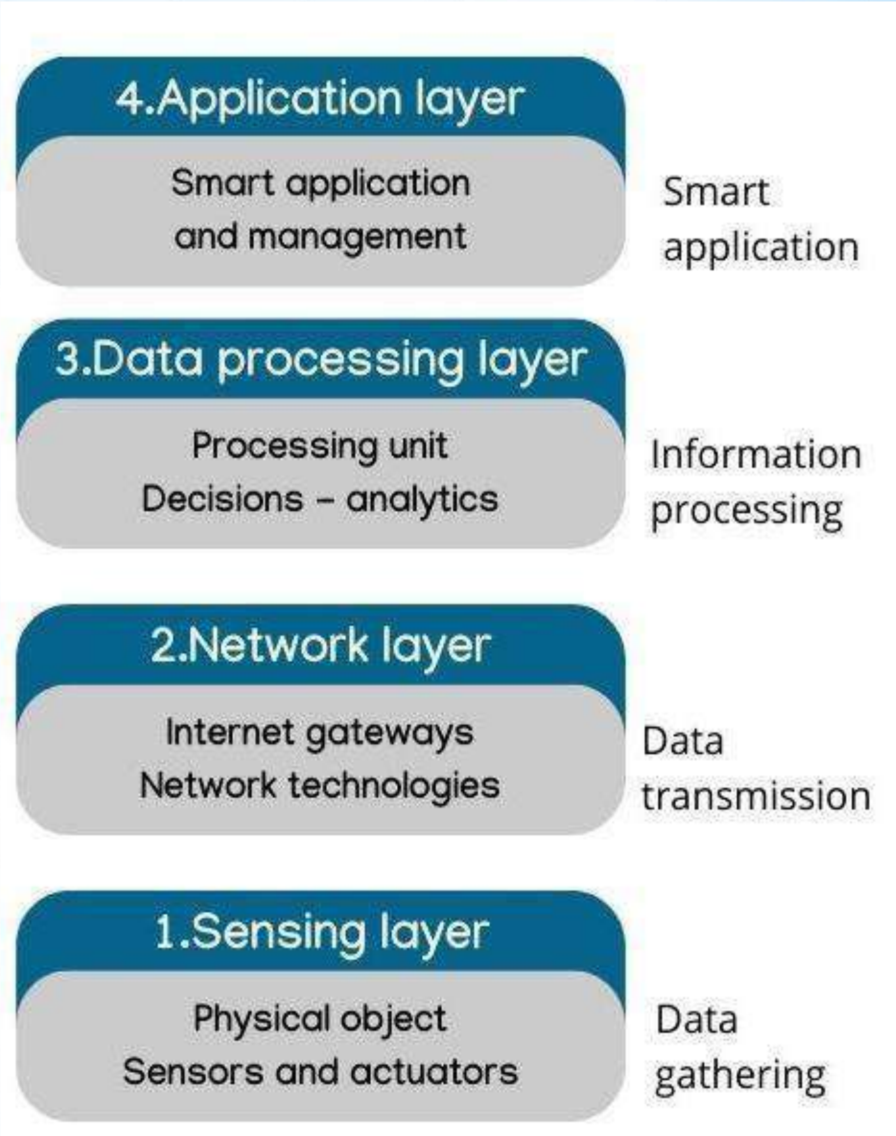
IoT devices must be able to process and analyze the data they collect and make decisions based on that data. This can be done through onboard software or by sending the data to a central server for processing. For example, a smart irrigation system may use data from weather sensors to determine the optimal amount of water, or a smart security camera may use face recognition to determine whether a person entering a home is an authorized user.

* IoT Architecture

The architecture of IoT depends upon its functionality and implementation in different sectors. it can be divided in 4 layers as follows:

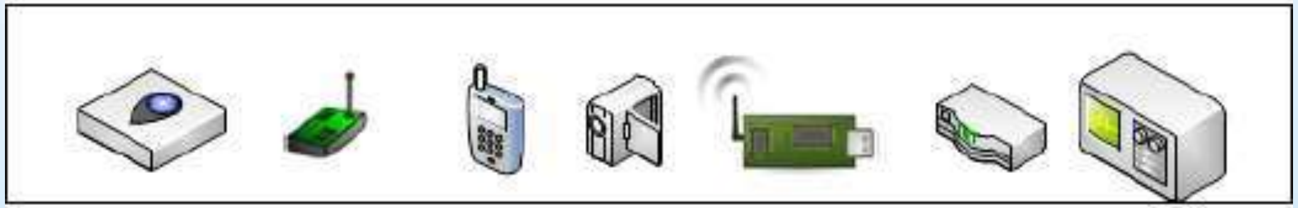
1. Sensing Layer.
2. Network Layer.
3. Data processing Layer.
4. Application Layer.

* OSI vs TCP/IP vs IoT



* 1. Sensing Layer

The sensing layer is the first layer of the IoT architecture and is responsible for collecting data from different sources. This layer includes *sensors* and *actuators* that are placed in the environment to gather information about temperature, humidity, light, sound, and other physical parameters. These devices are connected to the network layer through wired or wireless communication protocols.



