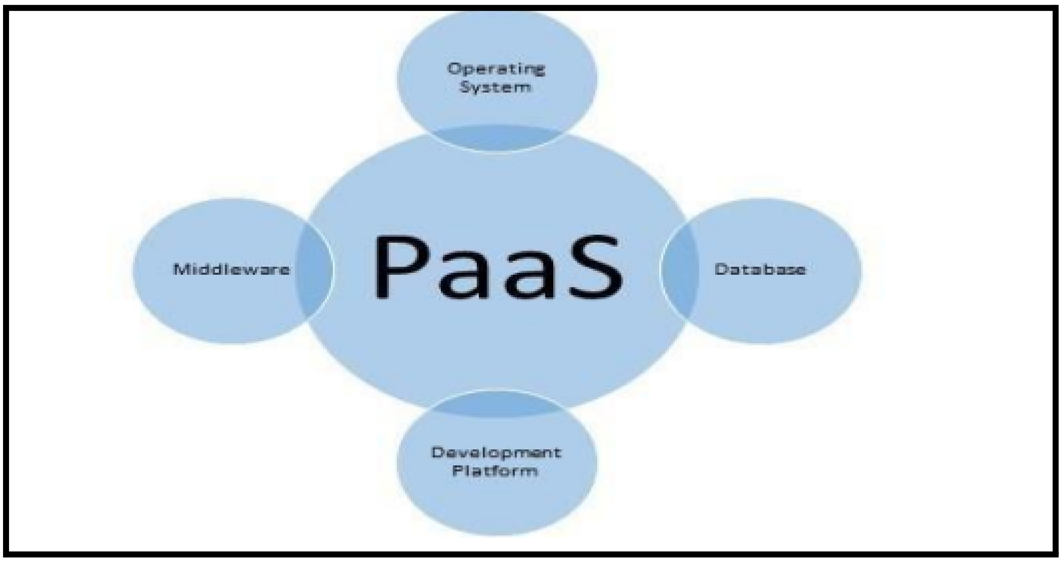
Iot Module - 4
DESIGN AND DEVELOPMENT OF AI ENABLED IOT APPLICATIONS

Cloud Computing

Cloud computing is a service provisioning technique where computing resources like hardware such as servers and storage devices, software's and complete platform for developing applications are provided as a service by the cloud providers to the customers

Cloud Service Models

Cloud services are categorised into three main models:

1. **Infrastructure-as-a-Service (IaaS)**:
   - **Description**: Provides virtualized hardware resources like servers, storage, and networking over the internet. The cloud provider manages the infrastructure, and clients pay based on usage.
   - **Examples**: Amazon EC2, Amazon S3.
2. **Platform-as-a-Service (PaaS)**:
   - **Description**: Offers a complete development and deployment environment. Provides infrastructure as well as tools for designing, developing, testing, deploying, and hosting applications. Removes the need for buying and managing hardware and software.
   - **Features**: Extends IaaS by also providing software and configurations for building applications.
   - **Examples**: Google App Engine.
   - 

3. **Software-as-a-Service (SaaS)**:
    - **Description**: Delivers software applications over the internet, accessible via a web browser. Users don't need to handle installation, maintenance, or updates.
    - **Examples**: Google Docs, Customer Relationship Management (CRM), Accounting software

Cloud Deployment Models

There are four main cloud deployment models, each catering to different needs and use cases:

1. **Public Cloud**:
    - **Description**: Services and infrastructure are hosted by a third-party provider and made accessible to the general public via the internet.
    - **Features**: Cost-effective and scalable, but with less control and security than private models.
    - **Example**: Google Cloud Platform, Microsoft Azure.
2. **Private Cloud**:
    - **Description**: Services and infrastructure are dedicated to a single organisation, either managed internally or by a third party, and accessed through a private network.
    - **Features**: Provides a higher level of security and control, ideal for sensitive data.
3. **Community Cloud**:
    - **Description**: Infrastructure is shared among a group of organisations with similar requirements or missions (e.g., healthcare, government).
    - **Features**: Offers collaborative benefits and cost-sharing among the organisations while providing security tailored to the community's needs.
4. **Hybrid Cloud**:
    - **Description**: Combines public and private clouds, allowing data and applications to be shared between them.
    - **Features**: Stores critical data in a private cloud for security, while less-sensitive applications and data reside in the public cloud, balancing flexibility, security, and cost-efficiency.

