1.4.5 Embedded Systems

An Embedded System is a computer system that has computer hardware and software embedded to perform specific tasks. In contrast to general purpose computers or personal computers (PCs) which can perform various types of tasks, embedded systems are designed to perform a specific set of tasks. Key components of an embedded system include, microprocessor or microcontroller, memory (RAM, ROM, cache), networking units (Ethernet, WiFi adapters), input/output units (display, keyboard, etc.) and storage (such as flash memory). Some embedded systems have specialized processors such as digital signal processors (DSPs), graphics processors and application specific processors. Embedded systems run embedded operating systems such as real-time operating systems (RTOS). Embedded systems range from low-cost miniaturized devices such as digital watches to devices such as digital cameras, point of sale terminals, vending machines, appliances (such as washing machines), etc. In the next chapter we describe how such devices form an integral part of IoT systems.

1.5 IoT Levels & Deployment Templates

In this section we define various levels of IoT systems with increasing completely. An IoT system comprises of the following components:

- **Device:** An IoT device allows identification, remote sensing, actuating and remote monitoring capabilities. You learned about various examples of IoT devices in section 1.2.1.
- **Resource:** Resources are software components on the IoT device for accessing, processing, and storing sensor information, or 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 either implemented using HTTP and REST principles (REST service) or using WebSocket protocol (WebSocket service). A comparison of REST and WebSocket is provided below:

- **Stateless/Stateful:** REST services are stateless in nature. Each request contains all the information needed to process it. Requests are independent of each other. WebSocket on the other hand is stateful in nature where the server maintains the state and is aware of all the open connections.
- **Uni-directional/Bi-directional:** REST services operate over HTTP and are uni-directional. Request is always sent by a client and the server responds to the requests. On the other hand, WebSocket is a bi-directional protocol and allows both client and server to send messages to each other.
- **Request-Response/Full Duplex:** REST services follow a request-response communication model where the client sends requests and the server responds to the requests. WebSocket on the other hand allow full-duplex communication between the client and server, i.e., both client and server can send messages to each other independently.
- **TCP Connections:** For REST services, each HTTP request involves setting up a new TCP connection. WebSocket on the other hand involves a single TCP connection over which the client and server communicate in a full-duplex mode.
- **Header Overhead:** REST services operate over HTTP, and each request is independent of others. Thus each request carries HTTP headers which is an overhead. Due the overhead of HTTP headers, REST is not suitable for real-time applications. WebSocket on the other hand does not involve overhead of headers. After the initial handshake (that happens over HTTP), the client and server exchange messages with minimal frame information. Thus WebSocket is suitable for real-time applications.
- **Scalability:** Scalability is easier in the case of REST services as requests are independent and no state information needs to be maintained by the server. Thus both horizontal (scaling-out) and vertical scaling (scaling-up) solutions are possible for REST services. For WebSockets, horizontal scaling can be cumbersome due to the stateful nature of the communication. Since the server maintains the state of a connection, vertical scaling is easier for WebSockets than horizontal scaling.

• **Analysis Component:** The Analysis Component is responsible for analyzing the IoT data and generate results in a form which are easy for the user to understand. Analysis of IoT data can be performed either locally or in the cloud. Analyzed results are stored in the local or cloud databases.

• **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 view the processed data.

1.5 IoT Leve

1.5.1 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 as shown in Figure 1.14. Level-1 IoT systems are suitable for modeling low-cost and low-complexity solutions where the data involved is not big and the analysis requirements are not computationally intensive.

Let us now consider an example of a level-1 IoT system for home automation. The system consists of a single node that allows controlling the lights and appliances in a home remotely. The device used in this system interfaces with the lights and appliances using electronic relay switches. The status information of each light or appliance is maintained in a local database. REST services deployed locally allow retrieving and updating the state of each light or appliance in the status database. The controller service continuously monitors the state of each light or appliance (by retrieving state from the database) and triggers the relay switches accordingly. The application which is deployed locally has a user interface for controlling the lights or appliances. Since the device is connected to the Internet, the application can be accessed remotely as well.

1.5.2 IoT Level-2

A level-2 IoT system has a single node that performs sensing and/or actuation and local analysis as shown in Figure 1.15. Data is stored in the cloud and 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 itself.

Let us consider an example of a level-2 IoT system for smart irrigation. The system consists of a single node that monitors the soil moisture level and controls the irrigation system. The device used in this system collects soil moisture data from sensors. The controller service continuously monitors the moisture levels. If the moisture level drops below a threshold, the irrigation system is turned on. For controlling the irrigation system actuators such as solenoid valves can be used. The controller also sends the moisture data to the computing cloud. A cloud-based REST web service is used for storing and retrieving moisture data which is stored in the cloud database. A cloud-based application is used for visualizing the moisture levels over a period of time, which can help in making decisions about irrigation schedules.

1.5.3 IoT Level-3

A level-3 IoT system has a single node. Data is stored and analyzed in the cloud and application is cloud-based as shown in Figure 1.16. Level-3 IoT systems are suitable for

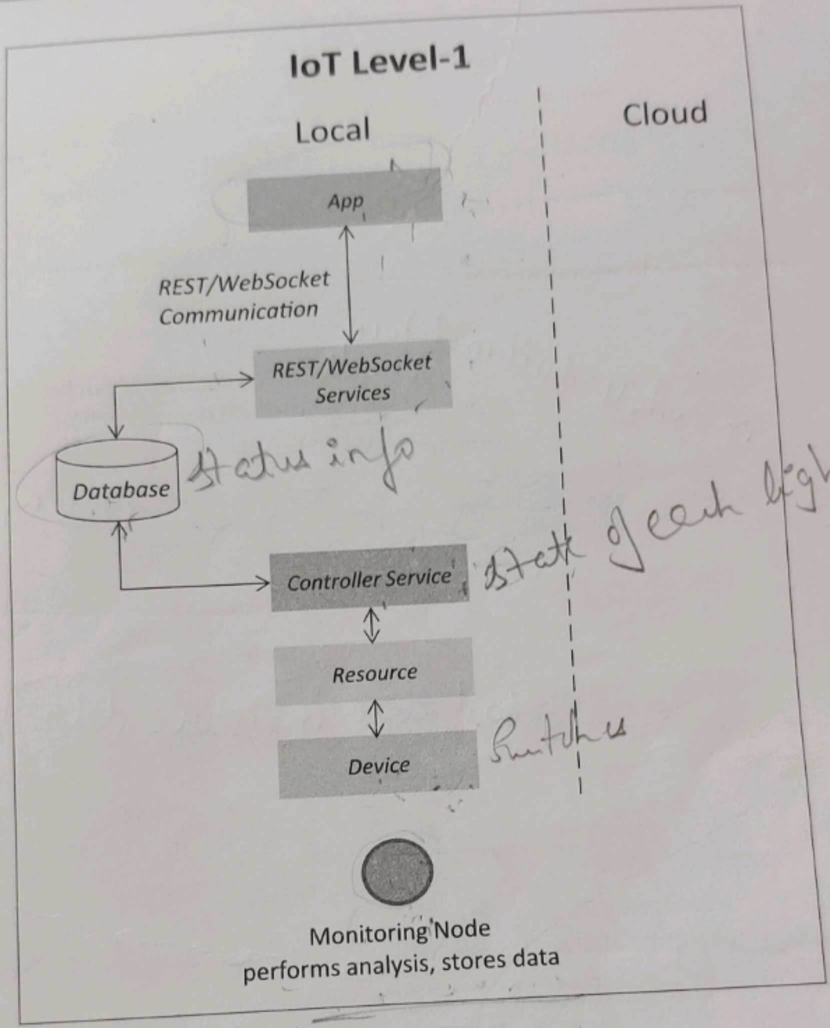

Figure 1.14: IoT Level-1

solutions where the data involved is big and the analysis requirements are computationally intensive.

