MU Sem 7 INFORMATION TECHNOLOGY

New Syllabus 2022-23

- With Solved Latest UNIVERSITY QUESTION PAPERS.
- Concepts cleared with diagrams and examples.

Strictly as per the New Syllabus (REV-2019 'C' Scheme) of Mumbai University w.e.f. academic year 2022-2023

Compulsory Subject (Code : ITC702)

INTERNET OF EVERYTHING

Manoj [illegible]avedia
(Thadomal Shahani Engineering College, Bandra)

Dr. Mahesh R. Sanghavi
Prof. Kainjan Sanghavi
Prof. Rajiv Bhandari
Prof. Dipesh Agrawal
(SNJB's Late Sau. Kantabai Bhavarlalji Jain College of Engineering, Chandwad (Nashik))

ISBN : 978-93-5583-177-4

TECH-NEO PUBLICATIONS
Where Authors Inspire Innovation
A Sachin Shah Venture

www.techneobooks.in
info@techneobooks.in

M7-63A

Price ₹ 395/-

Prerequisite :

(1) Python programming (2) C programming language (3) Computer Networks

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Ports, Timers, Programming of controller, How to use IDE to write code of microcontroller, TCP-IP protocol stack.	02	
1.	Introduction to IoT	Introduction to IoT- Defining IoT, Characteristics of IoT, Conceptual Framework of IoT, Physical design of IoT, Logical design of IoT, Functional blocks of IoT, Brief review of applications of IoT. Smart Object - Definition, Characteristics and Trends **Self-learning Topics :** Hardware and software development tools for - Arduino, NodeMCU, ESP32, Raspberry Pi, for implementing internet of things, Simulators-Circuit.io, Eagle, Tinkercad. **(Refer Chapter 1)**	04	CO1
2.	IoT Architecture	**Drivers Behind New Network Architectures** : Scale, Security, Constrained Devices and Networks, Data, Legacy Device Support **Architecture :** The IoT World Forum (IoTWF) Standardized Architecture : Layer 1-7, IT and OT Responsibilities in the IoT Reference Model, Additional IoT Reference Models, A Simplified IoT Architecture. The Core IoT Functional Stack : Layer 1-3, Analytics Versus Control Applications, Data Versus Network Analytics Data Analytics Versus Business Benefits, Smart Services. **IoT Data Management and Compute Stack** : Fog Computing, Edge Computing ,The Hierarchy of Edge, Fog, and Cloud **Self-learning Topics :** Brief review of applications of IoT : Connected Roadways, Connected Factory, Smart Connected Buildings, Smart Creatures etc. **(Refer Chapter 2)**	06	CO2
3.	Principles of Connected Devices and Protocols in IoT	RFID and NFC (Near-Field Communication), Bluetooth Low Energy (BLE) roles, LiFi , WPAN std : 802.15 standards : Bluetooth, IEEE 802.15.4, Zigbee, Z-wave, Narrow Band IoT, Internet Protocol and Transmission Control Protocol, 6LoWPAN, WLAN and WAN, IEEE 802.11, Long-range Communication Systems and Protocols : Cellular Connectivity-LTE, LTE-A, LoRa and LoRa WAN. **(Refer Chapter 3)**	08	CO3

Sr. No.	Module	Detailed Content	Hours	CO Mapping
4.	Edge to Cloud Protocol	HTTP, WebSocket, Platforms. HTTP - MQTT - Complex Flows : IoT Patterns : Real-time Clients, MQTT, MQTT-SN, Constrained Application Protocol (CoAP), Streaming Text Oriented Message Protocol (STOMP), Advanced Message Queuing Protocol (AMQP), Comparison of Protocols. **(Refer Chapter 4)**	08	CO4
5.	IoT and Data Analytics	Defining IoT Analytics, IoT Analytics challenges, IoT analytics for the cloud, Strategies to organize Data for IoT Analytics, Linked Analytics Data Sets, Managing Data lakes, The data retention strategy, visualization and Dashboarding - Designing visual analysis for IoT data, creating a dashboard, creating and visualizing alerts. **Self-learning Topics** : AWS and Hadoop Technology. **(Refer Chapter 5)**	06	CO5
6.	IoT Application Design	Prototyping for IoT and M2M, Case study related to : Home Automation (Smart lighting, Home intrusion detection), Cities (Smart Parking), Environment (Weather monitoring, weather reporting Bot, Air pollution monitoring, Forest fire detection, Agriculture (Smart irrigation), Smart Library. Introduction to I-IoT, Use cases of the I-IoT,IoT and I-IoT - similarities and differences, Introduction to Internet of Behavior (IoB). **Self-learning Topics** : Internet of Behaviors (IoB) and its role in customer services. **(Refer Chapter 6)**	04	CO6

Assessment

Internal Assessment (IA) for 20 Marks

IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format

- Question Paper will comprise of a total of **six questions each carrying 20 marks Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of four questions need to be answered.

❑❑❑

Index

➢ **MODULE 1**

▶ Chapter 1 : Introduction to Internet of Things ..... 1-1 to 1-32

➢ **MODULE 2**

▶ Chapter 2 : IoT Architecture ..... 2-1 to 2-41

➢ **MODULE 3**

▶ Chapter 3 : Principles of Connected Devices and Protocols in IoT... 3-1 to 3-30

➢ **MODULE 4**

▶ Chapter 4 : Edge to Cloud Protocol ..... 4-1 to 4-29

➢ **MODULE 5**

▶ Chapter 5 : IoT and Data Analytics ..... 5-1 to 5-85

➢ **MODULE 6**

▶ Chapter 6 : IoT Application Design ..... 6-1 to 6-26

❑❑❑

MODULE 1

Introduction to Internet of Things

Syllabus

Introduction to IoT- Defining IoT, Characteristics of IoT, Conceptual Framework of IoT, Physical design of IoT, Logical design of IoT, Functional blocks of IoT, Brief review of applications of IoT, Smart Object – Definition, Characteristics and Trends

Self-learning Topics : Hardware and software development tools for - Arduino, NodeMCU, ESP32, Raspberry Pi, for implementing internet of things, Simulators-Circuit.io, Eagle, Tinkercad.

1.0 Introduction of Internet of Things (IoT) 1-3

1.1 Defining IoT 1-3

1.2 Characteristics of IoT 1-4

1.3 Conceptualized framework of IoT 1-5

1.3.1 Physical Object + Controller, Sensor and Actuators + Internet = Internet of Things 1-5

1.3.2 Gather + Enrich + Stream + Manage + Acquire + organize and Analyze = Internet of Things 1-5

1.3.3 Gather + Consolidate + Connect + Collect + Assemble + Manage and Analyse = Internet of Things 1-6

1.4 Physical design of IoT 1-8

1.4.1 Things of IoT 1-8

1.4.2 IoT Protocols 1-10

1.5 Logical Design of IoT 1-15

1.5.1 IoT Functional Blocks .....
1.5.2 IoT Communication Model.....
1.5.3 IoT Communication Application Programmable Interface (API).....
1.6 Brief review of applications of IoT .....
1.7 Smart Object.....
1.7.1 Definition.....
1.7.2 Characteristics.....
1.7.3 Trends in Smart Objects.....
1.8 Self-learning Topics : Hardware and software development tools .....
1.8.1 Arduino .....
1.8.2 Raspberry Pi.....
1.8.3 Simulators-Circuit.io .....
1.8.4 NodeMCU.....
1.8.5 ESP32 .....
1.8.6 Eagle .....
1.8.7 Tinkercad.....
1.9 Multiple Choice Questions.....
- **Chapter End**... .....

1.0 INTRODUCTION OF INTERNET OF THINGS (IOT)

Internet of Things (IoT) is fully networked and connected devices sending analytics data back to the cloud or data center. It is a network in which each object or thing has a unique identification number, and data is transmitted over a network without verbal confrontation. Innovations in sensor networks, mobile devices, wireless communications, networking, and cloud technologies are driving IoT. It has a much broader application than simply connecting devices to the internet; it also enables data processing, communication, and control through the use of applications.

Conceptualization of IoT can be made from Data-Information-Knowledge. Data doesn't have significance until this is contextualized, processed, and transformed into useful information and knowledge inference. Consider the raw sensor measurements produced by a weather monitoring station ((79,44) ; (88,34) ;...), for instance, Fig. 1.1.1. On its own, this information is meaningless. However, the data becomes meaningful when we add context (information), such as the fact that 44 is humidity and 79 is temperature. We can infer knowledge by observing the data/information and processing it. The average temperature over the previous five minutes can be deduced, for instance, by looking at the temperature of the last five tuples. Based on this knowledge, actions can be taken, such as setting off an alarm if the temperature rises above 80.

(1A1)**Fig. 1.1.1 : IoT conceptualization to Data, Information and Knowledge**

1.1 DEFINING IOT

Formal Definition

- A **dynamic global network** infrastructure with **self-configuring capabilities** based on standard and **interoperable communication protocols** where physical and virtual "things" **have identities**, physical attributes, and visual personalities and use intelligent interfaces and are seamlessly **integrated into information networks** that communicate data associated with users and their environments.

Informal Definition

- Informally, IoT is the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment.
- As with computers, tablets, and smartphones, the term "Internet of Things" refers to a network of physical objects that send, receive, or communicate information using the Internet or other communication technologies and networks, allowing for the monitoring, coordinating, or control of processes across the Internet or another data network
- According to another source, the phrase "IoT" means: A network of actual physical objects, or "things," that are equipped with electronics, software, sensors, and connectivity in order to enhance their functionality and value by exchanging data with other connected devices, their manufacturer, and/or their operator. Despite being able to communicate with one another within the current Internet infrastructure, each object has an embedded computing system that enables it to be uniquely identified.

1.2 CHARACTERISTICS OF IOT

Dynamic Global network & Self-Adapting

- IoT devices should be able to change their behavior dynamically in response to their operating environment, the context of their users, or sensor data environments.
- A camera, for instance, can record information based on the lighting. It automatically switches from day to night mode. This strategy is self-adapting.

Self Configuring

IoT devices may be able to self-configure, enabling many devices to cooperate to provide specific functionality .

Ex : Configure themselves, configure the networking, and download the most recent software updates with least manual aid.

Interoperable Communication Protocols

- IoT Devices support a variety of open standard communication protocols
- They Interact with both infrastructure and other devices
- For example: A smart Phone is able to control the smart TV of different manufacturers.

Unique Identity

- Every IoT device possesses a distinct identity.
- This unique identity can be an IP Address or Uniform Resource Indicator (URI).
- Its identification element is very **useful** if it needs **to access data from a specific device.**
- Along with the control, configuration, and management infrastructure, the IoT device also enables status monitoring, and querying for the user.

Integrated into Information Network

- This allows them to communicate and exchange data with other devices to perform certain analysis.
- This makes the IoT systems **"Smarter".**
- For example: a weaher monitoring node can describe its monitoring capabilities to another connected node, such that they can communicate and exchange data.

1.3 CONCEPTUALIZED FRAMEWORK OF IOT

An IoT Conceptualized framework can be given in terms of three expressions :

1.3.1 Physical Object + Controller, Sensor and Actuators + Internet = Internet of Things

- This is defined by Adrian McEwen and Hakim Cassimally. This equation is a simple conceptualization of a framework for IoT with connectivity to a web service.
- The Internet of Things (IoT) is a network of physical devices and things that, in general, allows a number of objects to collect data from distant locations and communicate with units monitoring, acquiring, organising, and analysing the data for processes and services.

1.3.2 Gather + Enrich + Stream + Manage + Acquire + organize and Analyze = Internet of Things

This is an Equation based on Oracle IoT Architecture with connectivity to data centre, application or enterprise server for data storage, services and business process as shown in Fig. 1.3.1.

Following are the steps :

1. At level 1, data is gathered via the internet or from objects (things) using sensors.
2. A sensor that is linked to a gateway serves as a smart sensor (smart sensor refers to a sensor with computing and communication capacity). At level 2, the data is then enhanced, perhaps through transcoding at the gateway. Coding or decoding is referred to as transcoding when data is transferred between two entities.
3. At level 3, a communication management subsystem transmits or receives data streams.
4. At level 4, data from the device is received by the subsystems for device management, identity management, and access management.
5. At level 5, data is acquired by a data store or database.
6. At level 6, data from the devices and things is organized and analyzed. Data analysis is used, for instance, in corporate processes to gather business intelligence.

(1A3)**Fig. 1.3.1 : Oracle IoT Architecture**

1.3.3 Gather + Consolidate + Connect + Collect + Assemble + Manage and Analyse = Internet of Things

This is an Equation based on the IBM Framework. This concepts consists of a number of subsystems. The data is acquired at remote locations in a database or data store. The services and processes need data managing, acquiring, organising and analysing as shown in Fig. 1.3.2.

NOTES

(1A3)**Fig. 1.3.2 : IBM IoT Conceptual Framework**

The steps are as follows :

1. Levels 1 and 2 consist of a sensor network to gather and consolidate the data. First level gathers the data of the things (devices) using sensors circuits. The sensor connects to a gateway. Data then consolidates at the second level, for example, transformation at the gateway at level 2.
2. The gateway at level 2 communicates the data streams between levels 2 and 3. The system uses a communication-management subsystem at level 3.
3. An information service consists of connect, collect, assemble and manage subsystems at levels 3 and 4. The services render from level 4.
4. Real time series analysis, data analytics and intelligence subsystems are also at levels 4 and 5. A cloud infrastructure, a data store or database acquires the data at level 5.

Various conceptual frameworks of IoT find number of applications including the ones in M2M communication networks, wearable devices, city lighting, security and surveillance and home automation. Smart systems use the things (nodes) which consist of smart devices, smart objects and smart services. Smart systems use the user interfaces (UIs), application programming interfaces (APIs), identification data, sensor data and communication ports.

1.4 PHYSICAL DESIGN OF IOT

1.4.1 Things of IoT

- The term "Things" in the IoT typically refers to IoT components with distinct identities and capabilities for remote sensing, acting, and monitoring.
- Depending on time and space constraints, IoT devices can exchange data (directly or indirectly) with other connected devices and applications, collect data from other devices and process it locally, or send it to centralized servers or cloud-based applications back ends for processing (ie : Memory, processing calibrators, communication latencies and speed and deadlines).
- An IoT device connects various devices using wired or wireless technology.
- These incorporate :

  I) Sensor IoT interfaces,

  II) Internet connectivity interfaces

  III) Interfaces for memory and storage

  IV) Audio video interfaces.
- Temperature, humidity, and light intensity are just a few examples of the several sorts of data that an IoT device might gather from its internal or external sensors.
- IoT devices can also come in a variety of forms, such as smart watches, LED light vehicles, wearable sensors, and industrial machinery.