Features of Cloud Computing

1. **Elasticity**:
   - **Description**: Provides flexibility, allowing users to scale resources up or down based on their needs.
2. **Pay-per-Use**:
   - **Description**: Users are charged only for the resources they consume, making it cost-efficient.
3. **Managed Operations**:
   - **Description**: Cloud services are fully managed by the provider, relieving users from operational burdens.
4. **Reduced Capital Costs**:
   - **Description**: Eliminates the need for investing in hardware, software, licenses, and IT training, as resources are provided by the cloud.
5. **Remote Accessibility**:
   - **Description**: Data and applications can be accessed from any location with internet access, enabling global collaboration.
6. **Optimised IT Staff Utilisation**:
   - **Description**: Internal IT staff can focus on core tasks rather than managing infrastructure, as cloud providers handle maintenance and updates.

Cloud Services Examples: IaaS – Amazon EC2, Google Compute Engine, Azure VMs

1. **Amazon EC2 (Elastic Compute Cloud)**

- **Overview**:
  - Amazon EC2 is an IaaS (Infrastructure as a Service) offering from Amazon that provides scalable computing capacity in the cloud.
  - Users can launch virtual machine instances (VMs) on-demand using a simple web-based interface.
  - Amazon provides pre-configured **Amazon Machine Images (AMIs)**, which are templates of cloud instances. Users can also create their own AMIs with custom applications, libraries, and data.
- **Features**:
  - **Instance Types**: EC2 offers instances with high memory, high CPU resources, cluster compute instances, and high I/O instances.
  - **Operating Systems**: EC2 instances can be launched with a variety of operating systems.
  - **Scaling**: Users can load their applications on running instances and scale up or down depending on application performance requirements.
  - **Mass Provisioning**: EC2 allows users to provision hundreds or thousands of instances simultaneously.
- **Pricing**:
  - EC2 follows a **Pay-Per-Use** pricing model, where users are billed based on the number of instance hours used.
  - **Spot Instances**: Users can bid for unused EC2 capacity and run instances as long as their bid exceeds the spot price.

Amazon EC2 Instances

- **Instance Sizes**:
  - From **Small Instances** (1 virtual core, 1.7GB memory, 160GB instance storage) to **Extra Large Instances** (4 virtual cores with 2 EC2 compute units, 15GB memory, 1690 GB storage).
- **Key Features**:

    - **High Performance**: Instances for various use cases like high-performance computing, databases, and web servers.
    - **On-demand Scaling**: Users can easily scale up or down based on their needs.
    - **Resource Monitoring**: EC2 instances allow users to monitor resource usage and manage network access permissions.

---

2. Cloud Services Examples: IaaS – Google Compute Engine, Azure VMs

Google Compute Engine (GCE)

- **Overview**:
    - Google Compute Engine is an IaaS offering from Google.
    - GCE provides virtual machines (VMs) with scalable computing resources, ranging from small instances (e.g., 1 virtual core with 1.38 GCE unit and 1.7GB memory) to high-memory instances (e.g., 8 virtual cores with 22 GCE units and 5GB memory).
- **Features**:
    - **Custom VMs**: Offers the ability to configure instances based on requirements.
    - **Scaling**: GCE supports the scaling of resources to meet the growing demands of applications.
    - **Web Interface**: Users can easily manage VMs and deploy applications with a simple web interface.

Azure VMs

- **Overview**:
    - Azure Virtual Machines (VMs) are an IaaS offering from Microsoft Azure.
    - VMs provide virtual machines of various computing capacities, ranging from small instances (e.g., 1 virtual core with 1.75GB memory) to memory-intensive machine types (e.g., 8 virtual cores with 56GB memory).
- **Features**:

    - **Scalability**: Azure VMs are highly scalable, allowing users to adjust resources as required by their workloads.
    - **Managed by Microsoft**: Azure VMs are backed by Microsoft's cloud infrastructure, ensuring reliability and uptime.