Let us consider an example of a level-2 IoT system for tracking package handling. The system consists of a single node (for a package) that monitors the vibration levels for a package being shipped. The device in this system uses accelerometer and gyroscope sensors for monitoring vibration levels. The controller service sends the sensor data to the cloud in real-time using a WebSocket service. The data is stored in the cloud and also visualized using a cloud-based application. The analysis components in the cloud can trigger alerts if

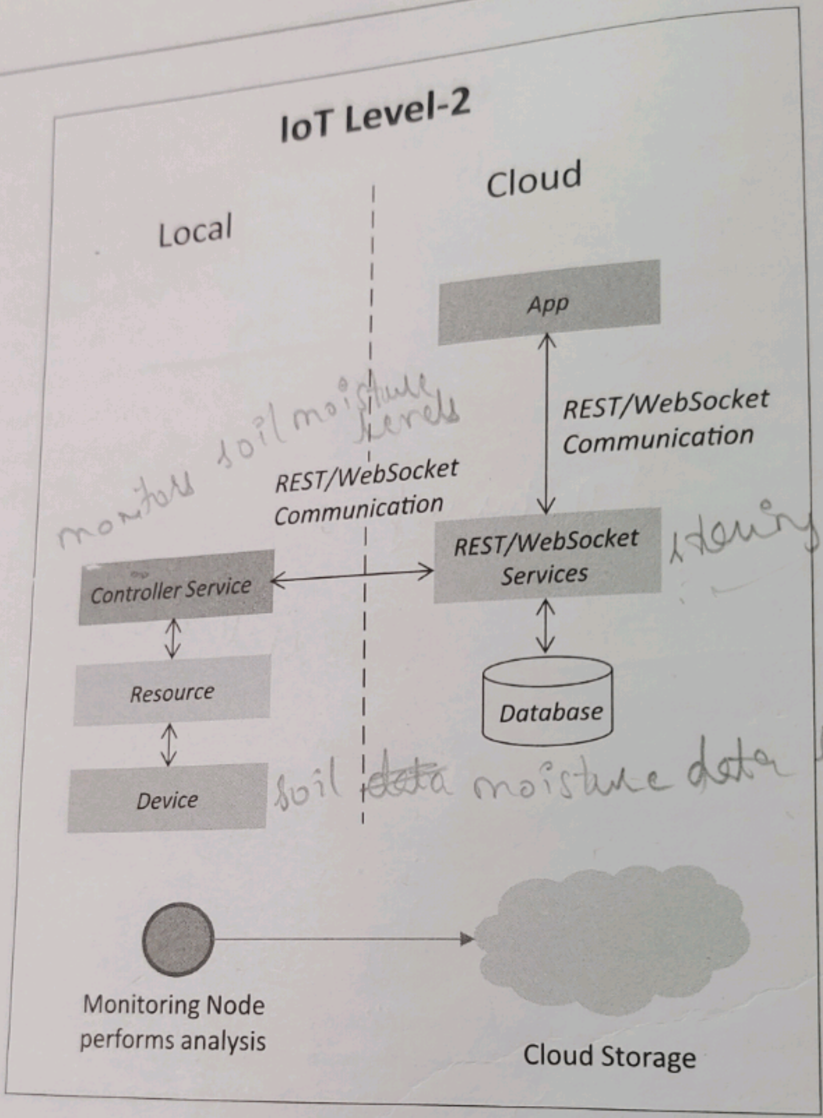

Figure 1.15: IoT Level-2

the vibration levels become greater than a threshold. The benefit of using WebSocket service instead of REST service in this example is that, the sensor data can be sent in real time to the cloud. Moreover, cloud based applications can subscribe to the sensor data feeds for viewing the real-time data.

1.5.4 IoT Level-4

A level-4 IoT system has multiple nodes that perform local analysis. Data is stored in the cloud and application is cloud-based as shown in Figure 1.17. Level-4 contains local and

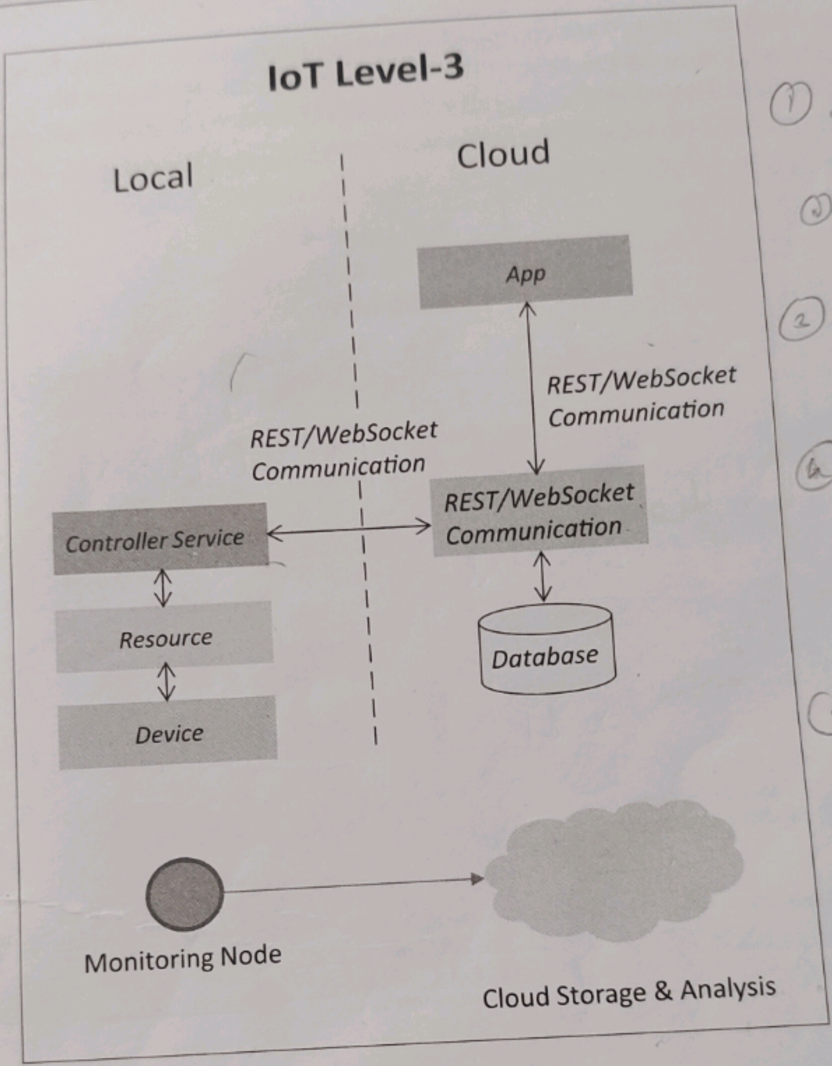

Figure 1.16: IoT Level-3

Eg: Noise Monitoring.

- Multiple nodes are used at diff places with sound sensors to collect data
- one node is independent of another
- data stored in cloud will be analysed & processed

① cloud & cloud based
② data is big
③ Multiple nodes are present
④ These just receive & subscribe the data from cloud
⑤ These nodes doesn't perform any control operation
⑥ A&R is computationally intensive

cloud-based observer nodes which can subscribe to and receive information collected in the cloud from IoT devices. Observer nodes can process information and use it for various applications, however, observer nodes do not perform any control functions. 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.

Let us consider an example of a level-4 IoT system for noise monitoring. The system consists of multiple nodes placed in different locations for monitoring noise levels in an area.