* 2. Network Layer

Layer 2 Functions:

- Communications Between Layer 1 Devices
- Reliable Delivery of Information Across the Network
- Switching and Routing
- Translation Between Protocols
- Network Level Security

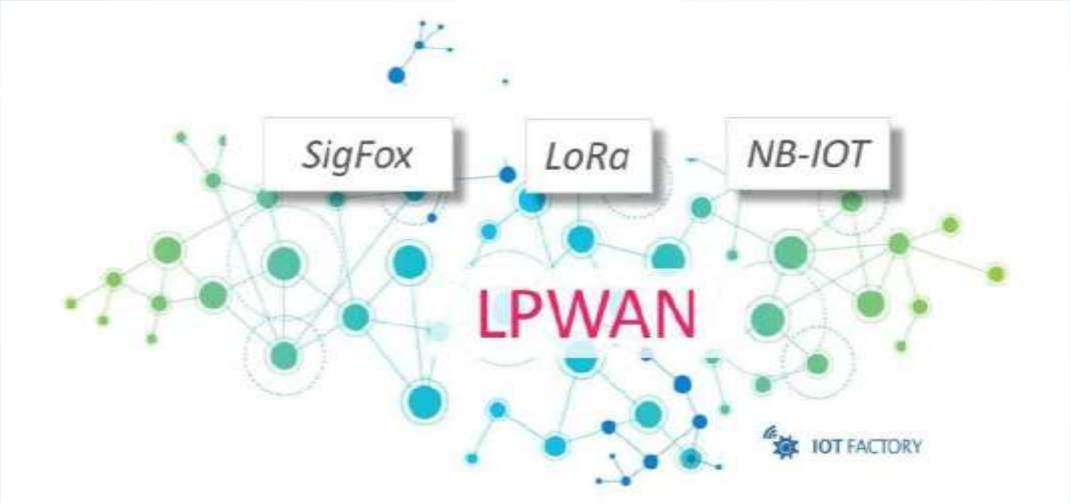
* 2. Network Layer

When you consider which networking technologies to adopt within your IoT application, be mindful of the following constraints:

1. Range
2. Bandwidth
3. Power usage
4. Intermittent connectivity
5. Interoperability
6. Security

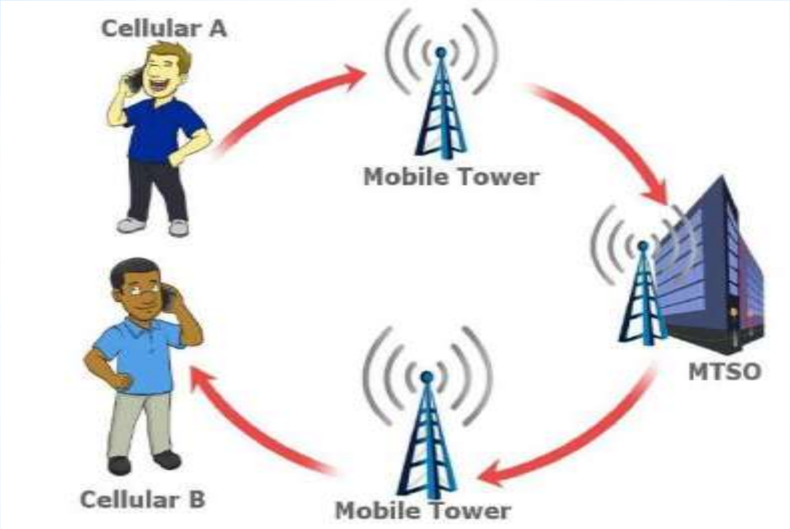
* 2. Network Layer

LPWAN (Low Power Wide Area Network) is a category of technologies designed for low-power, long-range wireless communication. They are ideal for large-scale deployments of low-power IoT devices such as wireless sensors. LPWAN technologies include LoRa (LongRange physical layer protocol), Haystack, SigFox, LTE-M, and NB-IoT (Narrow-Band IoT).



2. Network Layer

Cellular The LPWAN NB-IoT and LTE-M standards address low-power, low-cost IoT communication options using existing cellular networks. NB-IoT is the newest of these standards and is focused on long-range communication between large numbers of primarily indoor devices. LTE-M and NB-IoT were developed specifically for IoT, however existing cellular technologies are also frequently adopted for long-range wireless communication.



* 2. Network Layer

Bluetooth Low Energy (BLE)

BLE is a low-power version of the popular Bluetooth 2.4 GHz wireless communication protocol. It is designed for short-range (no more than 100 meters) communication, typically in a star configuration, with a single primary device that controls several secondary devices. Bluetooth operates across both layers 1 (PHY) and 2 (MAC) of the OSI model.

* 2. Network Layer

ZigBee operates on 2.4GHz wireless communication spectrum. It has a longer range than BLE by up to 100 meters. It also has a slightly lower data rate (250 kbps maximum compared to 270 kbps for BLE) than BLE. ZigBee is a mesh network protocol. ZigBee was designed for building and home automation applications.