---

3. Cloud Services Examples: PaaS – Google App Engine (GAE)

Google App Engine (GAE)

- **Overview**:
    - Google App Engine is a Platform as a Service (PaaS) offering from Google.
    - It allows developers to build scalable web applications and store data without managing the underlying infrastructure.
- **Features**:
    - **Automatic Scaling**: GAE automatically scales your applications based on demand.
    - **Languages Supported**: GAE supports various programming languages like **Java**, **Python**, **Go**, etc.
    - **Load Balancing**: Automatically balances the load to ensure performance during traffic spikes.
- **Development and Deployment**:
    - **GAE SDK**: Developers can use the GAE SDK to build and test applications on their local machine before uploading to the cloud.
    - **Easy Deployment**: Applications can be deployed to GAE with just one click.
- **Pricing**:
    - **Free Usage**: GAE offers free computing resources up to a certain limit.
    - **Pay-As-You-Go**: Once the limit is exceeded, users are charged based on resources used (bandwidth, storage, instance hours, etc.).

---

4. Cloud Services Examples: SaaS – Salesforce

Salesforce (SaaS)

- **Overview**:
  - Salesforce is a **cloud-based** Customer Relationship Management (CRM) platform that offers various tools for sales, service, and marketing.

Salesforce Sales Cloud

- **Overview**:
  - Sales Cloud is used to manage customer relationships, track sales opportunities, and optimize campaigns.
  - **Lead Management**: Helps manage leads (companies or individuals interested in the product) and optimize campaigns from lead generation to closure.

Salesforce Service Cloud

- **Overview**:
  - Service Cloud provides a customer service management platform with tools for creating, tracking, routing, and escalating customer service cases.
  - **Social Media Integration**: Service Cloud includes a social networking plug-in that helps address customer queries from social media channels.

Salesforce Marketing Cloud

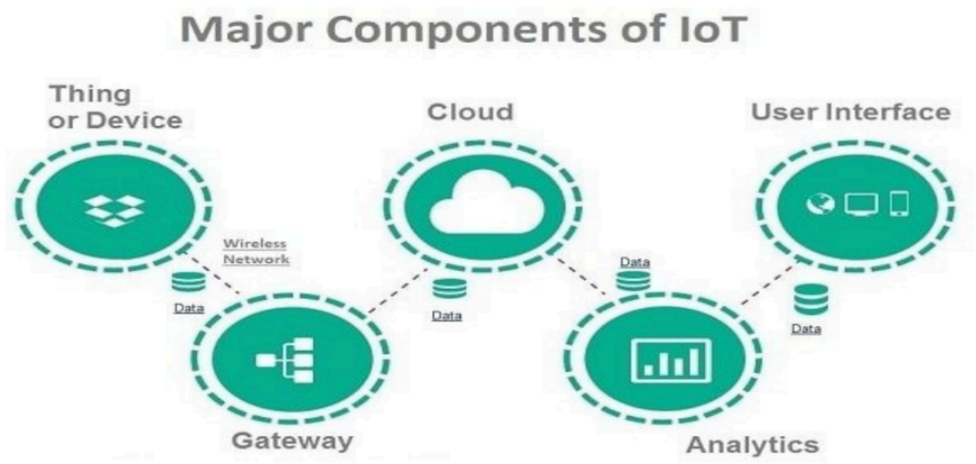
- **Overview**:
  - Marketing Cloud is a social marketing platform that helps companies engage customers through social media and track the performance of campaigns.
  - **Campaign Management**: Manages social media advertisement campaigns, identifies sales leads, and tracks the impact of social marketing activities.