The nodes in this example are equipped with sound sensors. Nodes are independent of each other. Each node runs its own controller service that sends the data to the cloud. The data is stored in a cloud database. The analysis of data collected from a number of nodes is done in the cloud. A cloud-based application is used for visualizing the aggregated data.

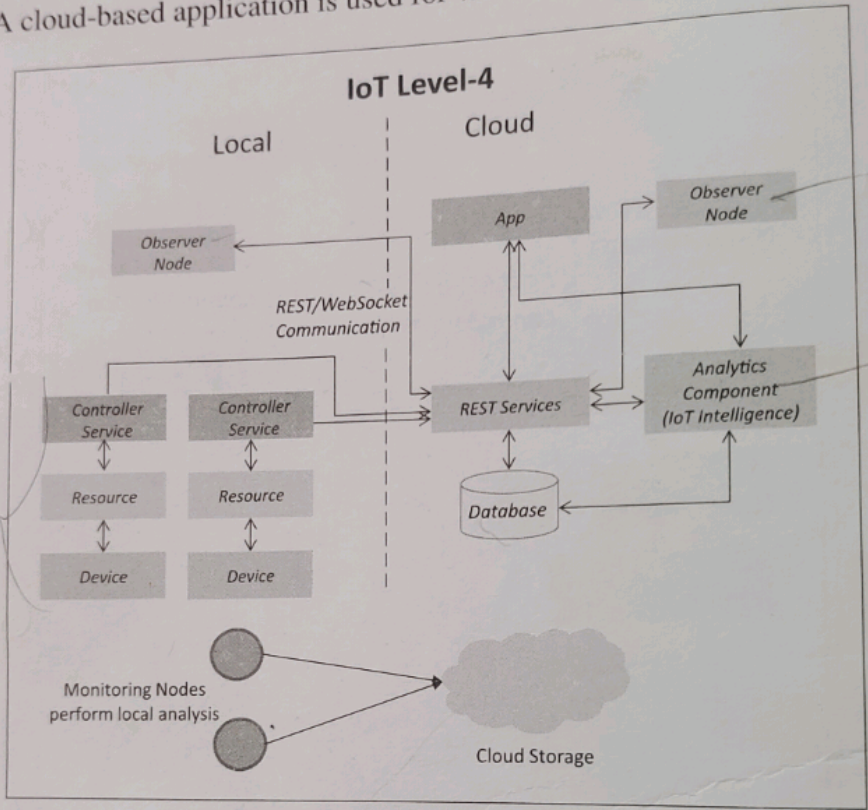

Figure 1.17: IoT Level-4

1.5.5 IoT Level-5

A level-5 IoT system has multiple end nodes and one coordinator node as shown in Figure 1.18. The end nodes that perform sensing and/or actuation. Coordinator node collects data from the end nodes and sends to the cloud. Data is stored and analyzed in the cloud and 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.

Let us consider an example of a level-5 IoT system for forest fire detection. The system consists of multiple nodes placed in different locations for monitoring temperature, humidity and carbon dioxide (CO_2) levels in a forest. The end nodes in this example are equipped with various sensors (such as temperature, humidity and CO_2). The coordinator node collects the data from the end nodes and acts as a gateway that provides Internet connectivity to the IoT system. The controller service on the coordinator device sends the collected data to the cloud. The data is stored in a cloud database. The analysis of data is done in the computing cloud to aggregate the data and make predictions. A cloud-based application is used for visualizing the data.

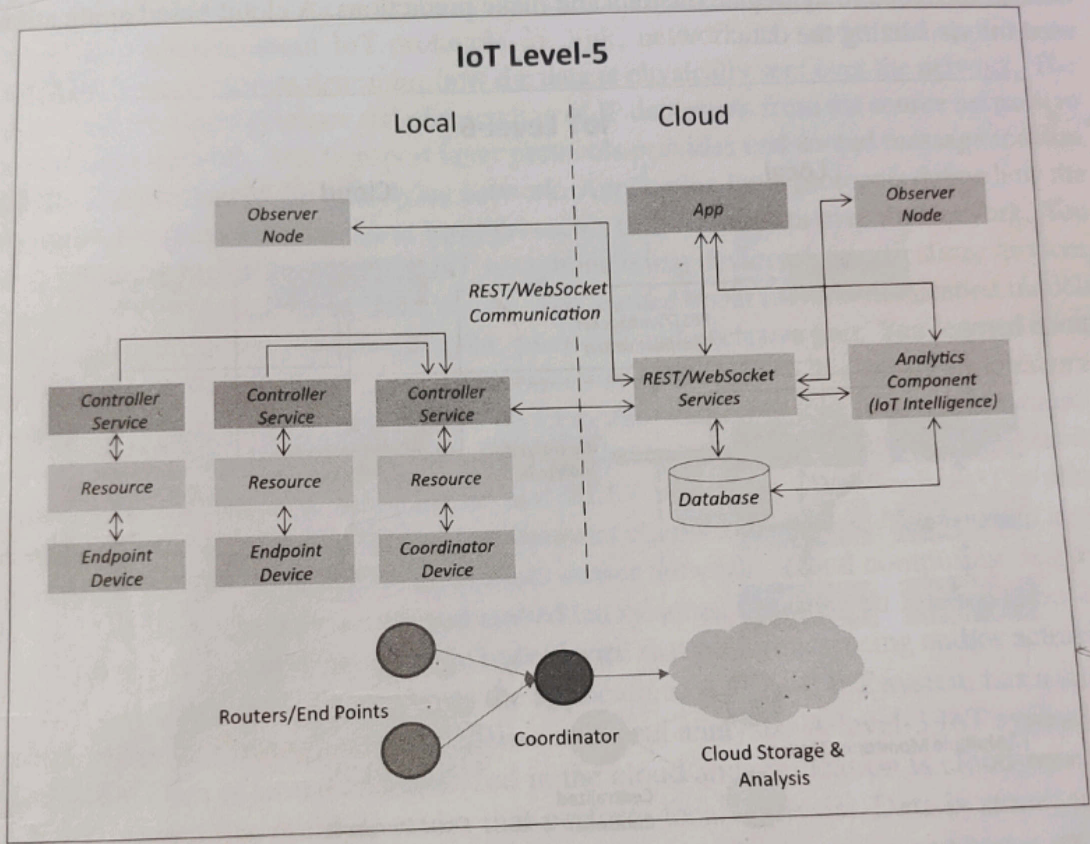

Figure 1.18: IoT Level-5

1.5.6 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 application is cloud-based as shown in Figure 1.19. 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.

Let us consider an example of a level-6 IoT system for weather monitoring. The system consists of multiple nodes placed in different locations for monitoring temperature, humidity and pressure in an area. The end nodes are equipped with various sensors (such as temperature, pressure and humidity). The end nodes send the data to the cloud in real-time using a WebSocket service. The data is stored in a cloud database. The analysis of data is done in the cloud to aggregate the data and make predictions. A cloud-based application is used for visualizing the data.