- Fig. 1.4.1 shows the block diagram of an IoT Device

(1A4)Fig. 1.4.1 : Generic block diagram of an IoT Devices

(1) **Connectivity :** Connectivity between the devices and the server is provided through tools like USB hosts and ETHERNET.

(2) **Processor :** Processor is processes the captured data.

(3) **Audio and Video Interfaces :** High-Definition Multimedia Interface (HDMI) and Radio Corporation of America (RCA) cables act as device interfaces used to record audio and videos in a system.

(4) **Input/Output Interface :** I/O interfaces are used for sensors and actuators. There are various I/O interfaces like

- Universal Asynchronous Receiver-Transmitter (UART) is one of the simplest and oldest forms of device-to-device digital communication
- Serial Peripheral Interface (SPI) is a synchronous serial communication interface specification used for short-distance communication, primarily in embedded systems.
- Inter-IC, a type of bus designed by Philips Semiconductors in the early 1980s, which is used to connect integrated circuits (ICs).
- A Controller Area Network (CAN bus) is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer.

(5) **Storage Interfaces** : Memory devices like Secured Digital (SD), Multimedia Card (MMC), Secured Digital Input Output (SDIO) are used to store the sensed data from IoT sensors.

IoT devices come in a variety of shapes and sizes, including wearable sensors, smart watches, LED light vehicles, and industrial machinery. Almost all IoT devices produce data of some kind, which when processed by data analytics systems yields useful information to guide further action locally or remotely.

1.4.2 IoT Protocols

- These protocols enable communication via the internet between a node device and a server. It is beneficial to use the internet to communicate with IoT devices and get data from them.
- We employ a variety of protocols, some of which exist on both the server and client sides and are controlled by several network layers, including the application, transport, network, and link layers. Fig. 1.4.2 shows various IoT Protocols.

(1A5)**Fig. 1.4.2 : IoT Protocols**

- IoT protocols are divided in four layers namely Link Layer, Network Layer, Transport Layer and the Application Layer

1. Link Layers

Link Layer protocols control the physical transmission of data over the network's physical layer or media (example copper wire, electrical cable, or radio wave).

It also controls the devices' coding and signaling of the packets.

a) 802.3 Ethernet

- It is a collection of technologies and protocols mostly used in LANs. For wired Ethernet networks, it specifies the physical layer and the media access control.
- Various standards of ethernet and their sharing medium are mentioned in Table 1.4.1.

Table 1.4.1 : Ethernet Standards

Sr. No.	Standard	Shared Medium
1.	802.3	Coaxial Cable (10BASE5)
2.	802.3.i	Copper Twisted Pair (10BASE-T)
3.	802.3.j	Fiber Optic (10BASE-F)
4.	802.3.ae	Fiber (10gbps)

b) 802.11 Wi-Fi

- It is a set of LAN protocols that outlines the physical layer and media access control protocols needed to construct wireless local area networks.
- Various standards of ethernet are mentioned in Table 1.4.2.

Table 1.4.2 : Wi-fi Standards

Sr. No.	Standard	Operates in
1.	8011.a	5 Ghz Band
2.	802.11.b and 802.11g	2.4 Ghz Band
3.	802.11.n	2.4/5 Ghz Bands
4.	802.11.ac	5 Ghz Band
5.	802.11.ad	60 Ghz Band

c) 802.11 Wi-Max

It is a set of wireless broadband standards mentioned in Table 1.4.3. Wi-Max standards provide data rates from 1.5 Mbps to 1Gbps.

Table 1.4.3 : Wi-Max Standards

Sr. No.	Standard	Operates in
1.	802.16m	100 Mbps for Mobile Station and 1 Gbps for fixed stations

d) 802.15.4 LR-WPAN

- It is a collection of standards for low-rate wireless personal area networks (LR-WPAN).
- It is the basis for high-level protocols such as ZigBee.
- It supports data rates from 40-250 kbps. It provides low-cost and low-speed communication for power constrained devices.

e) 2G/3G/4G - Mobile Communication

- The many mobile communication standards, including second generation (2G), are listed here GSM and CDMA, third generation (3G), includes UMTS and CDMA2000 and Fourth generation (4G), includes LTE.
- Data rates of these standards range from 9.6 (2G) Kbps to 100 Mbps (4G).

2. Network Layer

This layer sends IP Datagrams from the source network to the destination network.

IPv4 and IPv6 protocols are used to identify hosts while sending data packets.

a) Internet Protocol Version 4 (IPv4):

- Each device connected to the network is given a protocol address, which is a specific numerical label.
- An IP address serves two key purposes: addressing hosts and locations.
- An IP address for IPv4 is 32 bits long that allows a total of 2^{32} addresses.

b) Internet Protocol Version 6 (IPv6)

It is the newest version and an IPv4 successor that has an IP address length of 128 bits.

c) IPv6 Low Power Wireless Personal Area Network (6LoWPAN)

- Low-power devices with little computing capacity can now use IPv6 through low-power wireless personal area networks.
- These networks operate in the 2.4 GHz band and have data transfer rates as low as 50 kb/s.

3. Transport Layer

End-to-end message transfer is made possible by the Transport layer protocols, independent of the underlying network. This layer is used to manage error control, flow control, segmentation and congestion control.

On connections, the message transfer functionality can be configured with (TCP) or without (UDP) handshake acknowledgements

a) Transmission Control Protocol (TCP)

- This is the most widely used protocol for web-browsers along with hypertext transfer protocols (HTTP), HTTPS Application layer protocols, email programs (SMTP) and file transfer protocol (FTP).
- While IP protocol deals with delivering packets, TCP enables consistent transmissions of packets in the correct order. TCP is a connection-oriented and stateful protocol.
- TCP also has the capacity to deduce errors, allowing duplicate packets to be rejected and low-quality packets to be retransmitted. Flow control maintains the data rates of senders and receivers.

b) User Datagram Protocol (UDP)

- UDP is a connectionless protocol and dont need any connection setup.
- This is transaction oriented and stateless protocol. UDP does not offer guaranteed delivery, message ordering, or duplicate removal.

4. Application Layer

The protocols at the application layer specify how an application interacts with those at lower layers to transmit data over a network.

The data is often stored in files, is encoded by the application layer protocol and encapsulated in the transport layer protocol. Process-to-process connections are made possible via application layer protocol employing ports.

a) Hypertext Transfer Protocol (HTTP)

- This protocol forms the foundation of the World Wide Web (WWW) and to transmit the media documents.
- This includes commands such as GET, PUT, POST, DELETE, HEAD, TRACE, OPTIONS etc.
- This follows a request-response model where a request is submits to a server and a client waits for a response.
- The server does not store any information between queries so it is also known as stateless protocol.
- The Universal Resource Indicator (URI) is used by HTTP Protocols to identify HTTP resources.

b) Constrained Application Protocol (CoAP)

- This protocol is used for Machine to Machine (M2M) applications and designed for limited environments with constrained devices and networks.
- CoAP is a web transmission protocol similar to http and has a request-response format; however, it runs on top of UDP rather than TCP.

- CoAP employs a client-server architecture in which clients and servers communicate via connectionless datagrams.
- CoAP supports methods such as GET,POST, PUT and DELETE.

c) **WebSocket**

- This Protocol allows full-duplex communication over a single socket.
- It is based on TCP.
- Here, clients can be a browser, IoT device or mobile application.

d) **Message Queue Telemetry Transport (MQTT)**

- It is a compact messaging protocol built on the public-subscribe model.
- In MQTT, clients, such as Internet of Things (IoT) devices, connect to the server, also known as the MQTT broker, and publish messages to topics on the server.
- The broker forwards the message to the clients subscribed to the topic.
- MQTT is well suited for constrained environments where devices have limited processing, low memory and n/w bandwith requirement

e) **Extensible Messaging and Presence Protocol (XMPP)**

- It is a protocol for XML data streaming and real-time communication between network elements.
- Various applications, including messaging, presence, data syndication, multiparty gaming chat, and voice/voice conversations, are powered by XMPP.
- With XMPP, you may instantly exchange short XML data snippets from one network object to another.
- Both client-server and server-client communication paths are supported by XMPP.
- It is a decentralized protocol.
- Example: Due to the XMPP protocol, in WhatsApp conversation, when a message is viewed by the recipient, a blue tick displayed.

f) **Data Distribution Service (DDS)**

- It is a data-centric middleware standard for device-to-device or machine-to-machine communication.
- DDS is a publish-subscribe model where publishers create topics to which subscribers can use.
- Provides Quality-of-service control and configurable reliability.

g) **Advanced Message Queuing Protocols (AMQP)**

- It is used for business messaging.
- It supports both point-to-point and publisher/subscriber models, routing and queuing.
- Here, brokers receives the messages from publishers and route them over connections to consumers through messaging queues.

Table 1.4.3 summaizes the IoT Protocols.

Table 1.4.3 : Wi-Max Standards

Parameters	HTTP	CoAP	XMPP	DDS	AMQP	MQTT
Protocols	TCP	UDP	TCP	TCP and UDP	TCP	TCP
Network Layer	IP	6LowPAN	IP	IP	IP	IP
Architecture	Client-Server	Client-Server and Publish-Subcribe	Client-Server and Publish-Subcribe	Publish-Subcribe	Client-Server	Publish-Subcribe
Synchronization	Needed	Not Needed	Needed	Sometimes Needed, Sometimes Not	Needed	Needed
Designed for	Internet	IoT/M2M	IoT/M2M	Realtime Systems	M2M	IoT/M2M
Application	World Wise Web (WWW)	Retrieving sensor data	WhatsApp, Gaming, Google Talk	Volkswagen, Smart Cars for video assistants	Google Cloud	Facebook Messanger

1.5 LOGICAL DESIGN OF IOT

- The logical design of an IoT system refers to an abstract representation of entities and processes without going into the low-level specifications of implementation.
- It uses Functional Blocks, Communication Models, and Communication APIs to implement a system.

1.5.1 IoT Functional Blocks

An IoT system is made up of several functional building elements that provide the system the ability to identify, sense, act, communicate, and manage as shown in Fig. 1.5.1.

(1A6)**Fig 1.5.1 : Functional Blocks of IoT**

These functional blocks are explained as follows :

(1) **Device :** These devices are used to provide sensing and monitoring control functions that gather information from the outside world.

(2) **Communication :** This block takes care of the communication of the IoT System.

(3) **Services :** It offers some services, like controlling and monitoring a device, publishing and erasing data, and system restoration.

(4) **Management :** It supports various functions to manage the IoT System.

(5) **Security :** This block is used to safeguard the IoT System by offering various functions such as authentication, authorization, message and content integrity, and data security.

(6) **Application :** It is an interface that offers a control system for the users to check the system status and analyze the processed data.

1.5.2 IoT Communication Model

Communication Models determine the mechanism or the manner in which the data is exchanged or transfers between various devices in IoT Network.

There are four types of IoT Communication Models that are used for data exchange in the IoT Network as follows :

1.	Request-Response	2.	Publish-Subscribe
3.	Push-Pull	4.	Exclusive Pair

1. Request-Response

- This paradigm is a communication model in which a client requests data from a server, and the server provides the requested data.
- When a server receives a request, it fetches the requested information, gathers it, creates the response, and delivers it back to the client.
- Each request-response pair is independent of the others as this is a stateless communication model.
- Fig. 1.5.2 shows the Client-Server interaction in the request-response model.

(1A7)**Fig 1.5.2 : Request-response communication model**

2. Publish-Subscribe

- Publish–Subscribe is a communication model that involves publishers, brokers and consumers.
- Publishers are the source of data. Publishers send the data to the topics which are managed by the broker. Publishers are not aware of the consumers.
- Consumers subscribe to the topics which are managed by the broker.
- When the broker receives data for a topic from the publisher, it sends the data to all the subscribed consumers.
- Fig. 1.5.3 shows the Publish-Subscribe communication model.

(1A8)**Fig 1.5.3 : Publish-Subscribe communication model**

3. Push-Pull

- In this model publishers push the data in the queue and this pushed data is pulled from the queue by consumers as shown in Fig. 1.5.4.

(1A9)**Fig 1.5.4 : Push-pull communication model**

- Here, publishers do not need to know about the consumers.
- Queues help in decoupling the messaging between the producers and consumers.
- Additionally, queues function as a buffer that is useful when there is a discrepancy between the rates at which data is produced and consumed.

4. Exclusive Pair

- It uses a persistent connection between the client and server and is a bidirectional full duplex communication model.
- Once the connection is set up it remains open until the client sends a request to close the connection.
- After connection setup, client and server can send the messages to each other as shown in Fig. 1.5.5.

(1A10)Fig 1.5.5 : Exclusive Pair communication model

1.5.3 IoT Communication Application Programmable Interface (API)

In IoT, several APIs, including REST and WebSocket, are used for server-to-system communication.

Representational State Transfer (REST)-based communication APIs

- REST API uses a set of architectural principles used to design web services and web APIs as shown in Fig. 1.5.6.
- These APIs focus on the systems' resources and how the resouce states are addressed and transferred.
- These APIs follows request-response communication model. This API uses some architectural constraints as follows :

  (1) **Client-Server** : A client and server must be in communication, with the client's user interface being independent of the server's data storage.

  (2) **Stateless :** Each request made by a client to a server must include all the required information and the server must not bother about the client.

  (3) **Cache-able** : A response should be declared as cacheable or not. If set then another client cache is given the rights to reuse that response data for later.

(4) **Layered System** : In this case, the client making the request need not be aware of whether it is speaking with the real server, a proxy, or any other middleman.

(5) **Uniform Interface** : The mechanism of communication between the client and server must comply with standard interface requirements.

(6) **Code on demand** : REST provides for the extension of client capability through the downloading and executing of scripts or applets.

- REST architectural constraints apply to the components, connectors and data elements within a distributed hypermedia system.

(1A11)**Fig. 1.5.6 : Communication with REST APIs**

WebSocket based communication API

- The use of WebSocket APIs enables full-duplex, bi-directional communication between clients and servers.
- The exclusive pair communication model is used by WebSocket APIs, as seen in Fig. 1.5.7.
- Here, the client requests to set up the WebSocket connection and the server responds to the request.
- After a successful connection, the client sends the data frames to the server.
- Finally the connection close request is sent by the client and server to close the connection.

(1A12)**Fig 1.5.7 : Exclusive pair model used by Websocket APIs**

Websocket Vs REST

Comparison based on	REST	Websocket
State	Stateless	Stateful
Directional	Unidirectional	Bidirectional
Req-Res/Full Duplex	Request-Response Model	Exclusive Pair Model
Header Overhead	Each request carries HTTP Headers, hence not suitable for real-time	Does not involve overhead of headers
Scalability	Both horizontal and vertical are easier	Only vertical is easier

Physical Design Vs Logical Design

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

1.6 BRIEF REVIEW OF APPLICATIONS OF IOT

Various applications of IoT including home automation, health and fitness tracking, environmental protection, smart cities, energy system, retail, logistics, agriculture and industrial settings as shown in Fig. 1.6.1.