5. Cloud Concepts and Technologies

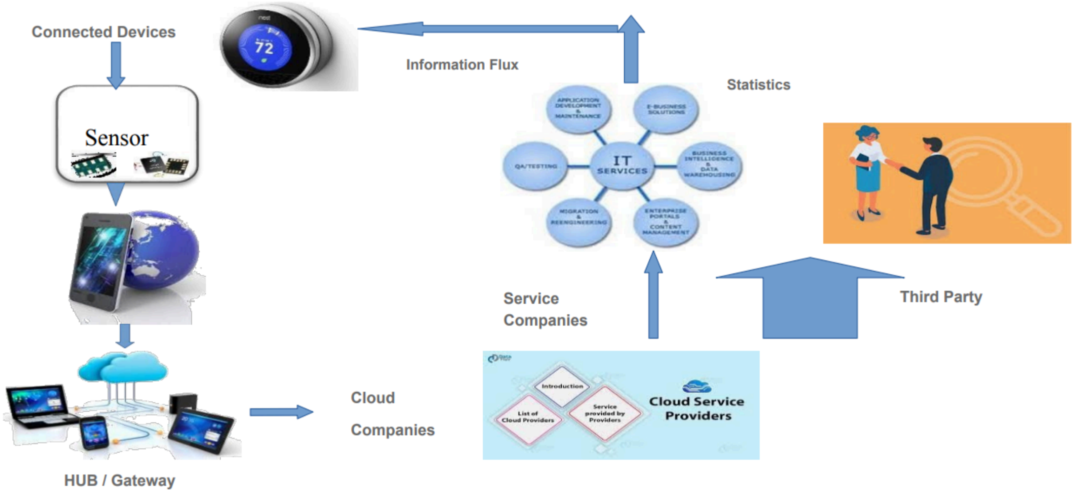
- **Virtualization**:
  - The process of creating virtual versions of physical resources like servers, storage, and networking.
- **Load Balancing**:
  - Distributes traffic evenly across multiple servers to ensure no single server is overwhelmed, improving performance and reliability.
- **Scalability and Elasticity**:
  - **Scalability**: The ability to increase/decrease resources as per demand.
  - **Elasticity**: The ability to automatically scale resources based on real-time demand.
- **Deployment**:
  - The process of launching an application or service into the cloud environment, ensuring it's ready for use.
- **Replication**:
  - Copying data across multiple locations to ensure redundancy and high availability.
- **Monitoring**:
  - Continuous tracking of resource usage, performance metrics, and system health to ensure optimal operation.
- **Software Defined Networking (SDN)**:
  - Virtualizes the network infrastructure, providing flexibility and better control over the network.
- **MapReduce**:
  - A framework for processing large datasets across distributed computing resources, commonly used in big data applications.
- **Identity and Access Management (IAM)**:
  - Controls who can access which resources in the cloud environment, ensuring security and compliance.
- **Service Level Agreements (SLAs)**:
  - Formal agreements between service providers and customers, outlining expected performance levels (e.g., uptime, support).
- **Billing**:

- Cloud providers usually offer usage-based billing, where users pay for the resources they consume, based on metrics such as compute hours, storage, and data transf