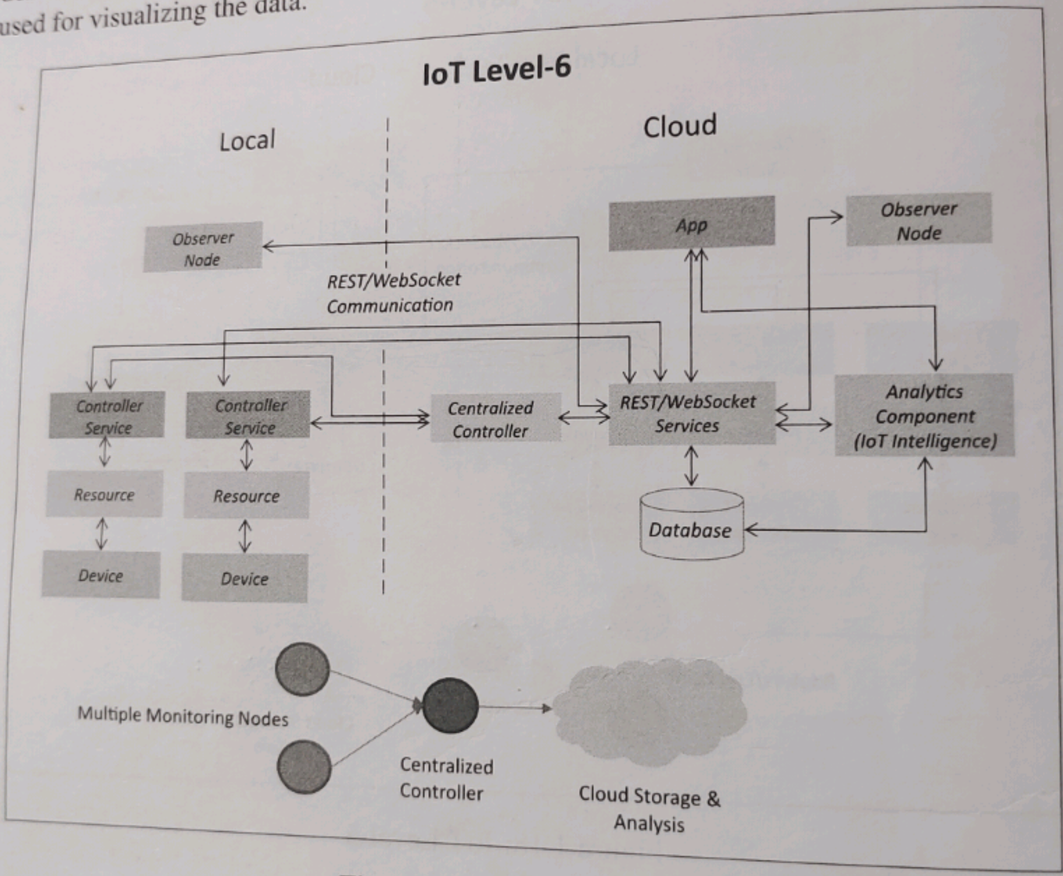

Figure 1.19: IoT Level-6

Summary

Internet of Things (IoT) refers to physical and virtual objects that have unique identities and are connected to the Internet. This allows the development of intelligent applications that make energy, logistics, industrial control, retail, agriculture and many other domains of

2.1 Introduction

The Internet of Things (IoT) applications span a wide range of domains including (but not limited to) homes, cities, environment, energy systems, retail, logistics, industry, agriculture and health. This chapter provides an overview of various types of IoT applications for each of these domains. In the later chapters the reader is guided through detailed implementations of several of these applications.

2.2 Home Automation

2.2.1 Smart Lighting

Smart lighting for homes helps in saving energy by adapting the lighting to the ambient conditions and switching on/off or dimming the lights when needed. Key enabling technologies for smart lighting include solid state lighting (such as LED lights) and IP-enabled lights. For solid state lighting solutions both spectral and temporal characteristics can be configured to adapt illumination to various needs. Smart lighting solutions for home achieve energy savings by sensing the human movements and their environments and controlling the lights accordingly. Wireless-enabled and Internet connected lights can be controlled remotely from IoT applications such as a mobile or web application. Smart lights with sensors for occupancy, temperature, lux level, etc., can be configured to adapt the lighting (by changing the light intensity, color, etc.) based on the ambient conditions sensed, in order to provide a good ambiance. In [19] controllable LED lighting system is presented that is embedded with ambient intelligence gathered from a distributed smart wireless sensor network to optimize and control the lighting system to be more efficient and user-oriented. A solid state lighting model is described in [20] and implemented on a wireless sensor network that provides services for sensing illumination changes and dynamically adjusting luminary brightness according to user preferences. In chapter-9 we provide a case study on a smart lighting system.

2.2.2 Smart Appliances

Modern homes have a number of appliances such as TVs, refrigerators, music systems, washer/dryers, etc. Managing and controlling these appliances can be cumbersome, with each appliance having its own controls or remote controls. Smart appliances make the management easier and also provide status information to the users remotely. For example, smart washer/dryers that can be controlled remotely and notify when the washing/drying cycle is complete. Smart thermostats allow controlling the temperature remotely and can learn the user preferences [22]. Smart refrigerators can keep track of the items stored (using RFID tags) and send updates to the users when an item is low on stock. Smart

TVs allows users to search and stream videos and movies from the Internet on a local storage drive, search TV channel schedules and fetch news, weather updates and other content from the Internet. OpenRemote [21] is an open source automation platform for homes and buildings. OpenRemote is platform agnostic and works with standard hardware. With OpenRemote, users can control various appliances using mobile or web applications. OpenRemote comprises of three components - a Controller that manages scheduling and runtime integration between devices, a Designer that allows you to create both configurations for the controller and create user interface designs and Control Panels that allow you to interact with devices and control them. An IoT-based appliance control system for smart homes is described in [23], that uses a smart central controller to set up a wireless sensor and actuator network and control modules for appliances.

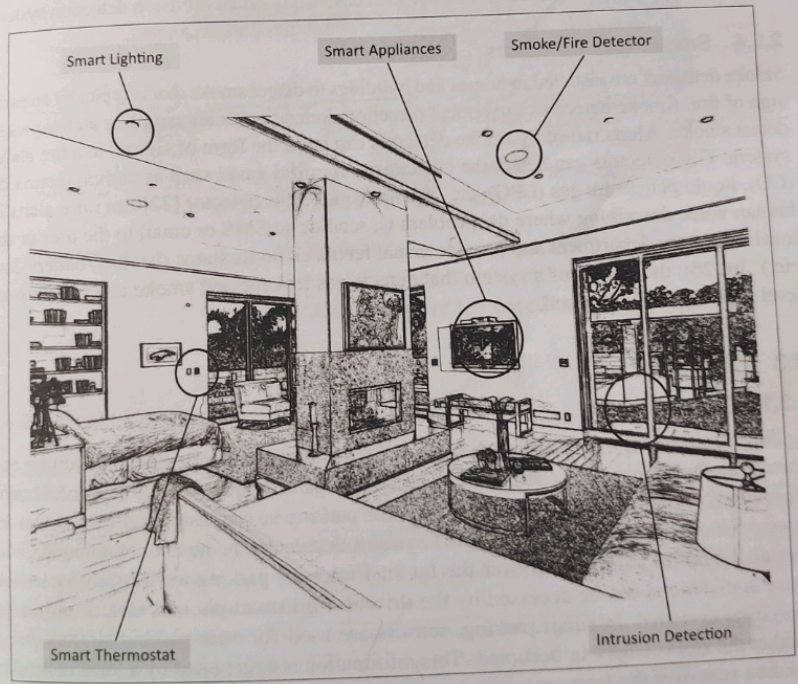

Figure 2.1: Applications of IoT for homes

2.2.3 Intrusion Detection

Home intrusion detection systems use security cameras and sensors (such as PIR sensors and door sensors) to detect intrusions and raise alerts. Alerts can be in the form of an SMS or an email sent to the user. Advanced systems can even send detailed alerts such as an image grab or a short video clip sent as an email attachment. A cloud controlled intrusion detection system is described in [24] that uses location-aware services, where the geo-location of each node of a home automation system is independently detected and stored in the cloud. In the event of intrusions, the cloud services alert the accurate neighbors (who are using the home automation system) or local police. In [25], an intrusion detection system based on UPnP technology is described. The system uses image processing to recognize the intrusion and extract the intrusion subject and generate Universal-Plug-and-Play (UPnP-based) instant messaging for alerts. In chapter-9 we provide a case study on an intrusion detection system.