(1A13)**Fig. 1.6.1 : IoT Applications**

(1) **Home** : Various applications are built using IoT such as Smart lighting, smart appliances, Intrusion detection, Smoke/Gas detection etc.

a) **Smart Lightning**

This helps in Saving energy by adapting the lightning to the ambient conditions and switching on/off or dimming the lights when needed.

b) **Smart Appliances**

- There are many appliances in modern homes, including TVs, refrigerators, audio systems, washers and dryers, etc.
- Each appliance has its own controls or remote controls, making it difficult to manage and regulate them.
- Smart appliances simplify management and give users remote access to status information.
- Examples of smart watches and dryers that may be operated remotely and send alerts when a cycle of washing or driving is completed. Or, when an item is running low on stock, a smart refrigerator may keep track of what is being stored and send the user an update.

c) **Home Intrusion**

- Security cameras, PIR sensors, and door sensors are utilized in home intrusion detection systems to detect intrusion and sound the alarm.
- A user may receive alerts in the form of an email or an SMS.

d) **Smoke Gas Detector**

- In order to detect smoke, which is frequently an early symptom of fire, smoke detectors are mounted in homes and buildings.
- Smoke detector warnings can include messages to the fire alarm system.
- The presence of dangerous gases can be detected with gas detectors.

(2) **Cities** : For cities, IoT has various applications such as smart parking, smart roads, structural health monitoring, emergency responses etc.

a) **Smart Parking:**

- Smart parking is supported by Internet of Things (IoT) systems that count the number of open parking spaces and transmit the data to the applications' back ends through the internet.
- Drivers can use these applications via their cellphones, tablets, and in-car navigation systems.

b) **Smart Lighting**

- Energy-saving smart lighting systems for buildings, parks, and roads are possible.
- Smart lighting enables dynamic lighting control and environment-specific lighting adaptation.
- Remote configuration of lighting schedules and intensity is possible with smart lights connected to the internet.

c) **Smart Road**

- Sensor-equipped "smart" roads can offer information on traffic conditions, travel time estimates, and alarms for dangerous road conditions, heavy traffic, and accidents.
- Such knowledge can improve road safety and lessen traffic congestion.

d) **Structural Health Monitoring**

- A network of sensors is used by structural health monitoring systems to track the vibration levels in buildings and bridges.
- It is possible to pinpoint the damage to a structure, find cracks and mechanical breakdown, as well as determine the structure's remaining life, by evaluating the data.
- By using such methods, a structure's impending failure can be forewarned of in advance.

e) **Surveillance**

- To ensure safety and security, surveillance of public transportation, infrastructure, and even cities is necessary.

- It is possible to build a city-wide surveillance system made up of numerous distributed, internet-connected video surveillance cameras.

f) **Emergency Response**

- Cities' vital infrastructure, including buildings, gas and water pipelines, public transportation, and power substation systems, can be monitored via IoT technologies.
- IoT technologies can assist in producing alerts and limiting their consequences on the vital infrastructure for fire, gas, and water leak directions.

(3) **Environment** : It has applications consisting of weather monitoring, air and noise pollution, forest fire detection, river flood detection etc.

a) **Weather Monitoring**

- IoT-based weather monitoring systems can transfer data to cloud-based applications and storage back-ends after collecting data from a variety of associated sensors, such as temperature, humidity, pressure, etc.

b) **Air pollution surveillance**

- Using gases and dermatological sensors, IoT-based air pollution monitoring systems can track dangerous gas emissions (CO_2, CO, NO, and NO_2) from facilities and autos.
- Making educated decisions on pollution control requires analysis of the collected data.

c) **Noise Pollution Monitoring**

- Stress and disturbed sleep are two health risks that can result from noise pollution in people.
- Monitoring noise pollution can be used to create noise maps for cities.
- Urban planners can use maps of urban noise to aid in regulating noise levels around residential areas, schools, and parks.
- The IoT-based smart metering system for noise pollution makes use of several noise monitoring stations that are placed around a metropolis.

d) **Forest Fire Detection**

- Various factors, such as lightning, human error, volcanic eruptions, and sparks from rock falls, can trigger forest fires.
- Forest fires that are detected early can have their impact reduced.
- An early warning system for forest fires gives possibility of forest fires a head start.

e) **River Flood Detection**

- A river flood monitoring system is described that detects river and weather conditions using wireless sensor nodes fitted with various sensors.
- Monitoring applications raise alert when a rapid increase in water level and flow rate is noticed.

- The systems contain a water level monitoring module and a data processing module that give raw data, forecasted data, and video feeds for flood information.

(4) **Energy** : Smart grid, renewable energy system, prognostics etc. included in this category.

a) **Smart Grid**

- The term "smart grid" refers to an electrical grid-integrated data communication network that gathers and analyses real-time data on electricity transmission, distribution, and consumption.
- Electricity generation (centralised or dispersed models) storage (or conversion of the energy into various forms), distributions, and equipment health data are all collected by smart grids.

b) **Smart Meters**

- Smart metres have the ability to remotely control electricity generation and consumption, as well as remotely turn off the supply when necessary.
- Smart metres can stop power thefts by evaluating data on power generation, transmission, and consumption.

c) **Prognostics**

- By assessing the degree of deviations from the system's typical operating patterns, IoT-based prognostic real-time health management systems may forecast the performance of energy or mechanical systems.

(5) **Retail** : Some of the applications comprises smart payments, inventory management, smart vending machines etc.

a) **Inventory Management**

- A radio frequency identification Internet of Things system RFID tags can be used to control inventory and keep the appropriate amounts of inventory.
- The products can be tracked in real time thanks to RFID tags that are attached to them, allowing inventory levels to be precisely calculated and low-stock items to be restocked.
- RFID scanners placed in the warehouse or on the shelves of retail establishments can be used for tracking.

b) **Smart Payments**

- Contact list payments using smart payment solutions are enabled by Bluetooth and NFC technologies.
- Smartphones and other gadgets can communicate with one another using a set of standards called near field communication by touching or bringing them close to one another.

c) **Smart Vending Machines**

- Smart vending machines with internet connections enable contactless payments, promotions, and remote monitoring of inventory levels.
- Users can record their preferences and preferred products in smart vending machines, which employ sensors to track their operation and send data to the cloud for preventative maintenance.
- Smart vending machines can share their inventory levels and connect with other vending machines nearby so that customers can be directed to the closest units in the event that a product runs out in one machine.

(6) **Logistics** : Applications in this category are route generation and scheduling, fleet tracking, shipment monitoring, remote vehicle tracking/diagnosting etc.

a) **Route Generation and Scheduling**

- The availability of vehicles allows the transportation system to offer new services like advanced route guidance and dynamic vehicle routing that anticipate customer demand for pickup and delivery issues.
- Another example is route generation and scheduling systems that are candidate end-to-end using combinations of road patterns and transportation smooth and feasible schedule.

b) **Fleet Monitoring**

- GPS technology is used by vehicle fleet monitoring systems to track the locations of vehicles in real time.
- When planned routes are diverted, alerts may be generated.
- The data on vehicle locations and routes can be combined and evaluated to find supply chain bottlenecks caused by things like traffic conditions, route of various supply chain parts, and alternate route generations.
- The system may examine messages provided from the cars to identify unexpected occurrences

c) **Shipment Monitoring**

- Transportation systems can monitor the conditions inside containers via shipment monitoring solutions.
- To avoid food deterioration, for instance, containers containing fresh vegetables might be watched.

d) **Remote Vehicle Diagnostic**

- Remote car diagnostic devices can identify vehicle defects and alert the driver to potential problems.
- These diagnostic systems integrate on-board diagnostic systems with IoT devices to collect data on vehicle operation, including speed, engine RPM, coolant temperature, fault code number, and status of the various vehicle subsystems.

(7) **Agriculture** : Several applications of agriculture are smart irrigation, greenhouse control etc.

a) **Smart Irrigation**

- Smart irrigation techniques can increase crop productivity.
- Smart irrigation systems use IoT devices with soil moisture sensors to
- Measure the soil's moisture content and only allow water to flow via irrigation pipes when the level of moisture falls below a predetermined threshold.
- Measurements of the moisture level are also collected by smart irrigation systems and stored on a computer or in the cloud, where the data is evaluated to determine when to water plants.

b) **Green House Control**

- To offer the greatest circumstances for plant growth, the climatological conditions inside a greenhouse can be monitored and managed.
- Sensors are used to measure the levels of temperature, humidity, soil moisture, light, and carbon dioxide, and actuation devices are used to automatically change these variables' climatological conditions.
- For better administration and maintenance of agricultural produce, the system uses a wireless sensor network to track and adjust agricultural characteristics including temperature and humidity in real-time.

(8) **Industry** : Numerous applications are included in this such as machine diagnosis and prognosis, indoor air quality monitoring etc.

a) **Machine Diagnosis and Prognosis**

- Machine prognosis is the process of estimating a machine's performance based on data analysis, present operating conditions, and the degree to which they deviate from ideal operating conditions.
- Finding the root causes of a machine fault is referred to as machine diagnostic.
- Machine sensors can keep an eye on things like operational temperatures and vibration levels.

b) **Indoor Air Quality Monitoring**

- Using a variety of gas sensors, an IoT-based gas monitoring system can assist in monitoring the interior air quality.
- For various locales, the quality of the indoor air can change.
- IoT-based wireless sensor networks can identify more dangerous areas so that remedial action can be taken to ensure adequate ventilation.

(9) **Health and Lifestyle** : Various applications such as wearable electronics, health and fitness monitoring etc.

a) **Health and fitness monitoring**

- Wearable Internet of Things (IoT) devices that are low-impact and continuously monitor physiological data can aid in continuous health and fitness monitoring.
- These accessories may be integrated into wristbands or take other shapes.

b) **Wearable Electronics**

- Wearable Electronics such as smart watches smart glasses wristband and fashion electronics (with electronic integrated in clothing and accessories, example Google glass for Moto 360 smart watches provide various functions and future to assist us in our daily activities and making us lead healthy Lifestyle.
- Smart watches allow users to search the internet, play audio and video files, make calls with or without paired mobile phones, play games, and use a variety of mobile applications.
- Smart watches enable users to take pictures and record videos, get directions, check flight status, and perform voice searches on the internet.

1.7 SMART OBJECT

1.7.1 Definition

Smart objects are any physical things with embedded technology that can communicate with one another or an outside agent, detect their surroundings, and/or interact with them in a meaningful way.

1.7.2 Characteristics

There are mainly four characteristics of smart objects described as follows :

1. Processing unit

- This unit acquires data, processes it and analyzes sensing information received by the sensor(s).
- Also sends coordinating control signals to any actuators, and controls a variety of functions on the smart object, including the communication and power systems.
- The most common processing unit is a microcontroller because of its small form factor, flexibility, programming simplicity, ubiquity, low power consumption, and low cost.

2. Sensor(s) and/or actuator(s)

A smart object is capable of interacting with the physical world through sensors and actuators.

3. Communication Unit

- This unit is responsible for connecting a smart object with other smart objects and the outside world (via the network).
- Communication devices for smart objects can be either wired or wireless.

4. Power source

- Smart objects have components that need to be powered.
- The most significant power consumption usually comes from the communication unit of a smart object.

1.7.3 Trends in Smart Objects

Trends are mentioned below :

(1) **Size is decreasing** : Smart objects must be kept as small as possible.

(2) **Power consumption is decreasing** : Issue of batteries is a big challenge.

(3) **Processing power is increasing** : Computing power should be higher.

(4) **Communication capabilities are improving** : High range network support is prime concern.

(5) **Communication is being increasingly standardized** : highly essential.

1.8 SELF-LEARNING TOPICS : HARDWARE AND SOFTWARE DEVELOPMENT TOOLS

IoT development is the process of integrating hardware components and software applications so that the final product may track particular values, gather and send data, evaluate received data, and direct the physical device to respond accordingly.

Such systems are extremely difficult to create. Various development tools of IoT are described below:

1.8.1 Arduino

- Arduino can be a wise choice if you're looking to build a computer that can perceive and control the physical world more effectively than your typical stand-alone computer.
- The open-source IoT prototyping platform Arduino provides the ideal fusion of IoT hardware and software. Interactive electronics can be used with Arduino to a set of hardware requirements.
- The Integrated Development Environment (IDE) and Arduino programming language make up the software of Arduino (IDE).

Software products are represented by :

(1) **Arduino IDE :** An open-source prototyping platform, which can be used to easily write code compatible with any Arduino board.

(2) **Arduino Cloud :** A single platform that enables the wireless communication of IoT devices, as well as their remote control and data collection.

(3) **IoT Cloud Remote :** An application for creating dashboards to control cloud-connected devices.

(4) **Web Editor :** An application for coding from a browser.

1.8.2 Raspberry Pi

- The developers of this IoT IDE are Raspberry Pi board lovers. It is a crucial IoT development tool with over 35,000 packages, numerous examples, quick installation using pre-compiled software, and more.
- It is frequently recognised as the greatest tool for developing IoT apps on the Raspberry Pi.
- Another excellent feature of this programme is that it is constantly being improved and has increased computing's applicability to the benefit of consumers.
- Explore the various sections of the online tutorial "From 0 to 1: Raspberry Pi and the Internet of Things" to learn more about using the Raspberry Pi for home automation.
- You learn about the fundamentals of IoT, the anatomy of the Raspberry Pi, Python programming, physical computing with the Raspberry Pi, and so much more!

1.8.3 Simulators-Circuit.io

- AutoDesk created a circuit/PCB designing tool and simulator known as Simulators-Circuit.io, that allows you to design the circuit, visualize it on the breadboard, use the well-known Arduino platform, simulate the circuit, and ultimately construct the PCB.
- This software simulation can directly be used to programme the Arduino.

Advantages

(1) The output design is easier to interpret and will be a handy reference while making a real life connection.

(2) It can simulate Arduino.

(3) The library is rich and has plenty of parts.

Limitations

(1) Designing circuit is bit tougher than other simulators.

(2) Can't draw a circuit quickly.

1.8.4 NodeMCU

- A cheap open source IoT platform is NodeMCU.
- It originally included hardware based on the ESP-12 module and firmware that runs on Espressif Systems ESP8266 Wi-Fi System on Chip (SoC).
- The name "NodeMCU" combines "node" and "MCU" (micro-controller unit).
- The term "NodeMCU" strictly speaking refers to the firmware rather than the associated development kits.

- It is a self-contained WiFi networking system that can run standalone programmes in addition to acting as a bridge between the current microcontroller and WiFi.
- Especially used to monitor and control things from anywhere in the world.