Another closely related technology to ZigBee is Z-Wave, which is also based on IEEE 802.15.4. Z-Wave was designed for home automation.

* 2. Network Layer

NFC The near field communication (NFC) protocol is used for very small range communication (up to 4 cm), such as holding an NFC card or tag next to a reader. NFC is often used for payment systems, but also useful for check-in systems and smart labels in asset tracking.

* 2. Network Layer

RFID stands for Radio Frequency Identification. RFID tags store identifiers and data. The tags are attached to devices and read by an RFID reader. The typical range of RFID is less than a meter. RFID tags can be active, passive, or assisted passive. Passive tags are ideal for devices without batteries, as the ID is passively read by the reader. Active tags periodically broadcast their ID, while assisted passive tags become active when RFID reader is present.

* 2. Network Layer

Wifi is standard wireless networking based on IEEE 802.11a/b/g/n specifications. 802.11n offers the highest data throughput, but at the cost of high-power consumption, so IoT devices might only use 802.11b or g for power conservation reasons.

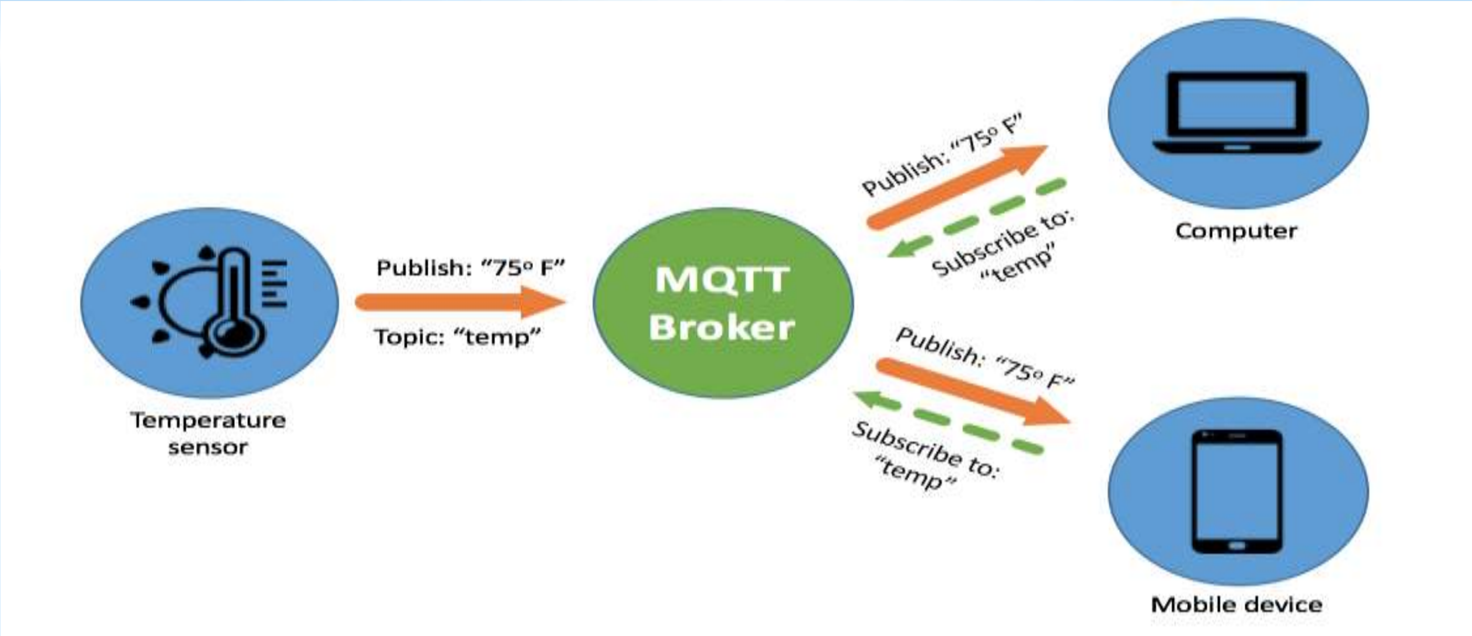
* 3. Data processing Layer

The data processing layer of IoT architecture refers to the software and hardware components that are responsible for collecting, analyzing, and interpreting data from IoT devices. This layer is responsible for receiving raw data from the devices, processing it, and making it available for further analysis or action. The data processing layer includes a variety of technologies and tools, such as data management systems, analytics platforms, and machine learning algorithms. These tools are used to extract meaningful insights from the data and make decisions based on that data. **Example** of a technology used in the data processing layer is a data lake, which is a centralized repository for storing raw data from IoT devices.