2.2.4 Smoke/Gas Detectors

Smoke detectors are installed in homes and buildings to detect smoke that is typically an early sign of fire. Smoke detectors use optical detection, ionization or air sampling techniques to detect smoke. Alerts raised by smoke detectors can be in the form of signals to a fire alarm system. Gas detectors can detect the presence of harmful gases such as carbon monoxide (CO), liquid petroleum gas (LPG), etc. A smart smoke/gas detector [22] can raise alerts in human voice describing where the problem is, send or an SMS or email to the user or the local fire safety department and provide visual feedback on its status (healthy, battery-low, etc.). In [26], the design of a system that detects gas leakage and smoke and gives visual level indication, is described.

2.3 Cities

2.3.1 Smart Parking

Finding a parking space during rush hours in crowded cities can be time consuming and frustrating. Furthermore, drivers blindly searching for parking spaces create additional traffic congestion. Smart parking make the search for parking space easier and convenient for drivers. Smart parking are powered by IoT systems that detect the number of empty parking slots and send the information over the Internet to smart parking application back-ends. These applications can be accessed by the drivers from smart-phones, tablets and in-car navigation systems. In smart parking, sensors are used for each parking slot, to detect whether the slot is empty or occupied. This information is aggregated by a local controller and then sent over the Internet to the database. In [29], Polycarpou *et. al.* describe latest trends in parking availability monitoring, parking reservation and dynamic pricing schemes.

Design and implementation of a prototype smart parking system based on wireless sensor network technology with features like remote parking monitoring, automated guidance, and parking reservation mechanism is described in [30]. In chapter-9 we provide a case study on a smart parking system.

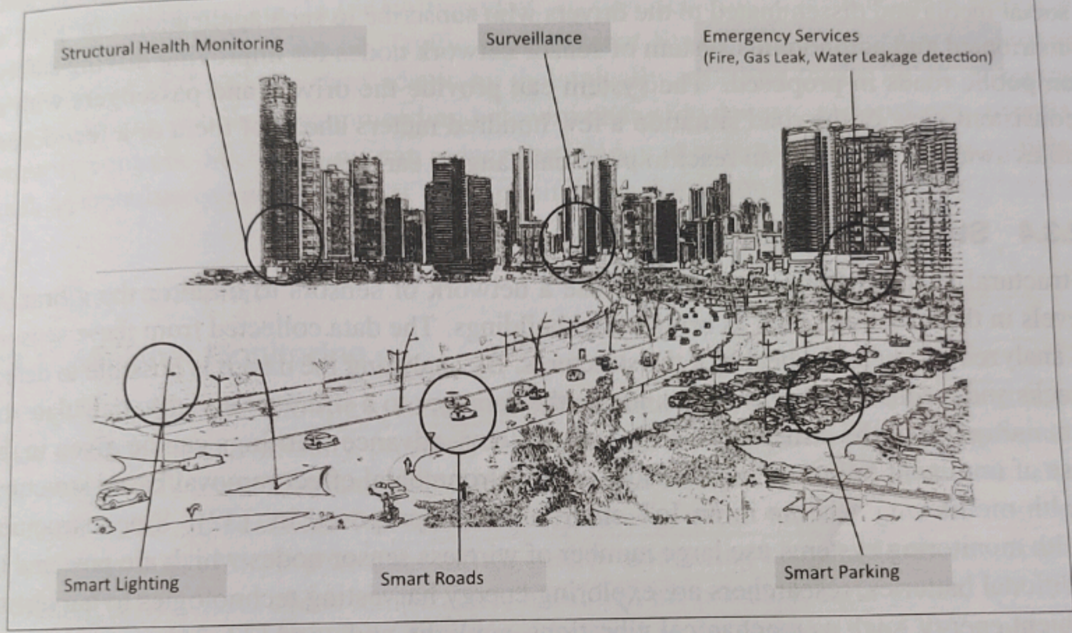

Figure 2.2: Applications of IoT for cities

2.3.2 Smart Lighting

Smart lighting systems for roads, parks and buildings can help in saving energy. According to an IEA report [27], lighting is responsible for 19% of global electricity use and around 6% of global greenhouse gas emissions. Smart lighting allows lighting to be dynamically controlled and also adaptive to the ambient conditions. Smart lights connected to the Internet can be controlled remotely to configure lighting schedules and lighting intensity. Custom lighting configurations can be set for different situations such as a foggy day, a festival, etc. Smart lights equipped with sensors can communicate with other lights and exchange information on the sensed ambient conditions to adapt the lighting. Castro *et. al.* [28] describe the need for smart lighting system in smart cities, smart lighting features and how to develop interoperable smart lighting solutions.

2.3.3 Smart Roads

Smart roads equipped with sensors can provide information on driving conditions, travel time estimates and alerts in case of poor driving conditions, traffic congestions and accidents. Such information can help in making the roads safer and help in reducing traffic jams. Information sensed from the roads can be communicated via Internet to cloud-based applications and social media and disseminated to the drivers who subscribe to such applications. In [31], a distributed and autonomous system of sensor network nodes for improving driving safety on public roads in proposed. The system can provide the drivers and passengers with a consistent view of the road situation a few hundred meters ahead of them or a few dozen miles away, so that they can react to potential dangers early enough.

2.3.4 Structural Health Monitoring

Structural Health Monitoring systems use a network of sensors to monitor the vibration levels in the structures such as bridges and buildings. The data collected from these sensors is analyzed to assess the health of the structures. By analyzing the data it is possible to detect cracks and mechanical breakdowns, locate the damages to a structure and also calculate the remaining life of the structure. Using such systems, advance warnings can be given in the case of imminent failure of the structure. An environmental effect removal based structural health monitoring scheme in an IoT environment is proposed in [32]. Since structural health monitoring systems use large number of wireless sensor nodes which are powered by traditional batteries, researchers are exploring energy harvesting technologies to harvesting ambient energy, such as mechanical vibrations, sunlight, and wind [33, 34].

2.3.5 Surveillance

Surveillance of infrastructure, public transport and events in cities is required to ensure safety and security. City wide surveillance infrastructure comprising of large number of distributed and Internet connected video surveillance cameras can be created. The video feeds from surveillance cameras can be aggregated in cloud-based scalable storage solutions. Cloud-based video analytics applications can be developed to search for patterns or specific events from the video feeds. In [35] a smart city surveillance system is described that leverages benefits of cloud data stores.

2.3.6 Emergency Response

IoT systems can be used for monitoring the critical infrastructure in cities such as buildings, gas and water pipelines, public transport and power substations. IoT systems for fire detection, gas and water leakage detection can help in generating alerts and minimizing their effects on the critical infrastructure. IoT systems for critical infrastructure monitoring enable aggregation and sharing of information collected from large number of sensors. Using cloud-based architectures, multi-modal information such as sensor data, audio, video feeds can be analyzed in near real-time to detect adverse events. Response to alerts generated by such systems can be in the form of alerts sent to the public, re-routing of traffic, evacuations of the affected areas, etc. In [36] Attwood *et. al*. describe critical infrastructure response framework for smart cities. A Traffic Management System for emergency services is described in [37]. The system adapts by dynamically adjusting traffic lights, changing related driving policies, recommending behavior change to drivers, and applying essential security controls. Such systems can reduce the latency of emergency services for vehicles such as ambulances and police cars while minimizing disruption of regular traffic.