1.8.5 ESP32

- The developers of the well-known ESP8266 SoC, Espressif Systems, have created the inexpensive ESP32 System on Chip (SoC) Microcontroller.
- The 32-bit Microprocessor replaces the ESP8266 SoC and has built-in Bluetooth and Wi-Fi. It is available in single-core and dual-core versions.
- The advantage of ESP32 is that it has inbuilt RF components such a power amplifier, a low-noise receiver amplifier, an antenna switch, filters, and an RF balun, similar to ESP8266.
- As a result, it is very simple to construct hardware around the ESP32 since minimal external components are needed.

1.8.6 Eagle

- Cloud Based platform to remotely monitor the environmental sensors.
- Eagle.io, a tool created for system integrators and consultants, enables you to transform time-series data into useful insight.
- You may gather data in real time from any data logger or text file, automatically modify it using processing and logic, get alerts for important occurrences, and provide your clients access.
- Because of this, some of the largest companies in the world rely on eagle.io to give them real-time access to information about their natural resources and environmental conditions.

1.8.7 Tinkercad

- Tinkercad is an open source and free web based online collection of software tools.
- Tinkercad provides us easy **3D design creation**, **code blocks programming** and **electronic circuits simulation.**
- In the **Circuits** section of Tinkercad, you can design your circuit in the easiest manner and can **simulate** your circuit using an Arduino or other boards or components available, with and without any knowledge of computer codes or programming.
- In the **3D design** of Tinkercad, you don't need prior knowledge of CAD. It is easy to create 3D designs you like.
- In **code blocks** of Tinkercad, you can design 3D objects not only using drag and drop of shapes but also using visual code blocks.

1.9 MULTIPLE CHOICE QUESTIONS

Q. 1.1 What is IoT?

(a) network of physical objects embedded with sensors

(b) network of virtual objects

(c) network of objects in the ring structure

(d) network of sensors

✓**Ans. : (a)**

Q. 1.2 Which of the following is false about IoT devices?

(a) IoT devices use the internet for collecting and sharing data

(b) IoT devices need microcontrollers

(c) IoT devices use wireless technology

(d) IoT devices are completely safe

✓**Ans. : (d)**

Q. 1.3 Which of the following is not an application of IoT?

(a) BMP280 (b) Smart home

(c) Smart city (d) Self-driven cars

✓**Ans. : (a)**

Q. 1.4 Which of the following is not a fundamental component of an IoT system?

(a) Sensors (b) Connectivity and data processing

(c) User interface (d) Transformer

✓**Ans. : (d)**

Q. 1.5 Which layer is used for wireless connection in IoT devices?

(a) Application layer (b) Network layer

(c) Data link layer (d) Transport layer

✓**Ans. : (c)**

Q. 1.6 Which of the following is used to capture data from the physical world in IoT devices?

(a) Sensors (b) Actuators

(c) Microprocessors (d) Microcontrollers

✓**Ans. : (a)**

Q. 1.7 Which of the following is true about Arduino IoT devices?

(a) They are open-source software(b) They can only read analog inputs

(c) They have their own operating systems(d) They don't have pre-programmed firmware

✓**Ans. : (a)**

Q. 1.8 IoT gateway must provide _____

(a) Protocol abstraction (b) Data storage

(c) Security with hardware (d) Simple and fast installation

✓**Ans. : (a)**

Q. 1.9 Which of the following protocols is used to link all the devices in the IoT?

(a) HTTP (b) UDP (c) Network (d) TCP/IP

✓**Ans. : (d)**

Q. 1.10 What is the component of an IoT system that executes a program?

(a) A sensor (b) A microcontroller

(c) An actuator (d) A digital to analog converter

✓**Ans. : (b)**

Q. 1.11 What is the sensor/protocol used in GSN?

(a) HTTP protocol (b) CoAP protocol

(c) MQTT protocol (d) XMPP protocol

✓**Ans. : (b)**

Q. 1.12 An IoT network is a collection of _____devices.

(a) Signal (b) Machine to Machine

(c) Interconnected (d) Network to Network

✓**Ans. : (c)**

Q. 1.13 _____ allows the user to control electronic components.

(a) Android API (b) RESTful API

(c) MQTT API (d) CoAP API

✓**Ans. : (b)**

Q. 1.14 Which of the following is not an application of IoT?

(a) Wearables (b) Smart Grid

(c) Arduino (d) Smart City

✓**Ans. : (c)**

Q. 1.15 Which of the following protocols does not exist at the data link layer?

(a) ZigBee Smart Energy (b) LoRaWAN

(c) WirelessHART (d) Secure MQTT

✓**Ans. : (d)**

Chapter Ends...

❑❑❑

MODULE 2

IoT Architecture

Syllabus

Drivers Behind New Network Architectures : Scale, Security, Constrained Devices and Networks, Data, Legacy Device Support

Architecture : The IoT World Forum (IoTWF) Standardized Architecture :Layer 1-7, IT and OT Responsibilities in the IoT Reference Model, Additional IoT Reference Models A Simplified IoT Architecture

The Core IoT Functional Stack : Layer 1-3 , Analytics Versus Control Applications , Data Versus Network Analytics Data Analytics Versus Business Benefits , Smart Services,

IoT Data Management and Compute Stack : Fog Computing , Edge Computing ,The Hierarchy of Edge, Fog, and Cloud.

Self-learning Topics : Brief review of applications of IoT: Connected Roadways , Connected Factory, Smart Connected Buildings , Smart Creatures etc.

2.1 Introduction 2-3

2.2 Drivers Behind New Network Architectures 2-3

2.2.1 IoT Architectural Drivers 2-6

2.3 Comparing IoT Architectures 2-7

2.3.1 The oneM2M IoT Standardized Architecture 2-7

2.3.2 The IoT World Forum (IoTWF) Standardized Architecture 2-9

2.3.3 IT(Information Technology and OT(Open Technology) Responsibilities in the IoT Reference Model 2-12

2.3.4 Additional IoT Reference Models 2-15

2.4 A Simplified IoT Model 2-16

2.4.1 The Core IoT Functional Stack 2-18

2.5 Build your own IoT Communication Network 2-19

2.5.1 Layer 1: Things: Sensors and Actuators Layer 2-19

2.5.2 Layer 2: Communications Network Layer 2-21
2.5.2(a) Access Network Sublayer 2-22
2.5.2(b) Gateways and Backhaul Sublayer 2-25
2.5.2(c) Network Transport Sub-layer 2-26
2.5.2(d) IoT Network Management Sub-layer 2-27
2.5.3 Layer 3: Applications and Analytics Layer 2-28
2.6 Analytics Versus Control Applications 2-28
2.6.1 Analytics Application 2-28
2.6.2 Control Application 2-29
2.7 Data Versus Network Analytics 2-29
2.7.1 Data Analytics 2-29
2.7.2 Network Analytics 2-29
2.8 Data Analytics Versus Business Benefits 2-30
2.8.1 Data Analytics and Business Analytics Comparison Table 2-31
2.8.2 Key Differences Between Data Analytics and Business Analytics 2-32
2.9 Smart Services 2-32
2.9.1 Smart Services Use IoT and Aim for Efficiency 2-33
2.9.2 Smart Services in hospitality 2-33
2.9.3 Smart services can be integrated into an IoT system like Smart Home. 2-33
2.9.4 Efficiency can be extended to larger systems Like Smart Grid 2-34
2.9.5 Efficiency also applies to M2M communications. 2-35
2.10 IoT Data Management and Compute Stack 2-36
2.10.1 Several Data-Related Problems Need to be Addressed 2-38
2.10.2 Fog Computing 2-38
2.10.2(A) Characteristic of Fog Computing 2-39
2.10.3 Edge Computing 2-39
2.10.3(a) The Hierarchy of Edge, Fog, and Cloud 2-40
- **Chapter End**... 2-41

2.1 INTRODUCTION

- Internet of Things (IoT) is a system of interrelated, internet-connected objects which are able to collect and transfer data over a wireless network without human intervention. For example, smart fitness bands or watches, driverless cars or drones, smart homes that can be unlocked through smartphones and smart cars, etc.
- Due to the ever-evolving nature of IoT devices, and the wide diversity of sensors, there is no one-size-fits-all architecture for IoT projects. However, some of the building blocks will be similar from project to project.
- First, you will need to build with scalability in mind. The amount of data that you will collect over time will take on enormous proportions and you will need a platform that can accommodate this in the long run.
- You will also need to ensure that you have high availability at any given time. Having system failures could make you lose some business in the best case, or could have fatal consequences in the worst cases.
- Finally, you will need a system that is flexible enough to accommodate quick and frequent changes. As your architecture evolves, or your business needs change, you will need to iterate quickly without breaking the existing architecture.
- Administrators use IoT architecture to manage and support IoT devices. IoT devices can be anything from an internet-connected light bulb to pressure safety sensors in a chemical plant.
- These devices use small sensors to collect data about their environment and send that data to a server for processing. Servers process this data to create information and insights for businesses. Many times this information is used to automate tasks that improve uptime and efficiency across multiple business systems.
- IoT architecture makes this all possible by ensuring data gets where it needs to and is processed correctly. Without proper IoT architecture, networks would become unreliable, defeating the entire purpose of investing in IoT in the first place.

2.2 DRIVERS BEHIND NEW NETWORK ARCHITECTURES

This begins by comparing how using an architectural blueprint to construct a house is similar to the approach we take when designing a network. Take a closer look at some of the differences between IT and IoT networks, with a focus on the IoT requirements that are driving new network architectures, and considers what adjustments are needed. Following are the IoT Architecture drivers

1. Scale,
2. Security,
3. Constrained Devices and Networks
4. Data,
5. Legacy Device Support

(1) Scale

- Generally an IT network is on the scale of order of several thousand devices—typically printers, mobile wireless devices, laptops, servers, and so on.
- The traditional three-layer campus networking model, supporting access, distribution, and core (with sub architectures for WAN, Wi-Fi, data center, etc.), is well understood.
- IoT introduces a model where an average-sized utility, factory, transportation system, or city could easily be asked to support a network of thousand end point to few million end points.
- Based on scale requirements of this order, IPv6 is the natural foundation for the IoT network layer.

(2) Security

- Already today machine and networks are targeted with malicious attacks using vulnerabilities in networked machines,
- Brute force attack , ransomware attack and many such which may effect the systems network badly. Even if there is world III then it amu be due to security in CyberSpace.
- The frequency and impact of cyber attacks in recent years has increased highly. This is now need of time and Responsibility of IT department for Protecting corporate data from intrusion and theft.
- IT departments also to protect servers, applications, and the network, setting up defense-in-depth models with layers of security designed to protect the cyber crown jewels of the corporation.
- Even if all the efforts taken to protect networks and data, hackers are smart enough to penetrate trusted networks.
- In IT networks, the circumference of defense is often the perimeter firewall. It is very critical to place any endpoint outside the firewall.
- But if the IoT endpoints are located in wireless sensor networks and if they are using unlicensed spectrum and are not only visible to the world and also accessible and widely distributed in the field.
- Traditional models of IT security are can be easily attacked by IoT System. IoT systems require consistent mechanisms of authentication, encryption, and intrusion prevention techniques that understand the behavior of industrial protocols and can respond to attacks on critical infrastructure.
- For optimum security, IoT systems must:
  - Be able to identify and authenticate all entities involved in the IoT service (that is, gateways, endpoint devices, home networks, roaming networks, service platforms)
  - Ensure that all user data shared between the endpoint device and back-end applications is encrypted

- Comply with local data protection legislation so that all data is protected and stored correctly
- Utilize an IoT connectivity management platform and establish rules-based security policies so immediate action can be taken if anomalous behavior is detected from connected devices
- Take a holistic, network-level approach to security

(3) Constrained Devices and Networks

- IoT sensors are designed for a single task, and this sensor are small and inexpensive. They have limited power, CPU, and memory, and they transmit only when there is something important.
- Because of the massive scale of these devices and the large, uncontrolled environments where they are usually deployed, the networks that provide connectivity also tend to be very lossy and support very low data rates.
- On the other and IT networks, which are speeds(Many gigabytes) and endpoints with powerful CPUs. In case if performance constraints, then network can be simply upgraded to a faster network.
- If too many devices are on one VLAN and are impacting performance, simply a new VLAN can be designed and continue to scale as much needed.
- However, this approach cannot meet the constrained nature of IoT systems. IoT requires a new technology of connectivity that meet both the scale and constraint limitations.

(4) Data

- IoT devices/Sensor generate a huge amount of data. In general, most IT shops don't really care much about the unstructured chatty data generated by devices on the network.
- The data generated by IT network is not that important as compared to data generated by the IoT network. The IoT the data can used to enhance the businesses to deliver new IoT services that enhance the customer experience, reduce cost, and deliver new revenue opportunities.
- The IoT-generated data is mostly unstructured, these data is to be processes through analytics, now its needed to create new business models to process this Data.
- Lets take example of smart city with a few hundred thousand smart streetlights , traffic signal and other sensors, all connected through an IoT network. All this information communicated between the network modules and the control centers, data pattern can help us in predicting when lights , signals and sensors need to be replaced or whether they can be turned on or off at certain times, which will save operational expense.
- But when all this data is combined, its very difficult to manage and analyze effectively. IoT systems are designed to move data consumption throughout the architecture, both to filter and reduce unnecessary data going upstream and to provide the fastest possible response to devices when necessary.

(5) Legacy Device Support

- Upgrading system in IT network is easy like OS version, Protocols support can be upgraded either automatically or manually.
- In IoT systems, end devices are likely to be on the network for a very long time sometimes decades. As IoT networks are deployed, they need to support the older devices already present on the network, as well as devices with new capabilities. In many cases, legacy devices are so old that they don't even support IP.
- For example, a factory may replace machines only once every 20 years or perhaps even longer! It does not want to upgrade multi-million-dollar machines just so it can connect them to a network for better visibility and control.
- However, many of these legacy machines might support older protocols, such as serial interfaces, and use RS-232. In this case, the IoT network must either be capable of some type of protocol translation or use a gateway device to connect these legacy endpoints to the IoT network.

2.2.1 IoT Architectural Drivers

Table 2.2.1 : IoT Architectural Drivers

Challenge	Description	IoT Architectural Change Required
Scale	The massive scale of IoT end-points (sensors) is far beyond that of typical IT networks	The IPv4 address space has reached exhaustion and is unable to meet IoT's scalability requirements. Scale can be met only by using IPv6. IT networks continue to use IPv4 through features like Network Address Translation (NAT).
Security	IoT devices, especially those on wireless sensor networks (WSNs), are often physically exposed to the world	Security is required at every level of the IoT network. Every IoT endpoint node on the network must be part of the overall security strategy and must support device-level authentication and link encryption. It must also be easy to deploy with some type of a zero-touch deployment model.
Devices and networks constrained By power, CPU, Memory and Link speed	Due to the massive scale and longer distances, the networks are often constrained, lossy, and capable of supporting only minimal data rates (tens of bps to hundreds of Kbps).	New last-mile wireless technologies are needed to support constrained IoT devices over long distances. The network is also constrained, meaning modifications need to be made to traditional network-layer transport mechanisms.
The massive volume of data generated	The sensors generate a massive amount of data on a daily basis, causing network bottlenecks and slow analytics in the cloud.	Data analytics capabilities need to be distributed throughout the IoT network, from the edge to the cloud. In traditional IT networks, analytics and applications typically run only in the cloud.