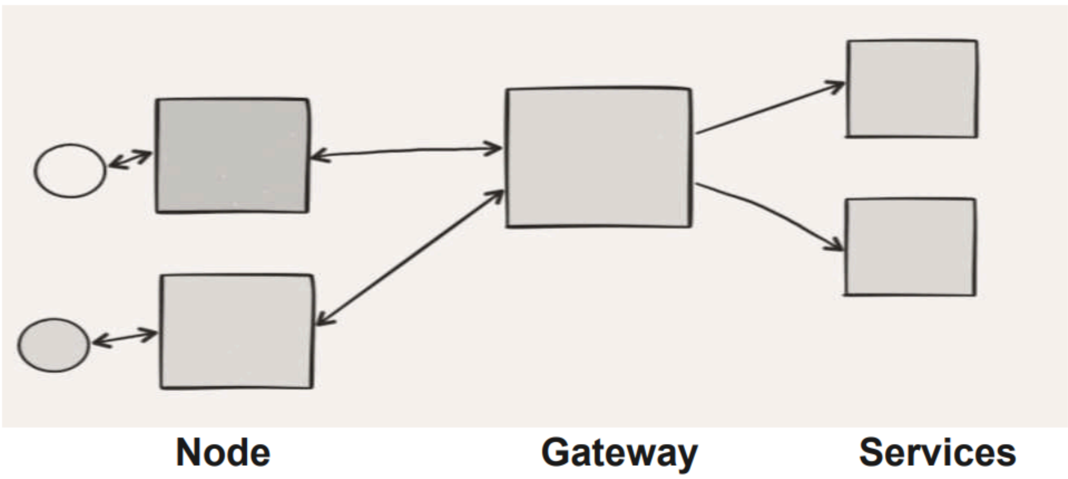
IoT System



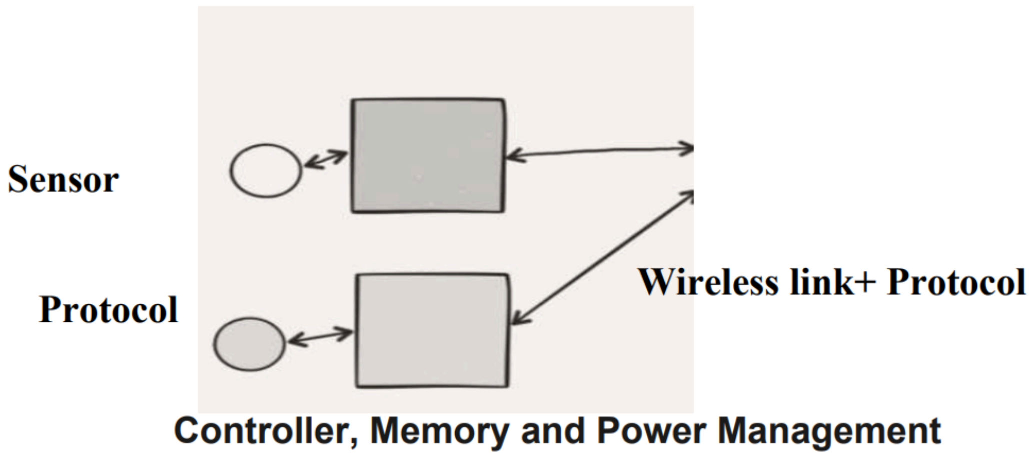
IoT System Design Cycle



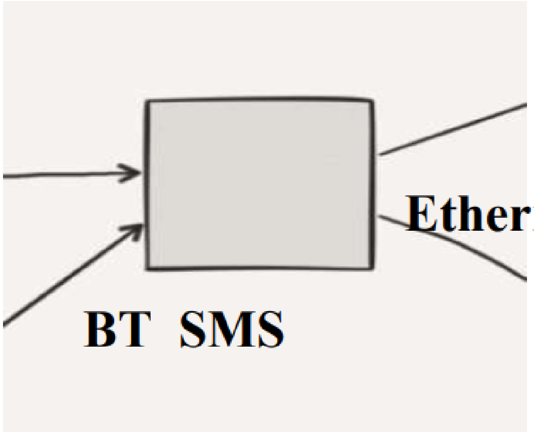
IoT Architecture



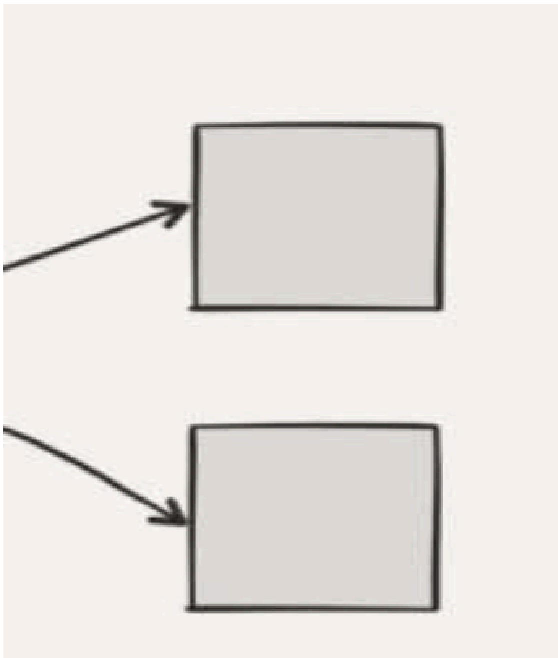
IoT Architecture : Node



IoT Architecture : Gateway



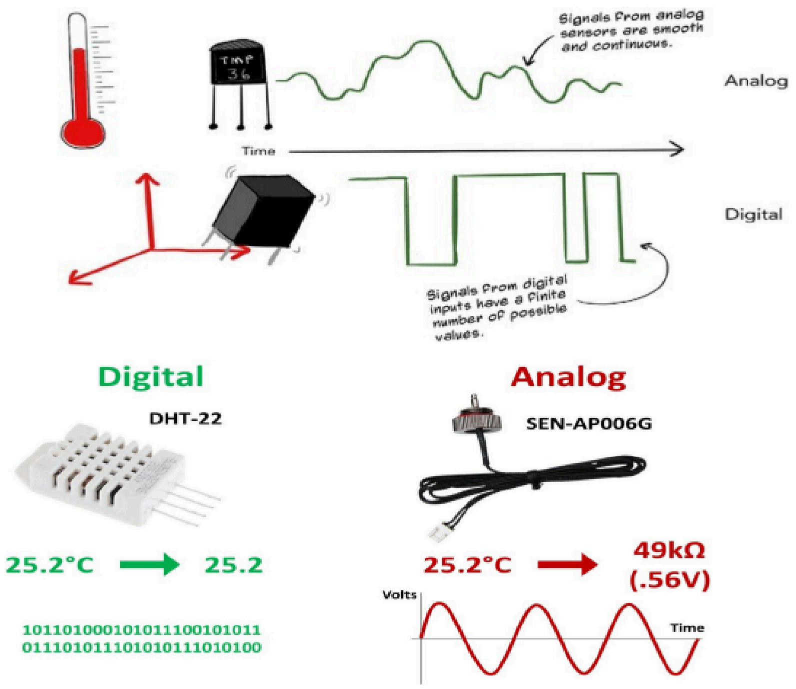
IoT Architecture : Services



Sensors

- Sensors measure or identify a particular quantity
- Convert physical quantities to electrical signals understood by machines

Type of Sensors : Analog and Digital



Type of batteries

1. Li-ion / Li-Poly (Lithium-ion / Lithium Polymer) Batteries

- **Chemistry**: Uses lithium ions as the primary component.
- **Energy Density**: Highest among rechargeable batteries, making them ideal for compact devices.
- **Applications**: Most popular for portable and wearable IoT devices due to their high energy density and lightweight nature.
- **Maintenance**: Low-maintenance; no regular full discharge needed to maintain capacity.
- **Handling**: Safe and relatively easy to handle with appropriate protection circuits.
- **Voltage**: Provides a typical voltage of 3.6-3.7V per cell.

2. Pb-Acid (Lead Acid) Batteries

- **Chemistry**: Based on lead dioxide and lead plates with sulfuric acid as an electrolyte.
- **Energy Density**: Lower than lithium-based batteries, making them bulkier.

- **Applications**: Commonly used in automotive, UPS (uninterruptible power supplies), and backup power systems.
- **Maintenance**: Requires periodic checks and water refilling (in some designs).
- **Handling**: Heavier and less suitable for portable applications.
- **Voltage**: Generally provides 2V per cell.

3. NiCd (Nickel Cadmium) Batteries

- **Chemistry**: Made of nickel oxide hydroxide and metallic cadmium.
- **Energy Density**: Moderate; not as high as lithium but better than lead acid.
- **Applications**: Often used in power tools, medical equipment, and aviation.
- **Maintenance**: Requires periodic full discharges to prevent "memory effect."
- **Handling**: Reliable under high-drain conditions, but contains toxic cadmium, making disposal challenging.
- **Voltage**: Provides around 1.2V per cell.