4. Application Layer

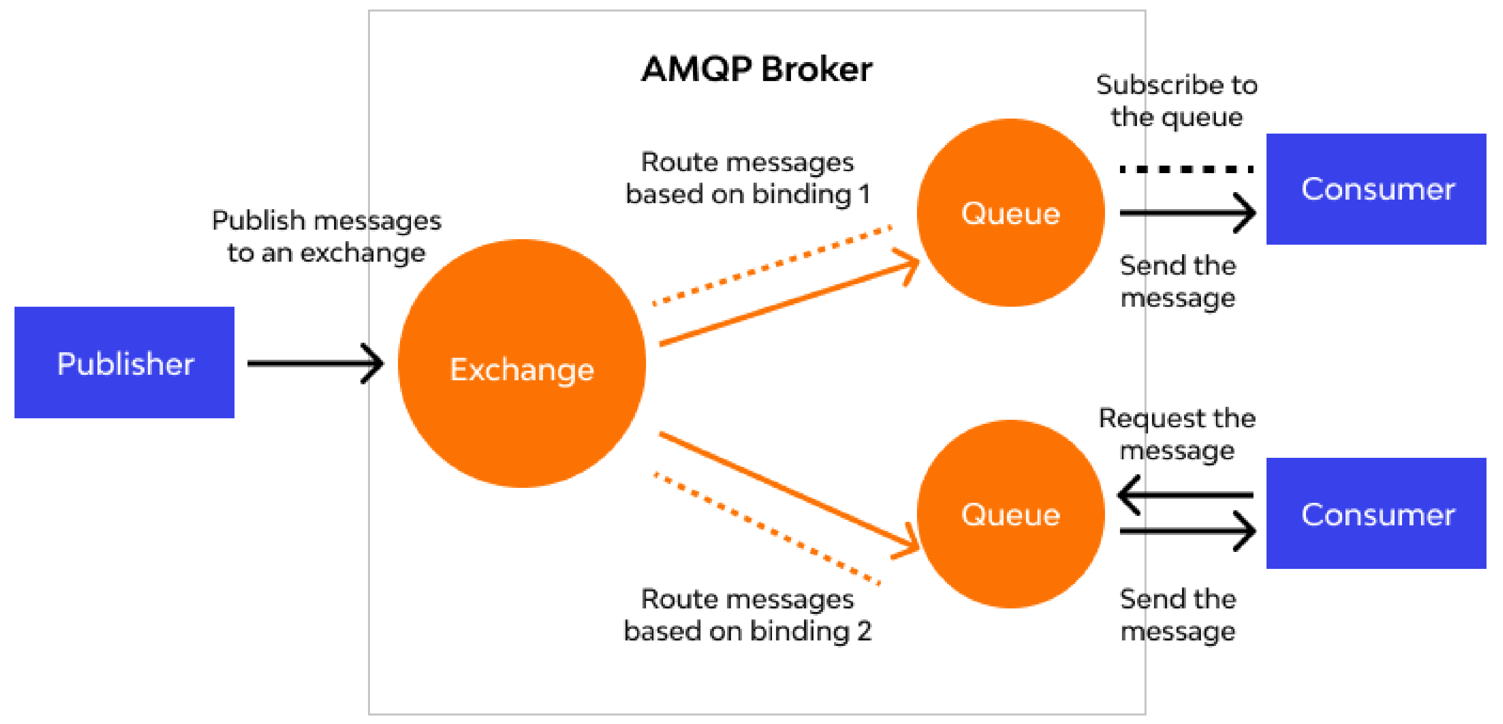
Messaging protocols used within IoT applications:

Message Queue Telemetry Transport (MQTT) is a publish/subscribe-based messaging protocol that was designed for use in low bandwidth situations, particularly for sensors and mobile devices on unreliable networks.



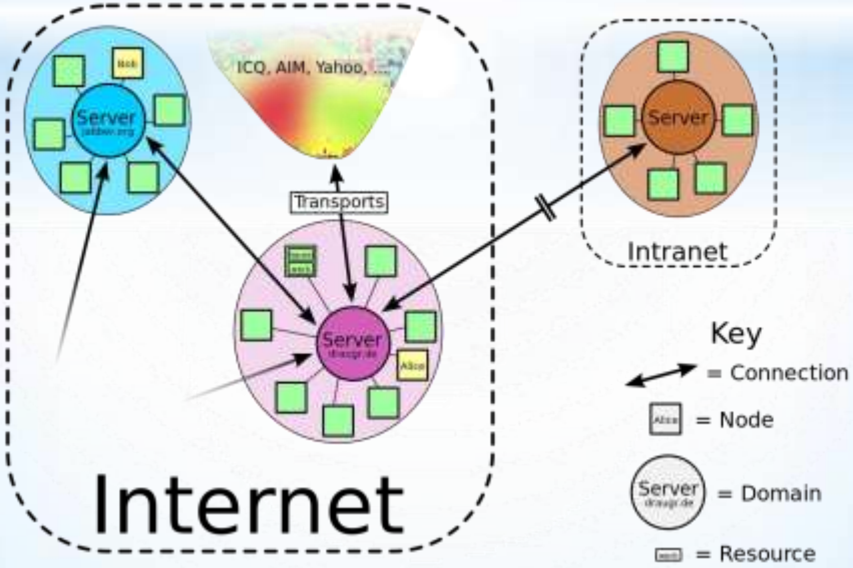
* 4. Application Layer

Advanced Message Queuing Protocol (AMQP) is an open standard messaging protocol that is used for message-oriented middleware. Most notably, AMQP is implemented by RabbitMQ.