Challenge	Description	IoT Architectural Change Required
Support for legacy devices	An IoT network often comprises a collection of modern, IP-capable endpoints as well as legacy, non-IP devices that rely on serial or proprietary protocols.	Digital transformation is a long process that may take many years, and IoT networks need to support protocol translation and/or tunneling mechanisms to support legacy protocols over standards-based protocols, such as Ethernet and IP.
The need for data to be analyzed in real time	Whereas traditional IT networks perform scheduled batch processing of data, IoT data needs to be analyzed and responded to in real-time	Analytics software needs to be positioned closer to the edge and should support real-time streaming analytics. Traditional IT analytics software (such as relational databases or even Hadoop), are better suited to batch-level analytics that occur after the fact.

2.3 COMPARING IOT ARCHITECTURES

- The challenges and requirements of IoT systems have driven a whole new discipline of network architecture. In the past several years, architectural standards and frameworks have emerged to address the challenge of designing massive-scale IoT networks.
- In effort to standardize the rapidly growing field of machine to machine (M2M) communications, the European telecommunications Standards Institute (ETSI) created the M2M Technical committee in 2008.
- The goal of this committee was to create a common architecture that would help accelerate the adoption of M2M applications and devices. Over the time, the scope has expanded to include the Internet of Things. Recognizing this need, in 2012 ETSI and 13 other founding members launched one M2M as a global initiative designed to promote efficient M2M communication systems and IoT.

2.3.1 The oneM2M IoT Standardized Architecture

- The one M2M architecture divides IoT functions into three major domains as shown in Fig. 2.3.1.
  - the application layer,
  - the services layer and
  - the network layer.
- While this architecture is very rich and promotes interoperability through IT-friendly APIs and supports a wide range of IoT technologies. Let's see each of these domains in turn:

1. **Applications Layer**
   - The one M2M architecture gives major attention to connectivity between devices and their applications.
   - This domain includes the application layer protocols and attempts to standardize northbound API definitions for interaction with Business Intelligence (BI) Systems.
   - Applications tend to be industry specific and have their own sets of data models and thus they are shown as vertical entities.
2. **Services Layer**
   - This layer is shown as a horizontal framework across the vertical industry applications.

Fig. 2.3.1 : Main Element of the one M2M IoT architecture

- At this layer, horizontal modules include the physical network that the IoT applications run on, the underlying management protocols, and the hardware.
- Examples include backhaul communications via cellular, MPLS (Multiprotocol label switching) networks, VPNs and so on. Riding on top is the common services layer.
- This conceptual layer adds APIs and middleware supporting third party services and applications.

3. **Network Layer**
   - This is the communication domain for the IoT devices and endpoints.
   - It includes the devices themselves and the communication network that links them. Embodiments of this communication infrastructure includes wireless mesh technologies such as IEEE 802.15.4 and wireless point to multipoint systems IEEE 801.11ah.
   - In many cases, the smart (and sometimes not so smart) devices communicate with each other. In other cases, machine-to-machine communication is not necessary, and the devices communicate through a field area network (FAN) to use case specific apps in the IoT application domain.

- Therefore, the device domain also includes the gateway device, which provides communications up into the core network and acts as a demarcation point between the device and network domains.

2.3.2 The IoT World Forum (IoTWF) Standardized Architecture

- In 2014 the IoTWF architectural committee (led by Cisco, IBM, Rockwell Automation, and others)published a seven-layer IoT architectural reference model shown in Fig.2.3.2.
- There are various IoT reference models but the one proposed by the IoT World Forum offers a clean, simplified perspective on IoT and includes edge computing, data storage, and access.
- It provides an expressed way of visualizing IoT from a technical perspective briefly and clearly.
- Each of the seven layers is broken down into specific functions, and security encompasses the entire model. Fig. 2.3.2. shows details the IoT Reference Model published by the IoTWF.

Fig. 2.3.2 : IoT Reference Model Published by the IoT World Forum

- As shown in Fig. 2.3.2, the IoT Reference Model defines a set of levels with control flowing from the center (this could be either a cloud service or a dedicated data center), to the edge, which includes sensors, devices, machines, and other types of intelligent end nodes.
- In general, data travels up the stack, originating from the edge, and goes northbound to the center.
- Using this reference model, we are able to achieve the following:
  - Decompose the IoT problem into smaller parts
  - Identify different technologies at each layer and how they relate to one another
  - Define a system in which different parts can be provided by different vendors
  - Have a process of defining interfaces that leads to interoperability
  - Define a tiered security model that is enforced at the transition points between levels

The following sections look more closely at each of the seven layers of the IoT Reference Model.

(a) Layer 1 : Physical Devices and Controllers Layer

- The first layer of the IoT Reference Model is the physical devices and controllers layer.
- In this layer connected devices, sensors and machines at the edge with which other devices, sensors and machines or people interact.
- This layer includes the various endpoint devices and sensors that send and receive information and its like home to the "things" in the IoT,.
- There are various size of "things" which ranges from miniature(microscopic) Sensors to big(giant) machines used in factory.
- Generation of data is primary function and also they are capable of being queried and/or controlled over a network.

(b) Layer 2 : Connectivity Layer

- In the second layer of the IoT Reference Model, the focus is on connectivity.
- These include backhaul and wireless systems generally provided by telcos and providers of telecommunications and networking equipment's that enable the connecting of Things to the internet.
- This IoT layer guarantees the reliable and timely transmission of data. This is main and important function of these layer
- Layer2 includes transmissions between Layer 1 devices and the network and between the network and information processing that occurs at Layer 3 (the edge computing layer).
- The connectivity layer hold within all networking elements of IoT and doesn't really distinguish between the last-mile network, gateway, and backhaul networks.
- Fig. 2.3.3 shows in details functions of the connectivity layer

Fig. 2.3.3 : IoT Reference Model Connectivity Layer Functions

(c) Layer 3: Edge Computing Layer

- This layer allows for a highly scalable low latency architecture by enabling computing close to where it is needed.
- Edge computing is often referred to as the "fog" layer.

- **Emphasis**: On data reduction and converting network data flows into information that is ready for storage and processing by higher layers.
- **Basic principles** of this reference model is that information processing is initiated as early and as close to the edge of the network as possible.
- Fig 2.3.4 shows the functions of Layer 3 of the IoT Reference Model.

Fig. 2.3.4 : IoT Reference Model Layer 3 Functions

- Layer 3 is responsible for evaluation of data to see if it can be filtered or aggregated before being sent to a higher layer.
- This also allows for reformatting of data or decoding, making additional processing by other systems easier.
- **Critical function** if the assessing data exceeds predefined thresholds are crossed and any action or alerts need to be sent is taken care by this layer.

Upper Layers: Layers 4–7

- The upper layers deal with handling and processing the IoT data generated by the bottom layer.

(d) Layer 4: Data Accumulation

- This layer allows the storage of captured application data in databases, files, or preferences, in internal or removable storage which can used when needed.
- This layer also converts event based to query based processing

(e) Layer 5: Data Abstraction

- This layer allows the aggregation of data from distributed nodes at the edge and in the cloud
- This layer restores multiple data formats and ensure consistent semantics from various source.
- This layer confirms that the data set is complete and consolidates data into one place or multiple data stores using Virtualization

(f) Layer 6 : Application Layer

- This layer consists of essential tasks and entities required to achieve a business goal Page
- **This layer interprets data using software tools and applications.**
- Applications are used to monitor ,control and provide reports based on the analysis of data.

(g) Layer 7: Collaboration and process

- This brings together and orchestrates applications to achieve a business process
- This layer consumes and share the application information.
- This layer help in collaborating on and communicating IoT information which often requires many steps.
- This layer can changes business process and deliver the benefits of IoT

2.3.3 IT(Information Technology and OT(Open Technology) Responsibilities in the IoT Reference Model

(a) Operational Technology (OT) Network

- OT or Operational technology is a category of a computing system which process operational data such as telecommunication, technical components, computers and it is used to monitor devices, various industrial process and some of the industry events and accordingly make adjustments if required in an industry or enterprise.
- In other words, OT uses the combination of software and hardware which can be used to perform real-time operations to detect if there is any change occurred during the whole process which can be done by directly controlling the industrial equipment and some of the enterprise's events which make them reliable and increase their rate of availability and reliability.
- OT systems ensure the safety of industrial operations by continually monitoring them and also helps to support infrastructure such as manufacturing, defense utilities, etc. OT network works at the industrial level to process the operational data of any organization.

(b) Information Technology (IT) Network

- IT or Information Technology deals with the systems mainly computers and telecommunication for performing various operations like for giving input, for storing, recovering, transmitting, manipulating and protecting data or information so that data can be exchanged among different organizations.
- IT Network encompasses hardware (computers, physical servers, and network equipment), software (operating systems, applications), and peripheral equipment. Instead of performing a static set of functions, IT can be adjusted and re-programmed in so many different ways to fulfill the evolving and changing applications, business requirements and user needs.

- IT network in any industry is used to manage the computer systems and companies data in a more secure way.

Fig. 2.3.5 : A differentiation point between IT and OT in the IoT Reference Model framework

- As Shown in Fig. 2.3.5 IoT systems have to cross many boundaries beyond the functional layers.
- The bottom of the stack is domain of OT. For an industry like oil, and gas, this includes sensors and devices connected to pipelines, oil rigs, refinery machinery. Also industries were manufacturing is needing controlled atmosphere , Precision Pharming were sensors are used to track different parameters.
- The top of the stack is in the IT area and includes things like the servers, databases, and applications, all of which run on a part of the network controlled by IT.
- Long back OT and IT have been independent and very little interaction between both. Since IoT is evolving the whole paradigm is now changing.
- In the bottom OT layers the devices/Sensors generate real-time data at their own rate—also huge amount of data is generated on daily basis. This huge amount of data is transmitted in the IoT network, the volume of data decides applications at the top layer will be able accepting data at the rate required.
- In order to synchronize between and data generated by sensors and accepted by application, data has to be buffered or stored at certain points within the IoT stack. Data management in this way throughout the stack helps the upper four layers to handle data at their own speed.
- As a result, the real-time "data in motion" close to the edge has to be organized and stored so that it becomes "data at rest" for the applications in the IT tiers. The IT and OT organizations need to work together for overall data management.

(c) Comparing OT and IT networks

Sr. No.	Criteria	Information Technology	Open Technology
1.	Scope and Ownership	Information Technology(IT) covers the spectrum of systems that support corporate function like Finance, HR, Supply Chain, Order Management, Sales etc. More often than not, these functions and their processes tend to have commonality across industries.	Operational Technology(OT) covers the spectrum of systems that deal with the physical transformation of products and services. They are task-specific systems, are highly customized for industries and considered mission-critical. They typically fall under the domain of engineering
2.	End- Point	In the world of Corporate IT, the end -points being managed is often a human(whose job tends to be information-intensive) using a computing device(that has been relatively homogenous until the recent and growing BYOD [bring your own device] trend)	In the world of OT, the end-point being managed is often a physical asset such as pumps, motors, conveyors, valves, for kifts etc. where these things come in all shapes ,sizes, level of complexity version and vintage
3.	Focus	The software applications that make up IT portfolio are people centric in the sense that they help people "make money" by managing and coordinating the higher-level processes and transactions of the business	In contrast, most of the software application in OT's portfolio are "thing-centric" in the sense that they help "make product" by controlling the physical equipment with a great deal of precision (and safety), where the humans role is supervisory (as automation increases)
4.	Architecture	Besides being pervasive in our personal lives, IT is a relatively standardized world and that is far more homogenous than OT) IT also tends to adapt far more quickly to multiple computing trends, from PCs to internet to mobility, all of which have broadly shaped todays corporate IT strategy.	In contrast, OT is filled with silos of proprietary architectures because of its tasks-specific nature. For example, a refinery is designed so it can run continuously for 5+ years before it is shut down for maintenance. In Other words, reliability can often trump innovation, open architecture, interoperability etc.

(d) Difference between OT Network and IT Network

Sr. No.	OT Network	IT Network
1.	OT network is industrial-oriented, which mainly interacts with machines.	IT network is business-oriented, which mainly deals with information rather machines.
2.	Different types of data in OT networks include : monitoring, control and supervisory data.	Different types of data in IT networks include: Transactional, voice, video and bulky data.
3.	OT network is connected with the outside world whose access is not limited.	IT network is limited to people which have certain privileges.
4.	OT network works on real-time processing of data.	IT network works on transactional processing of data.
5.	OT network may have risk regarding the information.	IT network may have automation risk.
6.	OT network failure can result in end-of life.	IT network failure can result in loss of data.
7.	OT has less changing environment as the requirements are not frequently changing.	IT has frequently changing environment.
8.	OT network requires network upgrades only during operational maintenance windows.	IT network often requires network upgrades.
9.	If there is any disturbance in OT network, it will directly impact the overall business.	IT network failure can be business impacting, and it depends on industry.
10.	OT network controls physical access to any device.	IT network ensures security by authenticating the devices and users to the network

2.3.4 Additional IoT Reference Models

- There are several other reference models exist apart the two model discussed.
- These models are endorsed by various organizations and standards bodies and are often specific to certain industries or IoT applications.
- Table 2-3 gives details of these additional IoT reference models.

IoT reference Model	Description
Purdue Model for Control Hierarchy	• The Purdue Model for Control Hierarchy (see www.cisco.com/c/en/us/td/docs/solutions/Verticals/EttF/EttFDIG/ch2_EttF.pdf) is a common and well-understood model that segments devices and equipment into hierarchical levels and functions. • It is used as the basis for ISA-95 for control hierarchy, and in turn for the IEC-62443 (formerly ISA-99) cyber security standard. • It has been used as a base for many IoT-related models and standards across industry.
Industrial Internet Reference Architecture (IIRA) by Industrial Internet Consortium (IIC)	• The IIRA is a standards-based open architecture for Industrial Internet Systems (IISs). • To maximize its value, the IIRA has broad industry applicability to drive interoperability, to map applicable technologies, and to guide technology and standard development. • The description and representation of the architecture are generic and at a high level of abstraction to support the requisite broad industry applicability. • The IIRA distills and abstracts common characteristics, features and patterns from use cases well understood at this time, predominantly those that have been defined in the IIC.
Internet of Things–Architecture (IoT-A)	• IoT-A created an IoT architectural reference model and defined an initial set of key building blocks that are foundational in fostering the emerging Internet of Things. • Using an experimental paradigm, IoT-A combined top-down reasoning about architectural principles and design guidelines with simulation and prototyping in exploring the technical consequences of architectural design choices.