2.4 Environment

2.4.1 Weather Monitoring

IoT-based weather monitoring systems can collect data from a number of sensor attached (such as temperature, humidity, pressure, etc.) and send the data to cloud-based applications and storage back-ends. The data collected in the cloud can then be analyzed and visualized by cloud-based applications. Weather alerts can be sent to the subscribed users from such applications. AirPi [38] is a weather and air quality monitoring kit capable of recording and uploading information about temperature, humidity, air pressure, light levels, UV levels, carbon monoxide, nitrogen dioxide and smoke level to the Internet. In [39], a pervasive weather monitoring system is described that is integrated with buses to measure weather variables like humidity, temperature and air quality during the bus path. In [40], a weather monitoring system based on wireless sensor networks is described. In chapter-9 we provide a case study on a weather monitoring system.

2.4.2 Air Pollution Monitoring

IoT based air pollution monitoring systems can monitor emission of harmful gases (CO_2, CO, NO, NO_2, etc.) by factories and automobiles using gaseous and meteorological sensors. The collected data can be analyzed to make informed decisions on pollutions control approaches. In [41], a real-time air quality monitoring system is presented that comprises of several distributed monitoring stations that communicate via wireless with a back-end server using machine-to-machine communication. In [42], an air pollution system is described that integrates a single-chip microcontroller, several air pollution sensors, GPRS-Modem, and a GPS module. In chapter-9 we provide a case study on an air pollution monitoring system.

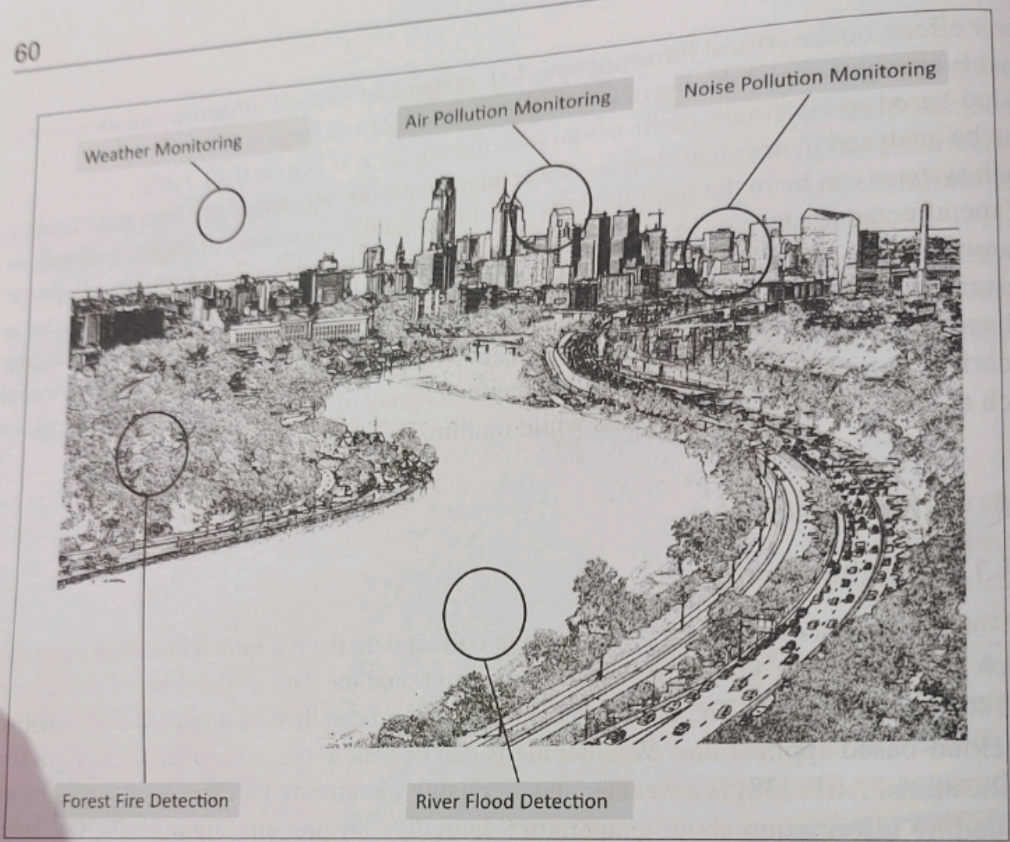

Figure 2.3: Applications of IoT for environment

2.4.3 Noise Pollution Monitoring

Due to growing urban development, noise levels in cities have increased and even become alarmingly high in some cities. Noise pollution can cause health hazards for humans due to sleep disruption and stress. Noise pollution monitoring can help in generating noise maps for cities. Urban noise maps can help the policy makers in urban planning and making policies to control noise levels near residential areas, schools and parks. IoT based noise pollution monitoring systems use a number of noise monitoring stations that are deployed at different places in a city. The data on noise levels from the stations is collected on servers or in the cloud. The collected data is then aggregated to generate noise maps. In [43], a noise mapping study for a city is presented which revealed that the city suffered from serious noise pollution. In [44], the design of smart phone application is described that allows the users to continuously measure noise levels and send to a central server where all generated information is aggregated and mapped to a meaningful noise visualization map.

2.4.4 Forest Fire Detection

Forest fires can cause damage to natural resources, property and human life. There can be different causes of forest fires including lightening, human negligence, volcanic eruptions and sparks from rock falls. Early detection of forest fires can help in minimizing the damage. IoT based forest fire detection systems use a number of monitoring nodes deployed at different locations in a forest. Each monitoring node collects measurements on ambient conditions including temperature, humidity, light levels, etc. A system for early detection of forest fires is described in [45] that provides early warning of a potential forest fire and estimates the scale and intensity of the fire if it materializes. In [46], a forest fire detection system based on wireless sensor networks in presented. The system uses multi-criteria detection which is implemented by the artificial neural network (ANN). The ANN fuses sensing data corresponding to multiple attributes of a forest fire (such as temperature, humidity, infrared and visible light) to detect forest fires.

2.4.5 River Floods Detection

River floods can cause extensive damage to the natural and human resources and human life. River floods occur due to continuous rainfall which cause the river levels to rise and flow rates to increase rapidly. Early warnings of floods can be given by monitoring the water level and flow rate. IoT based river flood monitoring system use a number of sensor nodes that monitor the water level (using ultrasonic sensors) and flow rate (using the flow velocity sensors). Data from a number of such sensor nodes is aggregated in a server or in the cloud. Monitoring applications raise alerts when rapid increase in water level and flow rate is detected. In [47], a river flood monitoring system in described that measures river and weather conditions through wireless sensor nodes equipped with different sensors. In [48], a motes-based sensor network for river flood monitoring is described. The system includes a water level monitoring module, network video recorder module, and data processing module that provides flood information in the form of raw data, predicted data, and video feed.

2.5 Energy

2.5.1 Smart Grids

Smart Grid is a data communications network integrated with the electrical grid that collects and analyzes data captured in near-real-time about power transmission, distribution, and consumption. Smart Grid technology provides predictive information and recommendations to utilities, their suppliers, and their customers on how best to manage power. Smart

Grids collect data regarding electricity generation (centralized or distributed), consumption (instantaneous or predictive), storage (or conversion of energy into other forms), distribution and equipment health data. Smart grids use high-speed, fully integrated, two-way communication technologies for real-time information and power exchange. By using IoT based sensing and measurement technologies, the health of equipment and the integrity of the grid can be evaluated. Smart meters can capture almost real-time consumption, remotely control the consumption of electricity and remotely switch off supply when required. Power thefts can be prevented using smart metering. By analyzing the data on power generation, transmission and consumption smart girds can improve efficiency throughout the electric system. Storage collection and analysis of smarts grids data in the cloud can help in dynamic optimization of system operations, maintenance, and planning. Cloud-based monitoring of smart grids data can improve energy usage levels via energy feedback to users coupled with real-time pricing information. Real-time demand response and management strategies can be used for lowering peak demand and overall load via appliance control and energy storage mechanisms. Condition monitoring data collected from power generation and transmission systems can help in detecting faults and predicting outages. In [49], application of IoT in smart grid power transmission is described.