4. Application Layer

The Extensible Messaging and Presence Protocol (XMPP) was originally designed for real-time human-to-human communication including instant messaging. This protocol has been adapted for machine-to-machine (M2M) communication to implement lightweight middleware and for routing XML data. XMPP is primarily used with smart appliances.



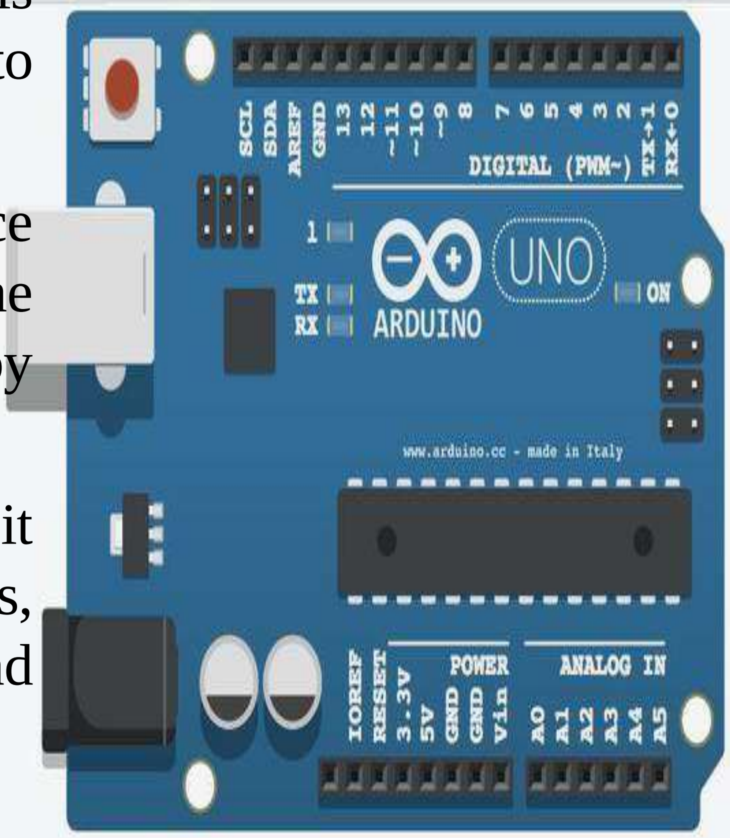
*IoT Platforms Overview:

The IoT concepts imply a creation of network of various devices interacting with each other and with their environment. Interoperability and connectivity wouldn’t be possible without platforms that help developers solve issues such as building autonomous interactive objects or completing common infrastructure related tasks.

An IoT platform is an application or service that provides built-in tools and capabilities to connect every “thing” in an IoT ecosystem. By providing functions including device management, device communication, data analytics, integration, and application enablement

What is Arduino?

- Arduino is an open-source electronics platform that includes both hardware (various boards, which can be programed (referred to as a microcontroller)) and software (Arduino IDE- Integrated Development Environment-which is used to write and upload the computer code to the physical board).
- Arduino board consists of an open source electronic chip with a microcontroller from the AMTL board on that is programmed by computer.
- It is ideal for building IoT projects because it allows users to easily connect sensors, actuators, and communication modules, and then program them to perform specific tasks.



* Arduino Key Features

1. **Open Source**: allowing anyone to study, modify, and create their own boards based on Arduino.
2. **Digital and Analog Pins** Support for **digital pins** to control devices like LEDs, motors, and relays. **Analog pins** for reading sensor inputs like temperature, light, and pressure.
3. **Built-in Microcontroller:** Each Arduino board comes with a microcontroller, like the ATmega328 on the Arduino Uno, which serves as the brain of the board.
4. **Libraries and Community Support**: **Extensive libraries** simplify the integration of various components like sensors, motors, and displays. While, **A vast and active community** offers tutorials, forums, and project ideas.
5. **Cost-Effective :** Arduino boards are relatively affordable, making them accessible for educational purposes and hobby projects.

* Arduino Key Features

6. **Expandability:** Compatible with plug-in boards that add functionality like motor control, GPS, or Ethernet connectivity.
7. **Cross-Platform Compatibility:** Arduino IDE supports multiple platforms, making it versatile for developers using different operating systems.
8. **Real-Time Interaction:** Arduino can interact with real-world components in real time, enabling projects like robotics, home automation, and wearable technology.

* Arduino Key Features

9. **Simple Programming:** The board functions are controlled by sending a set of instructions to the microcontroller on the board via Arduino IDE. Arduino is programmed in Arduino C language, which is a language derived from the C language, and it is considered one of the advanced programming languages.

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. Normally, all the apps are programmed in C/C++.