2.4 A SIMPLIFIED IOT MODEL

- An IoT framework discussed here highlights the fundamental building blocks that are common to most IoT systems and which is intended to help you in designing an IoT network.
- This framework is presented as two parallel stacks shown in Fig. 2.4.1.
  - The IoT Data Management and Compute Stack and
  - the Core IoT Functional Stack.

Fig. 2.4.1 : Simplified IoT Architecture

- Almost all IoT model includes core layers including
  - "things"
  - a communications network, and
  - applications.
- The framework separates the core IoT and data management into parallel and aligned stacks, allowing you to carefully examine the functions of both the network and the applications at each stage of a complex IoT system. This separation gives you better visibility into the functions of each layer.
- The **presentation** of the Core IoT Functional Stack in three layers is to simplify understanding of the IoT architecture into its most foundational building blocks.
- The **network communications** layer of the IoT stack itself involves a significant amount of detail and incorporates a vast array of technologies.
- The different types of IoT sensors and there are many different ways to connect them to a network. The network communications layer needs to add all these together.
- It also offer gateway and backhaul technologies, and ultimately bring the data back to a central location for analysis and processing.
- As compared to most IT networks, the applications and analytics layer of IoT doesn't necessarily exist only in the data center or in the cloud.
- This is unique challenges and requirements of IoT, it is often necessary to deploy applications and data management throughout the architecture in a tiered approach, allowing data collection, analytics, and intelligent controls at multiple points in the IoT system.
- The data management is aligned with each of the three layers of the Core IoT Functional Stack.
- The three data management layers shown in Fig. 2.4.2. are
  - the edge layer (data management within the sensors themselves),
  - the fog layer (data management in the gateways and transit network), and
  - the cloud layer (data management in the cloud or central data center).

Fig. 2.4.2 : Detailed View of the simplified IoT Architecture

- The communications layer is broken down into four separate sub-layers, those are
  - the access network,
  - gateways and backhaul,
  - IP transport, and
  - operations and management sub-layers.
- The applications layer of IoT networks is different from the application layer of a typical enterprise network.
- Instead of simply using business applications, IoT often involves a strong big data analytics component.
- IoT is not just about the control of IoT devices but, rather, the useful insights gained from the data generated by those devices.
- Thus, the applications layer typically has both analytics and industry-specific IoT control system components.

2.4.1 The Core IoT Functional Stack

IoT networks are built around the concept of "things," or smart objects performing functions and delivering new connected services. These objects are "smart" because they use a combination of contextual information and configured goals to perform actions.

Looking at the architecture there are several components working together to make an IoT network operational

A. **"Things" layer :** At this layer, the physical devices need to fit the constraints of the environment in which they are deployed while still being able to provide the information needed.

B. **Communications network layer:** When smart objects are not complete or independent, they need to communicate with an external system. Communication technology used is wireless is most of cases. This layer has four sublayers:

1. Access network sublayer
2. Gateways and backhaul network sublayer
3. Network transport sublayer
4. IoT network management sublayer

1. **Access network sublayer:** The very important part of IoT network is the access network. Uses wireless technologies such as 802.11ah, 802.15.4g, and LoRa. The sensors connected to the access network can be wired.
2. **Gateways and backhaul network sublayer:** For the multiple smart object in a specific area a common gate way and communication network is defined. The gateways function is to forward the collected information through a longer-range medium (called the backhaul) to a central station where the information is processed. This information exchange is a Layer 7 (application) function, due to this object is also called as a gateway. On IP networks, this gateway also forwards packets from one IP network to another, and it therefore acts as a router.
3. **Network transport sublayer:** For communication to be successful, network and transport layer protocols such as IP and UDP must be implemented to support the variety of devices to connect and media to use.
4. **IoT network management sublayer:** Additional protocols must be in place to allow the headend applications to exchange data with the sensors. Examples include CoAP(Constrained Application Protocol) and MQTT(Message Queuing Telemetry Transport).

C. **Application and analytics layer :** The upper most layer is an application process the collected data, It also control the smart objects when necessary, but to make intelligent decision based on the information collected and, in turn, instruct the "things" or other systems to adapt to the analyzed conditions and change their behaviors or parameters.

2.5 BUILD YOUR OWN IOT COMMUNICATION NETWORK

In these section all elements are discussed with which one can architect its own IoT communication network.

2.5.1 Layer 1: Things: Sensors and Actuators Layer

To architect these layer one should have knowledge and classification of sensor and actuators ie things like

- Is the thing Battery-powered or power-connected
- Is the thing Mobile or static

- Is the thing working at Low or high reporting frequency
- Data generated is Simple or rich data
- What is the Report range
- How objects per cell i.e. Object density per cell

a) Battery-powered or power-connected

- This classification is based on whether the object carries its own energy supply or receives continuous power from an external power source.
- Battery-powered things can be moved more easily than line-powered objects.
- However, batteries limit the lifetime and amount of energy that the object is allowed to consume, thus driving transmission range and frequency.

b) Mobile or static

- This classification is based on whether the "thing" should move or always stay at the same location. A sensor may be mobile because it is moved from one object to another (for example, a viscosity sensor moved from batch to batch in a chemical plant) or because it is attached to a moving object (for example, a location sensor on moving goods in a warehouse or factory floor).
- The frequency of the movement may also vary, from occasional to permanent. The range of mobility (from a few inches to miles away) often drives the possible power source.

c) Low or high reporting frequency

- This classification is based on how often the object should report monitored parameters.
- A rust sensor may report values once a month. A motion sensor may report acceleration several hundred times per second.
- Higher frequencies drive higher energy consumption, which may create constraints on the possible power source (and therefore the object mobility) and the transmission range.

Fig. 2.5.1 : Sensor Application based on Mobility and Throughput

d) Simple or rich data

- This classification is based on the quantity of data exchanged at each report cycle. A humidity sensor in a field may report a simple daily index value (on a binary scale from 0 to 255), while an engine sensor may report hundreds of parameters, from temperature to pressure, gas velocity, compression speed, carbon index, and many others.
- Richer data typically drives higher power consumption. This classification is often combined with the previous to determine the object data throughput (low throughput to high throughput). You may want to keep in mind that throughput is a combined metric.
- A medium-throughput object may send simple data at rather high frequency (in which case the flow structure looks continuous), or may send rich data at rather low frequency (in which case the flow structure looks bursty).

e) Report range

- This classification is based on the distance at which the gateway is located. For example, for your fitness band to communicate with your phone, it needs to be located a few meters away at most.
- The assumption is that your phone needs to be at visual distance for you to consult the reported data on the phone screen.
- If the phone is far away, you typically do not use it, and reporting data from the band to the phone is not necessary.
- By contrast, a moisture sensor in the asphalt of a road may need to communicate with its reader several hundred meters or even kilometers away

f) Object density per cell

- This classification is based on the number of smart objects (with a similar need to communicate) over a given area, connected to the same gateway.
- An oil pipeline may utilize a single sensor at key locations every few miles. By contrast, telescopes like the SETI (Search for Extraterrestrial Intelligence.) Colossus telescope at the Whipple Observatory deploy hundreds, and sometimes thousands, of mirrors over a small area, each with multiple gyroscopes, gravity, and vibration sensors.
- From a network architectural standpoint, your initial task is to determine which technology should be used to allow smart objects to communicate.

2.5.2 Layer 2 : Communications Network Layer

- After knowing the smart object form factor(Size)m its transmission capabilities (transmission range, data volume and frequency, sensor density and mobility), now smart object is ready to connect and communicate.
- Computer and network assets used in IoT are very different from those in IT environments because of physical form factors between devices used by IT and OT.

- The operational differences must be understood in order to apply the correct handling to secure the target assets. Temperature variances are an easily understood metric.
- The cause for the variance is easily attributed to external weather forces and internal operating conditions.
- Remote external locations, such as those associated with mineral extraction or pipeline equipment can span from the heat to the cold .
- Humidity fluctuations can impact the long-term success of a system as well.
- Shock and vibration needs vary based on the deployment scenario.
- Solid particulates can also impact the gear. Most IT environments must contend with dust build-up that can become highly concentrated due to the effect of cooling fans.
- Hazardous location design may also cause corrosive impact to the equipment. Caustic materials can impact connections over which power or communications travel.
- Furthermore, they can result in reduced thermal efficiency by potentially coating the heat transfer surfaces.
- In some scenarios, the concern is not how the environment can impact the equipment but how the equipment can impact the environment.
- Power supplies in OT systems are also frequently different from those commonly seen on standard IT equipment. A wider range of power variations are common attributes of industrial compute components

2.5.2(a) Access Network Sublayer

WPAN: Wireless Personal Area Network
WHAN: Wireless Home Area Network
WFAN: Wireless Field (or Factory) Area Network
WLAN: Wireless Local Area Network

WNAN: Wireless Neighborhood Area Network
WWAN: Wireless Wide Area Network
LPWA: Low Power Wide Area

Fig. 2.5.2 : Access Technologies and Distance

- One key parameter determining the choice of access technology is the range between the smart object and the information collector
- Range estimates are grouped by category names that illustrate the environment or the vertical where data collection over that range is expected.
- Common groups are as follows:
  - **PAN (personal area network):** Scale of a few meters. This is the personal space around a person. A common wireless technology for this scale is Bluetooth.
  - **HAN (home area network):** Scale of a few tens of meters. At this scale, common wireless technologies for IoT include ZigBee and Bluetooth Low Energy (BLE).
  - **NAN (neighborhood area network):** Scale of a few hundreds of meters. The term NAN is often used to refer to a group of house units from which data is collected.
  - **FAN (field area network):** Scale of several tens of meters to several hundred meters. FAN typically refers to an outdoor area larger than a single group of house units. The FAN is often seen as "open space" (and therefore not secured and not controlled). A FAN is sometimes viewed as a group of NANs, but some verticals see the FAN as a group of HANs or a group of smaller outdoor cells.
  - **LAN (local area network):** Scale of up to 100 m. This term is very common in networking, and it is therefore also commonly used in the IoT space when standard networking technologies (such as Ethernet or IEEE 802.11) are used.

Fig. 2.5.3 : Combines cost, range, power consumption, and typical available bandwidth for common IoT access technologies

- The amount of data to carry over a given time period along with correlated power consumption
- (driving possible limitations in mobility and range) determines the wireless cell size and structure.
- Similar ranges also do not mean similar topologies.
- Some technologies offer flexible connectivity structure to extend communication possibilities:
  - Point-to-point topologies
  - Point-to-multipoint topologies

(i) Point-to-point topologies

- These topologies allow one point to communicate with another point. This topology in its strictest sense is uncommon for IoT access, as it would imply that a single object can communicate only with a single gateway.
- However, several technologies are referred to as "point-to-point" when each object establishes an individual session with the gateway.
- The "point-to-point" concept, in that case, often refers to the communication structure more than the physical topology.

(ii) Point-to-multipoint topologies

- These topologies allow one point to communicate with more than one other point. Most IoT technologies where one or more than one gateways communicate with multiple smart objects are in this category.
- However, depending on the features available on each communicating mode, several subtypes need to be considered.
- A particularity of IoT networks is that some nodes (for example, sensors) support both data collection and forwarding functions, while some other nodes (for example, some gateways) collect the smart object data, sometimes instruct the sensor to perform specific operations, and also interface with other networks or possibly other gateways.
- For this reason, some technologies categorize the nodes based on the functions (described by a protocol) they implement.

(iii) Star and Clustered Star Topologies

- To form a network, a device needs to connect with another device. When both devices fully implement the protocol stack functions, they can form a peer-to-peer network. However, in many cases, one of the devices collects data from the others.
- For example, in a house, temperature sensors may be deployed in each room or each zone of the house, and they may communicate with a central point where temperature is displayed and controlled.

Fig. 2.5.4 : Star and Clustered Star Topologies

- The sensor can implement a subset of protocol functions to perform just a specialized part (communication with the coordinator). Such a device is called a reduced-function device (RFD).
- As shown in Fig. 2.5.4., An RFD cannot be a coordinator. An RFD also cannot implement direct communications to another RFD.
- The coordinator that implements the full network functions is called, by contrast, a full-function device (FFD). An FFD can communicate directly with another FFD or with more than one FFD, forming multiple peer-to-peer connections.
- Topologies where each FFD has a unique path to another FFD are called cluster tree topologies. FFDs in the cluster tree may have RFDs, resulting in a cluster star topology.

(iv) Mesh Topology

- Point-to-multipoint shown in Fig. 2.5.5. Technologies allow a node to have more than one path to another node, forming a mesh topology. This redundancy means that each node can communicate with more than just one other node.

Fig. 2.5.5 : Mesh Toplogy

- This communication can be used to directly exchange information between nodes (the receiver directly consumes the information received) or to extend the range of the communication link. In this case, an intermediate node acts as a relay between two other nodes.
- These two other nodes would not be able to communicate successfully directly while respecting the constraints of power and modulation dictated by the PHY layer protocol.
- Another property of mesh networks is redundancy. The disappearance of one node does not necessarily interrupt network communications.
- Nodes A and D are too far apart to communicate directly. In this case, communication can be relayed through nodes B or C. Node B may be used as the primary relay.
- However, the loss of node B does not prevent the communication between nodes A and D. Here, communication is rerouted through another node, node C.

2.5.2(b) Gateways and Backhaul Sublayer

- Data collected from a smart object may need to be forwarded to a central station where data is processed. As this station is often in a different location from the smart object, data directly received from the sensor through an access technology needs to be forwarded to another medium (the backhaul) and transported to the central station. The gateway is in charge of this inter-medium communication.

- In the DSRC case, the entire "sensor field" is moving along with the gateway, but the general principles of IoT networking remain the same.
- The range at which DSRC can communicate is limited. Similarly, for all other IoT architectures, the choice of a backhaul technology depends on the communication distance and also on the amount of data that needs to be forwarded.
- When the smart object's operation is controlled from a local site, and when the environment is stable (for example, factory or oil and gas field), Ethernet can be used as a backhaul. Mesh is a common topology to allow communication flexibility in this type of dynamic environment.