2.5.2 Renewable Energy Systems

Due to the variability in the output from renewable energy sources (such as solar and wind), integrating them into the grid can cause grid stability and reliability problems. Variable output produces local voltage swings that can impact power quality. Existing grids were designed to handle power flows from centralized generation sources to the loads through transmission and distribution lines. When distributed renewable energy sources are integrated into the grid, they create power bi-directional power flows for which the grids were not originally designed. IoT based systems integrated with the transformers at the point of interconnection measure the electrical variables and how much power is fed into the grid. To ensure the grid stability, one solution is to simply cut off the overproduction. For wind energy systems, closed-loop controls can be used to regulate the voltage at point of interconnection which coordinate wind turbine outputs and provides reactive power support [52].

2.5.3 Prognostics

Energy systems (smart grids, power plants, wind turbine farms, for instance) have a large number of critical components that must function correctly so that the systems can perform their operations correctly. For example, a wind turbine has a number of critical components, e.g., bearings, turning gears, for instance, that must be monitored carefully as wear and tear in such critical components or sudden change in operating conditions of the machines can

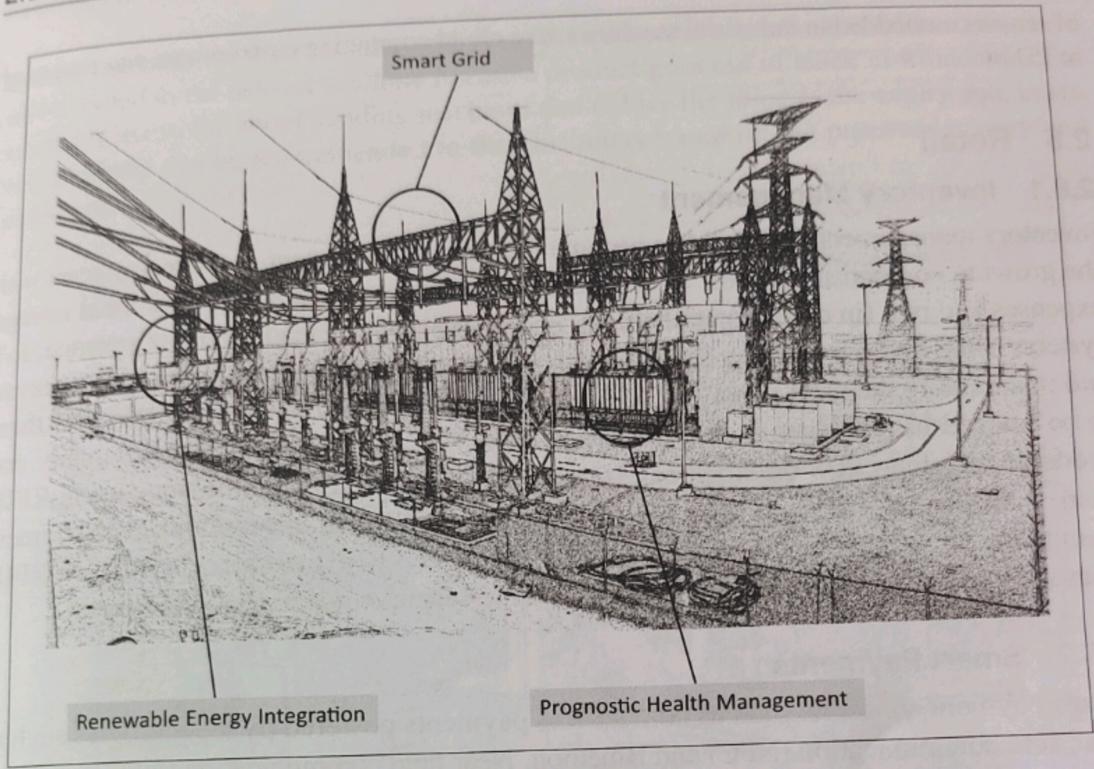

Figure 2.4: Applications of IoT for energy systems

result in failures. In systems such as power grids, real-time information is collected using specialized electrical sensors called Phasor Measurement Units (PMU) at the substations. The information received from PMUs must be monitored in real-time for estimating the state of the system and for predicting failures. Energy systems have thousands of sensors that gather real-time maintenance data continuously for condition monitoring and failure prediction purposes. IoT based prognostic real-time health management systems can predict performance of machines or energy systems by analyzing the extent of deviation of a system from its normal operating profiles. Analyzing massive amounts of maintenance data collected from sensors in energy systems and equipment can provide predictions for the impending failures (potentially in real-time) so that their reliability and availability can be improved. Prognostic health management systems have been developed for different energy systems. OpenPDC [50] is a set of applications for processing of streaming time-series data collected from Phasor Measurement Units (PMUs) in real-time. A generic framework for storage, processing and analysis of massive machine maintenance data, collected from a large number of sensors embedded in industrial machines, in a cloud computing environment was proposed in [51].

2.6 Retail

2.6.1 Inventory Management

Inventory management for retail has become increasingly important in the recent years with the growing competition. While over-stocking of products can result in additional storage expenses and risk (in case of perishables), under-stocking can lead to loss of revenue. IoT systems using Radio Frequency Identification (RFID) tags can help in inventory management and maintaining the right inventory levels. RFID tags attached to the products allow them to be tracked in real-time so that the inventory levels can be determined accurately and products which are low on stock can be replenished. Tracking can be done using RFID readers attached to the retail store shelves or in the warehouse. IoT systems enable remote monitoring of inventory using the data collected by the RFID readers. In [53], an RFID data-based inventory management system for time-sensitive materials is described.

2.6.2 Smart Payments

Smart payment solutions such as contact-less payments powered by technologies such as Near field communication (NFC) and Bluetooh. Near field communication (NFC) is a set of standards for smart-phones and other devices to communicate with each other by bringing them into proximity or by touching them. Customers can store the credit card information in their NFC-enabled smart-phones and make payments by bringing the smart-phones near the point of sale terminals. NFC maybe used in combination with Bluetooh, where NFC (which offers low speeds) initiates initial pairing of devices to establish a Bluetooh connection while the actual data transfer takes place over Bluetooh. The applications of NFC for contact-less payments are described in [54, 55].

2.6.3 Smart Vending Machines

Smart vending machines connected to the Internet allow remote monitoring of inventory levels, elastic pricing of products, promotions, and contact-less payments using NFC. Smart-phone applications that communicate with smart vending machines allow user preferences to be remembered and learned with time. When a user moves from one vending machine to the other and pairs the smart-phone with the vending machine, a user specific interface is presented. Users can save their preferences and favorite products. Sensors in a smart vending machine monitor its operations and send the data to the cloud which can be used for predictive maintenance. Smart vending machines can communicate with other vending machines in their vicinity and share their inventory levels so that the customers can be routed to the nearest machine in case a product goes out of stock in a machine. For perishable items, the smart vending machines can reduce the price as the expiry date nears. New products can be recommended to the customers based on the purchase history and preferences.