*Key Components

1. **Microcontroller**: The brain of the board that processes data and executes instructions. (e.g., ATmega328 on Arduino Uno).
2. **Power Supply**: Provides power to the board and connected components (e.g. Voltage regulators 5V or 3.3V).
3. **Digital Input/Output (I/O) Pins:** Used to control or read digital signals.
4. **Analog Input Pins:** Used for reading analog signals from sensors (e.g., temperature, light).And for Converting analog signals to digital values using an **Analog-to-Digital Converter (ADC).**
5. **Crystal Oscillator:** Provides a clock signal for the microcontroller to ensure stable operation and timing precision.

*Key Components

6. **Communication Interfaces:** Enable communication with external components like sensors, displays, and other microcontrollers.
7. **USB Port:** Used for programming the board and power supply, by establishing communication between the Arduino and a computer using the **Arduino IDE**.
8. **Memory:**
   - **Flash Memory**: Stores the program (non-volatile memory).
   - **SRAM**: Stores variables during execution (volatile memory).
   - **EEPROM**: Stores small amounts of non-volatile data for long-term use.

*Popular Boards for IoT

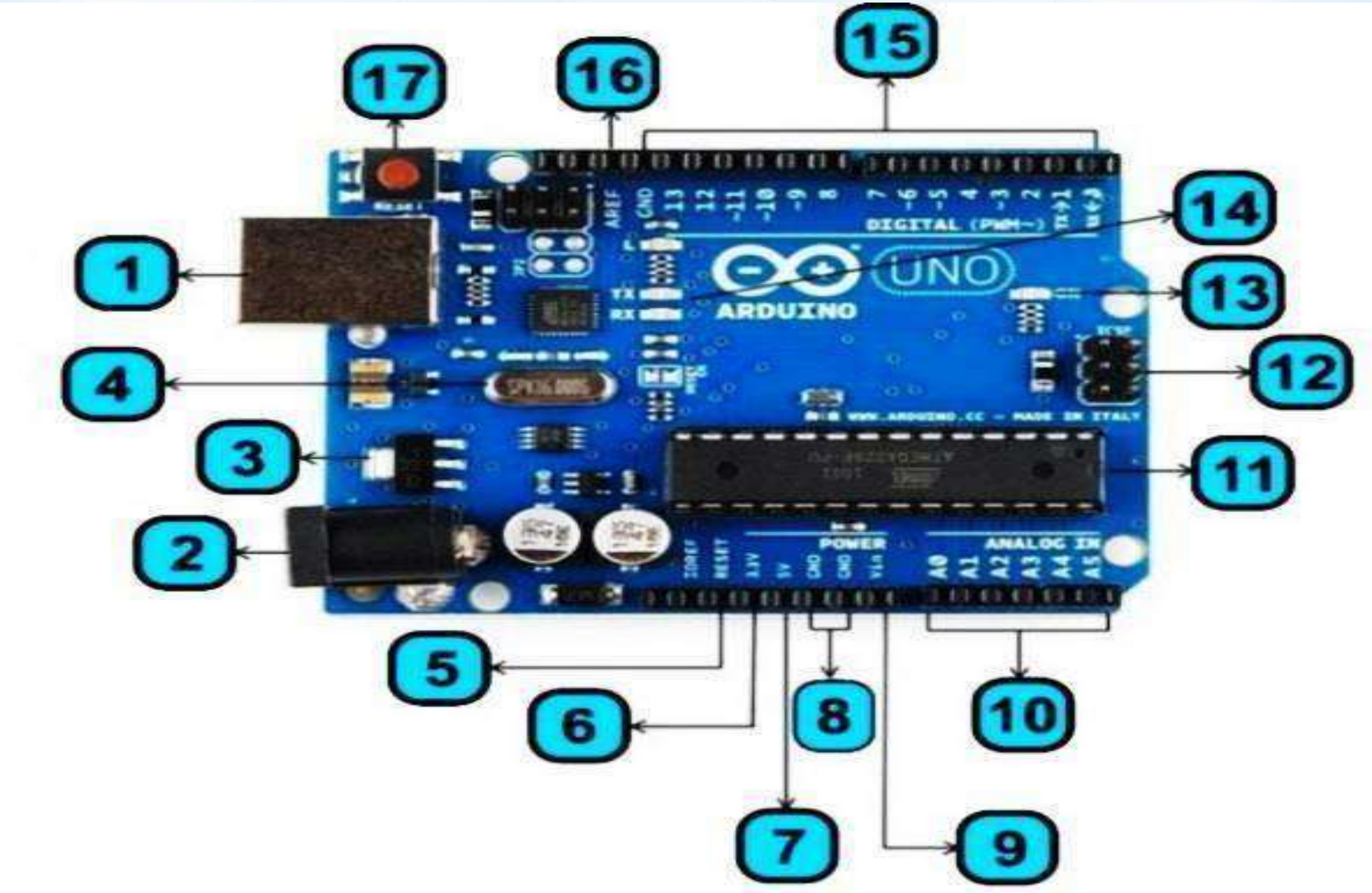
Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc.

- Arduino Uno
- Arduino Nano
- Arduino Mega
- Arduino MKR1000 (Wi-Fi enabled)
- Arduino Nano 33 IoT

We will study the Arduino UNO board because it is the most popular board in the Arduino board family.