Table 2.5.1 : Compares the main solutions from an architectural angle

Technology	Type and Range	Architectural Characteristics
Ethernet	Wired, 100 m max	Requires a cable. per sensor/sensor group; adapted to static sensor position in a stable environment; range is limited; link is very reliable
Wi-Fi (2.4 GHz, 5 GHz)	Wireless, 100 m (multipoint) to a few kilometers (P2P)	Can connect multiple clients (typically fewer than 200) to a single AP: range is Limited: adapted to cases where client power is not an issue (continuous power or client battery recharged easily); large bandwidth available, but interference from other systems likely; AP needs a cable.
802.11ah (Halo W, Wi-Fi in sub-1 GHz	Wireless, 1.5 km (multipoint), 10 km (P2P)	Can connect a Large number of clients (up to 6000 per AP); longer range than traditional Wi-Fi; power efficient; limited bandwidth; low adoption; and cost may be an issue
WiMAX (802.16)	Wireless,, several kilometers (last mile). up to 50 km (backhaul)	Can connect a Large number of clients:. Large bandwidth available in licensed spectrum (fee-based); reduced bandwidth in license-free spectrum (interferences from other systems likely); adoption varies on location
Cellular (for example, LTE)	Wireless, several kilometers	Can connect a Large number of clients; large bandwidth available' licensed spectrum (interference-free; license-based)

2.5.2(c) Network Transport Sub-layer

- Distribution automation (DA) allows your meter to communicate with neighboring meters or other devices in the electrical distribution grid. With such communication, consumption load balancing may be optimized.
- For example, your air conditioning pulses fresh air at regular intervals. With DA, your neighbor's AC starts pulsing when your system pauses; in this way, the air in both houses is kept fresh, but the energy consumed from the network is stable instead of spiking up and down with uncoordinated start and stop points.

- Similarly, smart meter may communicate with your house appliances to evaluate their type and energy demand. With this scheme, washing machine can be turned on in times of lower consumption from other systems, such as at night, while powering home theater system will never be deprived, always turning on when you need it.
- Once the system learns consumption pattern, charging of electric car can start and stop at intervals to achieve the same overnight charge without creating spikes in energy demand. Data may flow locally, or it may have to be orchestrated by a central application to coordinate the power budget between houses.
- This communication structure thus may involve peer-to-peer, point-to-point, point-to-multipoint, unicast and multicast communications. In a multitenant environment (for example, electricity and gas consumption management), different systems may use the same communication pathways.
- This communication occurs over multiple media (for example, power lines inside your house or a shortrange wireless system like indoor Wi-Fi and/or ZigBee), a longer-range wireless system to the gateway, and yet another wireless or wired medium for backhaul transmission.
- To allow for such communication structure, a network protocol with specific characteristics needs to be implemented. The protocol needs to be open and standard-based to accommodate multiple industries and multiple media.
- Scalability (to accommodate thousands or millions of sensors in a single network) and security are also common requirements. IP is a protocol that matches all these requirements.
- The flexibility of IP allows this protocol to be embedded in objects of very different natures, exchanging information over very different media, including low-power, lossy, and low-bandwidth networks.
- For example, RFC 2464 describes how an IPv6 packet gets encapsulated over an Ethernet frame and is also used for IEEE 802.11 Wi-Fi.
- Similarly, the IETF 6LoWPAN working group specifies how IPv6 packets are carried efficiently over lossy networks, forming an "adaption layer" for IPv6, primarily for IoT networks.
- Finally, the transport layer protocols built above IP (UDP and TCP) can easily be leveraged to decide whether the network should control the data packet delivery (with TCP) or whether the control task should be left to the application (UDP). UDP is a much lighter and faster protocol than TCP.
- However, it does not guarantee packet delivery. Both TCP and UDP can be secured with TLS/SSL (TCP) or DTLS (UDP).

2.5.2(d) IoT Network Management Sub-layer

- IP, TCP, and UDP bring connectivity to IoT networks. Upper-layer protocols need to take care of data transmission between the smart objects and other systems. Multiple protocols have been leveraged or created to solve IoT data communication problems.

- Some networks rely on a push model (that is, a sensor reports at a regular interval or based on a local trigger), whereas others rely on a pull model (that is, an application queries the sensor over the network), and multiple hybrid approaches are also possible.
- Following the IP logic, some IoT implementers have suggested HTTP for the data transfer phase. After all, HTTP has a client and server component. The sensor could use the client part to establish a connection to the IoT central application (the server), and then data can be exchanged.
- To find HTTP in some IoT applications, but HTTP is something of a fat protocol and was not designed to operate in constrained environments with low memory, low power, low bandwidth, and a high rate of packet failure. Despite these limitations, other web-derived protocols have been suggested for the IoT space.
- One example is WebSocket. WebSocket is part of the HTML5 specification, and provides a simple bidirectional connection over a single connection. Some IoT solutions use WebSocket to manage the connection between the smart object and an external application. WebSocket is often combined with other protocols, such as MQTT (described shortly) to handle the IoT-specific part of the communication.
- To respond to the limits of web-based protocols, another protocol was created by the IETF Constrained Restful Environments (CoRE) working group: Constrained Application Protocol (CoAP).
- CoAP uses some methods similar to those of HTTP (such as Get, Post, Put, and Delete) but implements a shorter list, thus limiting the size of the header. CoAP also runs on UDP (whereas HTTP typically uses TCP). CoAP also adds a feature that is lacking in HTTP and very useful for IoT: observation.
- Observation allows the streaming of state changes as they occur, without requiring the receiver to query for these changes.

2.5.3 Layer 3 : Applications and Analytics Layer

- Once connected to a network, your smart objects exchange information with other systems.
- Once IoT network spans more than a few sensors, the power of the Internet of Things appears in the applications that make use of the information exchanged with the smart objects.

2.6 ANALYTICS VERSUS CONTROL APPLICATIONS

2.6.1 Analytics Application

- Collects data from multiple smart objects
- Processes the collected data
- Displays information resulting from the data that was processed
- Application processes the data to convey a view of the network that cannot be obtained from solely looking at the information displayed by a single smart object.

2.6.2 Control Application

- Controls the behavior of the smart object or the behavior of an object related to the smart object.
- Used for controlling complex aspects of an IoT network with a logic that cannot be programmed inside a single IoT object
- An example of control system architecture is SCADA. SCADA was developed as a universal method to access remote systems and send instructions. One example where SCADA is widely used is in the control and monitoring of remote terminal units (RTUs) on the electrical distribution grid.
- Advanced IoT applications include both analytics and control modules. Data is collected from the smart objects and processed in the analytics module in many cases.
- The result of this processing may be used to modify the behavior of smart objects or systems related to the smart objects.
- The control module is used to convey the instructions for behavioral changes. When evaluating an IoT data and analytics application, you need to determine the relative depth of the control part needed for your use case and match it against the type of analytics provided.

2.7 DATA VERSUS NETWORK ANALYTICS

2.7.1 Data Analytics

- Processes the data collected by smart objects and combines it to provide an intelligent view related to the IoT system.
- A simple dashboard can display an alarm when a weight sensor detects that a shelf is empty in a store. A complex case, temperature, pressure, wind, humidity, and light levels collected from thousands of sensors may be combined and then processed to determine the chances of a storm and its possible path
- Data processing can be very complex and may combine multiple changing values over complex algorithms.
- Data analytics can also monitor the IoT system itself. For example, a machine or robot in a factory can report data about its own movements. This data can be used by an analytics application to report degradation in the movement speeds, which may be indicative of a need to service the robot before a part breaks.

2.7.2 Network Analytics

- Since the system is built with smart objects connected to the network. A loss or degradation in connectivity is likely to affect the efficiency of the system.
- For example, open mines use wireless networks to automatically pilot dump trucks. A loss of connectivity may result in an accident or degradation of operations efficiency (automated dump trucks typically stop upon connectivity loss).

- Also loss of connectivity means that data stops being fed to your data analytics platform, and the system stops making intelligent analyses of the IoT system.
- Most analytics applications employ both
  - Data Analytics and
  - Network Analytics modules.

(i) Network Analytics Modules

- Network analytics is necessary for connected systems. However, the depth of analysis depends on your use cases.
- A basic connectivity view may be enough if the smart objects report occasional status, without expectation for immediate action based on this report.
- Detailed analysis and trending about network performance are needed if the central application is expected to pilot in near-real-time connected systems.

(ii) Data Analytics Modules

- Data analytics is a wider space with a larger gray area (in terms of needs) than network analytics. Basic systems analytics can provide views of the system state and state trend analysis.
- More advanced systems can refine the type of data collected and display additional information about the system. The type of collected data and processing varies widely with the use case.

2.8 DATA ANALYTICS VERSUS BUSINESS BENEFITS

- A smarter architectural choice may be to allow for an open system where the network is engineered to be flexible enough that other sensors may be added in the future, and where both upstream and downstream operations are allowed.
- This flexibility allows for additional processing of the existing sensors and also deeper and more efficient interaction with the connected objects. This enhanced data processing can result in new added value for businesses that are not envisioned at the time when the system is initially deployed.
- An example of a flexible analytics and control application is Cisco Jasper, which provides a turnkey cloud-based platform for IoT management and monetization. Consider the case of vending machines deployed throughout a city. At a basic level, these machines can be connected, and sensors can be deployed to report when a machine is in an error state.
- A repair person can be sent to address the issue when such a state is identified. This type of alert is a time saver and avoids the need for the repair team to tour all the machines in turn when only one may be malfunctioning.
- This alert system may also avoid delay between the time when a machine goes into the error state and the time when a repair team visits the machine location. With a static platform, this use case is limited to this type of alert. With a flexible platform like Cisco Jasper, new applications may be imagined and developed over time.

- For example, the machine sensors can be improved to also report when an item is sold. The central application can then be enhanced to process this information and analyze what item is most sold, in what location, at what times.
- This new view of the machines may allow for an optimization of the items to sell in machines in a given area. Systems may be implemented to adapt the goods to time, season, or location—or many other parameters that may have been analyzed. In short, architecting open systems opens the possibility for new applications.

2.8.1 Data Analytics and Business Analytics Comparison Table

Following is the list of points that show the comparisons between Data Analytics and Business Analytics :

Basis For Comparison	Business Analytics	Data Analytics
Focus	A business analyst would be responsible for making the reports, KPI (Key Performance Index) matrix, trends in the data which would help the organization	A data analyst would just play with the data to find patterns, correlations and even build models to see how the data responds to his/her models.
Process	A business analyst would do a static and comparative study of the data.	A data analyst would do an explanatory analysis and then will try to experiment with data mining processes so as to give a good visual representation of the data.
Data Sources	A business analysts would pre-plan his/her sources of data as to what all are necessary and which should be excluded which is a slow process.	A data analyst finds a correlation on some data which is not a part of his earlier dataset then he/she would add the data source on the fly as needed.
Transform	A business analyst would transform the data upfront which is carefully planned.	All the transformations are done in-database and whenever there is a demand to enrich data it is done on the fly.
Data Quality	A business analyst would always present the data as a single version of truth	A business analyst would go by the phrase "Good enough" or theoretically with the probabilities
Data Model	A business analyst would go with schema on load data model	A data analyst would go with schema on query data model.
Analysis	Retrospective, descriptive	Predictive, prescriptive
Field	A subset of computer science and management where the study of data is done by using different methods and technologies	Covers entire technological field which is a superset of Data Science

2.8.2 Key Differences Between Data Analytics and Business Analytics

Below are the lists of points, describe the key Differences Between Data Analytics and Business Analytics :

- The key tasks of a business analyst will be checking the requirement assessing it with a point of operations and functions whereas a data analyst will only analyze the data in terms of collecting, manipulating and analyzing the data.
- The business analyst goes through all the requirements by scoping and de-scoping the requirements and then assign the tasks to the developers to develop the code whereas a data analyst would be preparing dashboards charts or various visualizations which would help the higher management to take calls on what should be done next.
- The business analyst would research and try to gain valuable insights from the data, finding the optimal model for the business also lies with the business analyst whereas a data analyst would concentrate on developing new algorithms or to optimize the already developed algorithms.
- An example and try to differentiate between the two:
  1. We have a study where a telecom company needs to segregate their customers in order find the unwanted customers or let's just say the churn rate. A business analyst would ask the developers to build models by giving them all the data they require and then try to evaluate which model describes the best.
  2. Whereas a data analyst would be taking care of cleaning the data, transforming the data so that it could fit good enough for the model, tweaking the model for better results, building visual outputs so as to make the model easily understandable.

2.9 SMART SERVICES

- Due to the availability of large amounts of data and the possibility to use it for specific purposes, such as the analysis or improvement of certain processes, digital (smart) services can be offered via the Internet. Such digital services are known from online shops, for example, when personalized recommendations or individualized advertising banners are displayed based on the available data.
- The German National Academy of Science and Engineering acatech defines Smart Services as 'packages of products, services and features individually configured via the Internet, tailored to the preferences of private and commercial users in a needs-based and situation-specific 'as a service' manner.
- Digital platforms play a central role: This is where products and services are mapped virtually, combined, enhanced with additional digital services and offered as Smart Services'.
- Transferred to industrial applications, these services extend or improve existing functions for product services. They can thus be used as a basis for decisions on optimising, controlling or adapting individual processes (at customers).

2.9.1 Smart Services Use IoT and Aim for Efficiency

- For example, sensors can be installed on equipment to ensure ongoing conformance with regulations or safety requirements. This angle of efficiency can take multiple forms, from presence sensors in hazardous areas to weight threshold violation detectors on trucks.
- Smart services can also be used to measure the efficiency of machines by detecting machine output, speed, or other forms of usage evaluation. Entire operations can be optimized with IoT.

Fig. 2.9.1 : Smart Services in Safety and aim for efficiency

2.9.2 Smart Services in Hospitality

- For example, presence and motion sensors can evaluate the number of guests in a lobby and redirect personnel accordingly. The same type of action can be taken in a store where a customer is detected as staying longer than the typical amount of time in front of a shelf.
- Personnel can be deployed to provide assistance. Movement of people and objects on factory floors can be analyzed to optimize the production flow.

Fig. 2.9.2 : Smart Services in hospitality

2.9.3 Smart Services can be Integrated into an IoT System like Smart Home

- For example, sensors can be integrated in a light bulb. A sensor can turn a light on or off based on the presence of a human in the room.
- An even smarter system can communicate with other systems in the house, learn the human movement pattern, and anticipate the presence of a human, turning on the light just before the person enters the room.

Fig. 2.9.3 : Smart services can be integrated into an IoT system like Smart Home

- An even smarter system can use smarter sensors that analyze multiple parameters to detect human mood and modify accordingly the light color to adapt to the learned preferences, or to convey either a more relaxing or a more dynamic environment.
- By connecting to other systems in the house, efficiencies can be coordinated. For example, the house entry alarm system or the heating system can coordinate with the presence detector in a light bulb to adapt to detected changes. The alarm system can disable volumetric movement alarms in zones where a known person is detected. The heating system can adapt the temperature to human presence or detected personal preferences.

2.9.4 Efficiency can be Extended to Larger Systems Like Smart Grid

- For example, smart grid applications can coordinate the energy consumption between houses to regulate the energy demand from the grid.