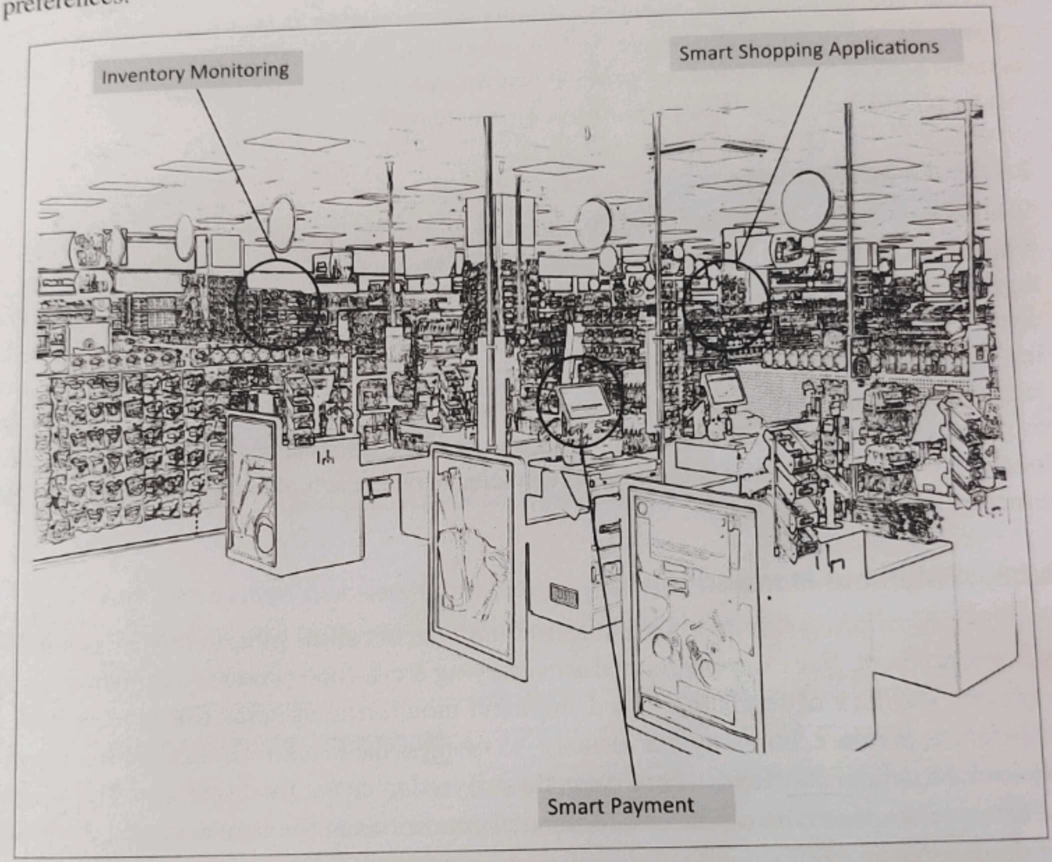

Figure 2.5: Applications of IoT for retail

2.7 Logistics

2.7.1 Route Generation & Scheduling

Modern transportation systems are driven by data collected from multiple sources which is processed to provide new services to the stakeholders. By collecting large amount of data from various sources and processing the data into useful information, data-driven transportation systems can provide new services such as advanced route guidance [62, 63], dynamic vehicle routing [64], anticipating customer demands for pickup and delivery problem, for instance. Route generation and scheduling systems can generate end-to-end routes using combination of route patterns and transportation modes and feasible schedules based on the availability of vehicles. As the transportation network grows in size and complexity, the number of possible route combinations increases exponentially. IoT based systems backed by the cloud can provide fast response to the route generation queries and can be scaled up to serve a large transportation network.

2.7.2 Fleet Tracking

Vehicle fleet tracking systems use GPS technology to track the locations of the vehicles in real-time. Cloud-based fleet tracking systems can be scaled up on demand to handle large number of vehicles. Alerts can be generated in case of deviations in planned routes. The vehicle locations and routes data can be aggregated and analyzed for detecting bottlenecks in the supply chain such as traffic congestions on routes, assignments and generation of alternative routes, and supply chain optimization. In [58], a fleet tracking system for commercial vehicles is described. The system can analyze messages sent from the vehicles to identify unexpected incidents and discrepancies between actual and planned data, so that remedial actions can be taken.

2.7.3 Shipment Monitoring

Shipment monitoring solutions for transportation systems allow monitoring the conditions inside containers. For example, containers carrying fresh food produce can be monitored to prevent spoilage of food. IoT based shipment monitoring systems use sensors such as temperature, pressure, humidity, for instance, to monitor the conditions inside the containers and send the data to the cloud, where it can be analyzed to detect food spoilage. The analysis and interpretation of data on the environmental conditions in the container and food truck positioning can enable more effective routing decisions in real time. Therefore, it is possible to take remedial measures such as - the food that has a limited time budget before it gets rotten can be re-routed to a closer destinations, alerts can be raised to the driver and the distributor about the transit conditions, such as container temperature exceeding the allowed limit, humidity levels going out of the allowed limit, for instance, and corrective actions can be taken before the food gets damaged. A cloud-based framework for real-time fresh food supply tracking and monitoring was proposed in [61]. For fragile products, vibration levels during shipments can be tracked using accelerometer and gyroscope sensors attached to IoT devices. In [59], a system for monitoring container integrity and operating conditions

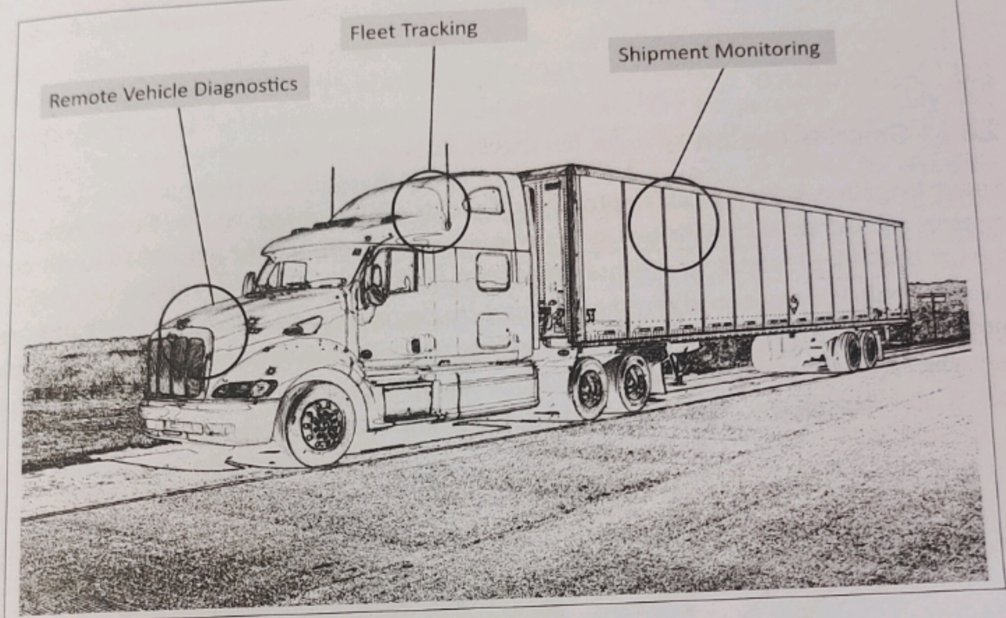

Figure 2.6: Applications of IoT for logistics

is described. The system monitors the vibration patterns of a container and its contents to reveal information related to its operating environment and integrity during transport, handling and storage.

2.7.4 Remote Vehicle Diagnostics

Remote vehicle diagnostic systems can detect faults in the vehicles or warn of impending faults. These diagnostic systems use on-board IoT devices for collecting data on vehicle operation (such as speed, engine RPM, coolent temperature, fault code number) and status of various vehicle sub-systems. Such data can be captured by integrating on-board diagnostic systems with IoT devices using protocols such as CAN bus. Modern commercial vehicles support on-board diagnostic (OBD) standards such as OBD-II. OBD systems provide real-time data on the status of vehicle sub-systems and diagnostic trouble codes which allow rapidly identifying the faults in the vehicle. IoT based vehicle diagnostic systems can send the vehicle data to centralized servers or the cloud where it can be analyzed to generate alerts and suggest remedial actions. In [60], a real-time online vehicle diagnostics and early fault estimation system is described. The system makes use of on-board vehicle diagnostics