* Arduino UNO



17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN
AREF
GND
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
TX
RX

Pins No.	Description
1)	Power USB : Arduino board can be powered by using the USB cable from your computer.
2)	Power (Barrel Jack): Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack
3)	Voltage Regulator: The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4)	Crystal Oscillator :The crystal oscillator helps Arduino in dealing with time issues.
5, 17)	Arduino Reset: You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labeled RESET
6, 7 8, 9	Pins (3.3V, 5V, GND, Vin) • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10)	Analog pins: The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor and convert it into a digital value that can be read by the microprocessor.

Pins No.	Description
11)	Main microcontroller: Each Arduino board has its own microcontroller (11). It implements the brain of your board. The microcontrollers are usually of the ATMEL Company.
12)	ICSP pin: The ICSP pin (In-Circuit Serial Programming pin) on an Arduino is a header used to program the microcontroller directly without requiring a bootloader or additional circuitry. consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. *
13)	Power LED indicator: This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14)	TX and RX LEDs : The TX LED (Transmit LED) Blinks when the Arduino is sending data to another device. RX (Receive LED) flashes during the receiving process.
15)	Digital I/O : The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc.
16)	AREF:AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

* ICSP pin (In-Circuit Serial Programming pin)

The **ICSP pin** (In-Circuit Serial Programming pin) allows for more advanced and flexible programming of the microcontroller, especially in cases where the standard USB programming method isn't suitable.

Purpose:

* Used to flash the microcontroller firmware directly.
* Facilitates low-level programming (e.g., burning the bootloader or firmware).

Pin Configuration: The ICSP header typically consists of 6 pins:

1. **MISO (Master In Slave Out)**: Used to send data from the Arduino to the programmer.
2. **MOSI (Master Out Slave In)**: Used to send data from the programmer to the Arduino.
3. **SCK (Serial Clock)**: Provides the clock signal for synchronization.
4. **RESET**: Resets the microcontroller to enable programming mode.
5. **VCC**: Provides power to the ICSP circuit.
6. **GND**: Ground connection.

* How Arduino Works in IoT

1. Sensors:

- ➢ Collect data from the environment (e.g., DHT11 for temperature and humidity).
- ➢ Send data to the Arduino board through analog or digital pins.

2. Processing:

- ➢ Arduino processes data using its microcontroller based on pre-written logic (sketches).
- ➢ Sketches are written in the Arduino IDE using a simplified version of C++.

3. Communication:

- ➢ Arduino sends processed data to IoT platforms or cloud services via communication modules.

4. Actuators:

- ➢ Based on the processed data or cloud instructions, Arduino controls actuators (e.g., turning on a fan when it’s hot).

* Challengages of Using Arduino in IoT

1. Limited computational power compared to other platforms like Raspberry Pi.
2. Limited storage capacity.
3. May require additional shields or modules for advanced connectivity.
4. Not ideal for large-scale or high-performance IoT applications.

* Raspberry Pi

* **Raspberry Pi:** is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom.
* It was initially developed for teaching computer science in schools.
* Later it gained popularity for robotics and weather monitoring due to its low cost, modularity, and open design.
* The Raspberry Pi hardware has evolved through several versions that feature variations in the type of the central processing unit, amount of memory capacity, networking support, and peripheral-device support.

* What is a Raspberry Pi?

* A credit-card-sized computer
* Runs on several operating systems
* Possible to connect with a variety of sensors to interact with the physical world.
* Make decisions based on processing of gathered sensor data
* It promotes Python and Scratch as the main programming language, with support for many other languages.

* Generations and Models

In 2012, the company launched the Raspberry Pi and the current generations of regular Raspberry Pi boards are Zero, 1, 2, 3, and 4. Generation 1 Raspberry Pi had the following four options:

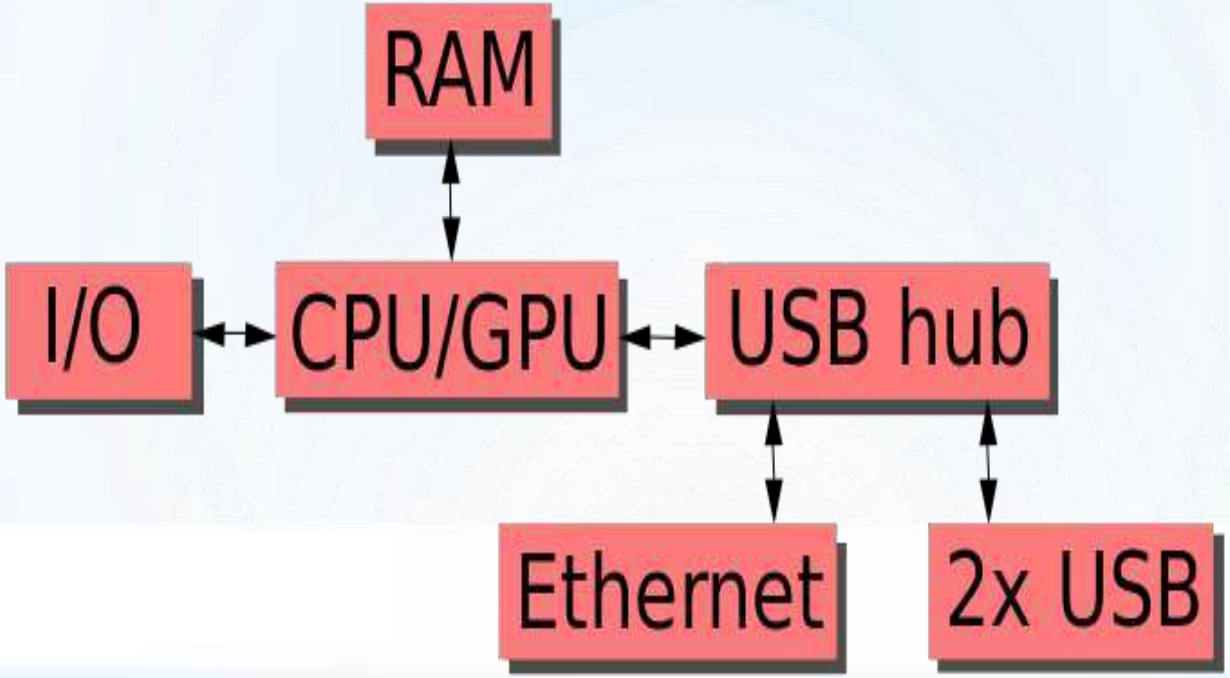
* Model A
* Model A +
* Model B
* Model B +

Among these models, the Raspberry Pi B models are the original credit-card sized format. On the other hand, the Raspberry Pi A models have a smaller and more compact footprint and hence, these models have the reduced connectivity options. Raspberry Pi Zero models, which come with or without GPIO (general-purpose input output) headers installed, are the most compact of all the Raspberry Pi boards types.

*Raspberry Pi Features

1. All models feature a Broadcom system on a chip (SoC), which includes a central processing unit (CPU) and an on-chip graphics
2. processing unit (GPU, a VideoCore IV).
3. CPU speed ranges from 700 MHz to 1.2 GHz for the Pi 3 and on board memory range from 256 MB to 1 GB RAM.
4. Secure Digital (SD) cards are used to store the operating system and program memory.
5. Most boards have between one and four USB slots, HDMI (High-Definition Multimedia Interface) and composite video output, and a 3.5 mm phono jack for audio.
6. The B-models have an 8P8C Ethernet port and the Pi 3 and Pi Zero W have on board Wi-Fi 802.11n and Bluetooth.

Raspberry Pi Basic Architecture



- This block diagram describes Model B and B+;
- Models A, A+, and the Pi Zero are similar, but lack the Ethernet and USB hub components.

*Raspberry Pi in IoT

The Raspberry Pi is a powerful and versatile tool for Internet of Things (IoT) applications due to its compact size, affordability, and robust capabilities. It acts as a hub, gateway, or edge device for collecting, processing, and transmitting data in IoT systems.

*Why Use Raspberry Pi in IoT?

1. Computing Power: Offers sufficient processing power for edge computing tasks, data analysis, and AI/ML workloads.

2. Connectivity Options:

- Built-in Ethernet, Wi-Fi, and Bluetooth on newer models like Raspberry Pi 4 and Raspberry Pi Zero W.

- USB and GPIO pins for interfacing with various sensors and actuators.

* Why Use Raspberry Pi in IoT?

3. Open-Source Ecosystem: Supported by a large community, extensive documentation, and compatible open-source software.

4. Programmability: Supports multiple programming languages (Python, C++, Java, etc.), making it accessible to developers.

5. Affordability: Low cost compared to other development boards and computing devices.

*Raspberry Pi's Roles in IoT?

Raspberry Pi's Role in	Description	Examples
Edge Device	Local data processing and decision-making	Real-time air quality monitoring
Gateway	Bridges IoT devices and the cloud	Smart home hubs
Prototyping Tool	Rapid development of IoT solutions	Testing automation systems
Data Logger	Local storage of sensor data	Environmental sensors in remote areas
Sensor Interface	Integration with physical devices	Soil moisture sensors in farms
Remote Monitoring	Enables remote device management	Smart thermostat control
AI/ML Deployment	Runs lightweight AI models for smart IoT systems	Object recognition in smart cameras
Cloud Connectivity	Sends and receives data to/from cloud platforms	IoT dashboards for industrial data

* Arduino or Raspberry Pi for Internet of Thing

Arduino finds application in systems that perform simple repetitive tasks that need only one action at a time. Often, Arduino will be a more cost-effective and time-consuming solution than Pi.

The Raspberry Pi, which is faster and more powerful than Arduino, can multitask and run more complex functions. It entails playing media, performing calculations, and collecting various parameters. Raspberry Pi is the right choice if an IoT system needs to:

* Collect data from multiple sensors
* Pull data from the Internet
* Connect to a smartphone, and,
* Display a complex output