4. NiMH (Nickel Metal Hydride) Batteries

- **Chemistry**: Uses nickel oxide hydroxide and hydrogen-absorbing alloy.
- **Energy Density**: Higher than NiCd but less than Li-ion.
- **Applications**: Common in household rechargeable batteries, hybrid vehicles, and small electronics.
- **Maintenance**: Low maintenance compared to NiCd, with minimal memory effect.
- **Handling**: Safer and more environmentally friendly than NiCd.
- **Voltage**: Also around 1.2V per cell, similar to NiCd.

Sensors in IoT

1. Health Sensors

- **Types**: Heart rate, blood pressure, blood oxygen, glucose monitors, etc.
- **Applications**: Wearable health devices, remote patient monitoring, personal health tracking.

- **Purpose**: Tracks vital health parameters for monitoring and diagnostics.

2. Environmental Sensors

- **Types**: Light, temperature, humidity.
- **Applications**: Smart home devices, weather stations, agricultural monitoring.
- **Purpose**: Monitors environmental conditions to ensure optimal settings or track climate changes.

3. Building Automation Sensors

- **Types**: Pressure, temperature, motion.
- **Applications**: Smart building systems, HVAC control, security systems.
- **Purpose**: Automates building systems to enhance energy efficiency and security.

4. Motion and Orientation Sensors

- **Types**: Gyroscope, accelerometer.
- **Applications**: Transportation (vehicle tracking), smartphone orientation, gaming controllers.
- **Purpose**: Measures movement and orientation changes for positioning and tracking.

5. Chemical Sensors

- **Types**: Gas sensors, smoke detectors, pH sensors.
- **Applications**: Industrial monitoring, environmental safety, air quality monitoring.
- **Purpose**: Detects specific chemicals or gases for safety and environmental monitoring.

6. Industrial, Environment, Security, and Public Safety Sensors

- **Types**: Temperature, humidity, gas, vibration, light.
- **Applications**: Industrial automation, environmental monitoring, security alarms.
- **Purpose**: Ensures safety and efficiency by monitoring various parameters in industrial and public spaces.

7. Retail and Logistics Sensors

- **Types**: RFID, barcode, GPS.
- **Applications**: Inventory management, asset tracking, supply chain optimization.
- **Purpose**: Streamlines retail operations and improves logistics through real-time tracking of goods.

Smart Sensors

A **Smart Sensor** is an advanced type of sensor that integrates additional electronics to enhance its capabilities beyond basic sensing. Here's a detailed, pointwise explanation:

Key Features of Smart Sensors:

1. **Integrated Electronics**:
   - Smart sensors include built-in electronics such as microcontrollers, which allow them to process and convert raw data directly.
   - They perform functions like data conversion, calibration, and preprocessing, making the data usable without additional circuitry.
2. **Bidirectional Communication**:
   - These sensors support bidirectional communication, allowing them not only to send data but also to receive instructions or configurations.
   - This feature enables remote adjustments, diagnostics, and firmware updates.
3. **Decision-Making and Logic Operations**:
   - With onboard processing capabilities, smart sensors can analyze data, make decisions, and perform logical operations independently.
   - This can reduce the load on the central system by performing initial data processing at the sensor level.
4. **Built-in Integrated Circuit (IC)**:
   - Smart sensors have an integrated circuit (IC) that combines the sensing element and microcontroller.

    - The IC allows the sensor to provide a processed output (such as a digital signal) when connected to a supply voltage and programmed.

Example: Smart Temperature Sensor

- For a temperature sensor, the smart sensor can output data directly in a digital format, such as hexadecimal or binary, depending on the programmed calibration.
- For instance, if a smart temperature sensor is calibrated to output data in UART serial format (10 bits), it might give a binary output like 01100100 for 100°C.
- This means the sensor has already converted the analog temperature measurement to a digital output (in this example, binary or hexadecimal), making it ready for direct use by other systems.