Fig. 2.9.4 : Efficiency can be extended to larger systems Like Smart Grid

- It is already mentioned that your washing machine may be turned on at night when the energy demand for heating and cooling is lower.
- Just as your air conditioning pulses can be coordinated with your neighbor's, your washing machine cycles can be coordinated with the appliances in your house and in the neighborhood to smooth the energy demand spikes on the grid.

2.9.5 Efficiency also applies to M2M Communications

- In mining environments, vehicles can communicate to regulate the flows between drills, draglines, bulldozers, and dump trucks.

Fig. 2.9.5 : Efficiency also applies to M2M communications

- For example, making sure that a dump truck is always available when a bulldozer needs it. In smart cities, vehicles communicate.
- A traffic jam is detected and anticipated automatically by public transportation, and the system can temporarily reroute buses or regulate the number of buses servicing a specific line based on traffic and customer quantity, instantaneous or learned over trending.

2.10 IOT DATA MANAGEMENT AND COMPUTE STACK

Fig. 2.10.1 : Growth of devices connected to Internet

- Fig. 2.10.1 shows how the "things" connected to the Internet are continuing to grow exponentially, with a prediction by Cisco that by 2020 there will be more than 50 billion devices connected to some form of an IP network.
- However, beyond the network architecture itself, consider the data that is generated by these devices. If the number of devices is beyond conventional numbers, surely the data generated by these devices will be huge and must be thought about it seriously.
- In fact, the data generated by IoT sensors is one of the single biggest challenges in building an IoT system.
- In the case of modern IT networks, the data sourced by a computer or server is typically generated by the client/server communications model, and it serves the needs of the application.
- In sensor networks, the vast majority of data generated is unstructured and of very little use on its own.
- For example, the majority of data generated by a smart meter is nothing more than polling data; the communications system simply determines whether a network connection to the meter is still active.
- This data on its own is of very little value. The real value of a smart meter is the metering data read by the meter management system (MMS). However, if you look at the raw polling data from a different perspective, the information can be very useful.
- For example, a utility may have millions of meters covering its entire service area. If whole sections of the smart grid start to show an interruption of connectivity to the meters, this data can be analyzed and combined with other sources of data, such as weather reports and electrical demand in the grid, to provide a complete picture of what is happening.

- This information can help determine whether the loss of connection to the meters is truly a loss of power or whether some other problem has developed in the grid. Moreover, analytics of this data can help the utility quickly determine the extent of the service outage and repair the disruption in a timely fashion.
- In most cases, the processing location is outside the smart object. A natural location for this processing activity is the cloud. Smart objects need to connect to the cloud, and data processing is centralized.
- However, this model also has limitations. As data volume, the variety of objects connecting to the network, and the need for more efficiency increase.
- These new requirements include the following:
  - **Minimizing latency :** Analyzing data close to the device that collected the data can make a difference between averting disaster and a cascading system failure.
  - **Conserving network bandwidth :** It is not practical to transport vast amounts of data from thousands or hundreds of thousands of edge devices to the cloud. Nor is it necessary because many critical analyses do not require cloud-scale processing and storage.
  - **Increasing local efficiency:** Collecting and securing data across a wide geographic area with different environmental conditions may not be useful.
- The volume of data also introduces questions about bandwidth management.
- As the massive amount of IoT data begins to funnel into the data center, does the network have the capacity to sustain this volume of traffic?
- Does the application server have the ability to ingest, store, and analyze the vast quantity of data that is coming in?
- This is sometimes referred to as the "impedance mismatch" of the data generated by the IoT system and the management application's ability to deal with that data.

Fig. 2.10.2 : Traditional IT cloud Computing Model

- As shown in Fig. 2.10.2 data management in traditional IT systems is very simple.
- The endpoints (laptops, printers, IP phones, and so on) communicate over an IP core network to servers in the data center or cloud.
- Data is generally stored in the data center, and the physical links from access to core are typically high bandwidth, meaning access to IT data is quick.

2.10.1 Several Data-Related Problems Need to be Addressed

- **Bandwidth** in last-mile IoT networks is very limited. When dealing with thousands/millions of devices, available bandwidth may be on order of tens of Kbps per device or even less.
- **Latency** can be very high. Instead of dealing with latency in the milliseconds range, large IoT networks often introduce latency of hundreds to thousands of milliseconds.
- **Network** backhaul from the gateway can be unreliable and often depends on 3G/LTE or even satellite links. Backhaul links can also be expensive if a per-byte data usage model is necessary.
- The volume of data transmitted over the backhaul can be high, and much of the data may not really be that interesting (such as simple polling messages).
- **Big data** is getting bigger. The concept of storing and analyzing all sensor data in the cloud is impractical. The sheer volume of data generated makes real-time analysis and response to the data almost impossible.

2.10.2 Fog Computing

- The solution to the challenges mentioned in the previous section is to distribute data management throughout the IoT system, as close to the edge of the IP network as possible. The best-known embodiment of edge services in IoT is fog computing.
- Any device with computing, storage, and network connectivity can be a **fog node.**
- **Examples** include industrial controllers, switches, routers, embedded servers, and IoT gateways.
- Analyzing IoT data close to where it is collected minimizes latency, offloads gigabytes of network traffic from the core network, and keeps sensitive data inside the local network.

Fig. 2.10.3 : The Iot Data Management and Compute stack with Fog Computing

- Fog services are typically accomplished very close to the edge device, sitting as close to the IoT endpoints as possible.
- The fog node has contextual awareness of the sensors it is managing because of its geographic proximity to those sensors.

- The fog node is able to analyze information from all the sensors and can provide contextual analysis of the messages it is receiving and may decide to send back only the relevant information over the backhaul network to the cloud.

- IoT fog computing enables data to be preprocessed and correlated with other inputs to produce relevant information. This data can then be used as real-time, actionable knowledge by IoT-enabled applications. Longer term, this data can be used to gain a deeper understanding of network behavior and systems for the purpose of developing proactive policies, processes, and responses.
- Fog applications are as diverse as the Internet of Things itself. What they have in common is data reduction monitoring or analyzing real-time data from network-connected things and then initiating an action, such as locking a door, changing equipment settings, applying the brakes on a train, zooming a video camera, opening a valve in response to a pressure reading, creating a bar chart, or sending an alert to a technician to make a preventive repair.

2.10.2(a) Characteristic of Fog Computing

- Contextual location awareness and low latency: The fog node sits as close to the IoT endpoint as possible to deliver distributed computing.
- Geographic distribution: In sharp contrast to the more centralized cloud, the services and applications targeted by the fog nodes demand widely distributed deployments.
- Deployment near IoT endpoints: Fog nodes are typically deployed in the presence of a large number of IoT endpoints. For example, typical metering deployments often see 3000 to 4000 nodes per gateway router, which also functions as the fog computing node.
- Wireless communication between the fog and the IoT endpoint: Although it is possible to connect wired nodes, the advantages of fog are greatest when dealing with a large number of endpoints, and wireless access is the easiest way to achieve such scale.
- Use for real-time interactions: Important fog applications involve real-time interactions rather than batch processing. Preprocessing of data in the fog nodes allows upper-layer applications to perform batch processing on a subset of the data.

2.10.3 Edge Computing

- Fog computing solutions are being adopted by many industries, and efforts to develop distributed applications and analytics tools are being introduced at an accelerating pace. The natural place for a fog node is in the network device that sits closest to the IoT endpoints, and these nodes are typically spread throughout an IoT network.
- IoT devices and sensors often have constrained resources, however, as compute capabilities increase. Some new classes of IoT endpoints have enough compute capabilities to perform at least low-level analytics and filtering to make basic decisions. For example, consider a water sensor on a fire hydrant.

- While a fog node sitting on an electrical pole in the distribution network may have an excellent view of all the fire hydrants in a local neighborhood, a node on each hydrant would have clear view of a water pressure drop on its own line and would be able to quickly generate an alert of a localized problem.
- The fog node would have a wider view and would be able to ascertain whether the problem was more than just localized but was affecting the entire area. Another example is in the use of smart meters.
- Edge compute–capable meters are able to communicate with each other to share information on small subsets of the electrical distribution grid to monitor localized power quality and consumption, and they can inform a fog node of events that may pertain to only tiny sections of the grid. Models such as these help ensure the highest quality of power delivery to customers.

2.10.3(a) The Hierarchy of Edge, Fog and Cloud

Fig. 2.10.4 : Distributed Computer and Data Management across an IoT System

- This model suggests a hierarchical organization of network, compute, and data storage resources. At each stage, data is collected, analyzed, and responded to when necessary, according to the capabilities of the resources at each layer. As data needs to be sent to the cloud, the latency becomes higher.

- Edge and fog thus require an abstraction layer that allows applications to communicate with one another.
- The abstraction layer :
  - exposes a common set of APIs for monitoring, provisioning, and controlling the physical resources in a standardized way.
  - requires a mechanism to support virtualization, with the ability to run multiple operating systems or service containers on physical devices to support multitenancy and application consistency across the IoT system.

From an architectural standpoint, fog nodes closest to the network edge receive the data from IoT devices. The fog IoT application then directs different types of data to the optimal place for analysis:

- The most time-sensitive data is analyzed on the edge or fog node closest to the things generating the data.
- Data that can wait seconds or minutes for action is passed along to an aggregation node for analysis and action.
- Data that is less time sensitive is sent to the cloud for historical analysis, big data analytics, and long-term storage. For example, each of thousands or hundreds of thousands of fog nodes might send periodic summaries of data to the cloud for historical analysis and storage.

Chapter Ends...

❑❑❑

MODULE 3

Principles of Connected Devices and Protocols in IoT

Syllabus

RFID and NFC (Near-Field Communication), Bluetooth Low Energy (BLE) roles, LiFi , WPAN std : 802.15 standards: Bluetooth, IEEE 802.15.4, Zigbee, Z-wave, Narrow Band IoT, Internet Protocol and Transmission Control Protocol, 6LoWPAN, WLAN and WAN , IEEE 802.11, Long-range Communication Systems and Protocols: Cellular Connectivity-LTE, LTE-A, LoRa and LoRaWAN

3.1 Radio Frequency Identification (RFID) 3-3

3.1.1 Architecture of RFID 3-3

3.1.2 Different Types of RFID System 3-4

3.1.3 RFID Applications and Use cases 3-4

3.2 Near Field Communication (NFC) 3-4

3.2.1 History of Near Field Communication 3-4

3.2.2 Modes of Operation 3-5

3.2.3 Applications of Near Field Communication 3-5

3.3 Bluetooth Low Energy (BLE) 3-6

3.3.1 Difference Between Bluetooth Low Energy and Bluetooth Classic is shown below 3-6

3.3.2 Bluetooth Low Energy Architecture 3-7

3.4 Li-Fi (Light Fidelity) 3-8

3.4.1 Applications of Li-Fi 3-9

3.4.2 Advantages of Li-Fi Over Wi-Fi 3-9

3.5 WPAN (Wireless Personal Area Network) 3-10

3.6 802.15 Standard Bluetooth: 3-10

3.6.1 Bluetooth Applications 3-11
3.6.2 Bluetooth Protocol Stack 3-12
3.6.3 Functions of the Core Protocols 3-12
3.7 IEEE 802.15.4 3-12
3.7.1 IEEE 802.15. Protocol Stacks include: 3-12
3.8 Zigbee 3-13
3.9 Z-Wave 3-15
3.9.1 Z-Wave Network 3-15
3.10 Narrow Band IoT 3-16
3.11 Internet Protocol (IP) 3-16
3.11.1 Transmission Control Protocol 3-19
3.12 6LoWPAN 3-21
3.12.1 Advantages of 6LoWPAN 3-21
3.12.2 6LoWPAN Application Areas 3-22
3.13 WLAN and WAN 3-22
3.13.1 Types of WLANS 3-23
3.13.2 Frame Format of IEEE 802.11 3-23
3.13.3 Advantages of WLANs 3-24
3.13.4 Disadvantages of WLANs 3-24
3.13.5 Wide Area Network 3-24
3.14 Long Range Communication System and Protocol:Cellular Connectivity LTE, LTE-A, LoRa and LoRaWAN 3-25
3.14.1 Long Term Evolution 3-25
3.14.2 LTE-A 3-26
3.14.3 LoRa 3-27
3.14.4 LoRaWAN 3-28
3.15 Multiple Choice Questions 3-29
- **Chapter End**... 3-30

3.1 RADIO FREQUENCY IDENTIFICATION (RFID)

- Tags and readers are the two separate entities of smart transmission known as Radio Frequency Identification (RFID). One or multiple antenna's are there for receiver that emits radio waves and receive signal back from transmitter i.e. Tags. Tags can be active or passive which uses radio waves to communicate their presence and supporting information to nearby readers.
- Passive RFID Tags are operating by support of readers and do not have battery in their architecture. Passive tags are used in application like tracking inventory, access control and in wholesale and retail sector.
- Active tags are having inbuilt battery support to perform actions. RFID tags can contain a variety of data, ranging from a single serial number to many pages of information. Readers are mobile in nature so that they can carry from one location to another location very easily and they can be easily mounted on anywhere. The active tages are used to track real time location and high speed environments like toll. Reader systems may also be included into the design of a cabinet, chamber, or building.

3.1.1 Architecture of RFID

- The architecture mainly consists of Transceiver which is connected with antenna and set of tags where the data is stored. The antenna creates communication between transmitter and receiver. The server is responsible for storing and matching the data, if data is matched then valid entry is found and data is validated or if data is not matched then data it shows error.
- Static readers and mobile readers are the two different categories of RFID readers.The RFID reader is a network-connected gadget that can be carried about or fixed to a surface. It sends signals that turn on the tag using radio waves. After being turned on, the tag returns a wave to the antenna, where it is converted into information.

(1C1)**Fig. 3.1.1 : Architecture of RFID System**

3.1.2 Different Types of RFID System

1. **Low Frequency RFID System :** This system has very low transmission range generally from few inches to 5 feet's. Its frequency range is from 30 KHz to 500 KHz.
2. **High-frequency RFID system :** Its operating frequency is from 3MHz to 30MHz. The standard range is anywhere from a few inches to several feet.
3. **UHF RFID systems :** Its frequency ranges from 300 MHz to 960 MHz and can generally be read from 25+ feet away.
4. **Microwave RFID systems :** These run at 2.45 Ghz and can be read from 30+ feet away.

3.1.3 RFID Applications and Use Cases

1. Retail Sector
2. Transport
3. Inventory Control
4. Vehicle Tracking
5. Customer service and loss control.
6. Shipping
7. Healthcare
8. Animal Tracking

3.2 NEAR FIELD COMMUNICATION (NFC)

- A set of protocols known as near-field communication (NFC) allows two electronic devices to communicate at a 4 cm distance.NFC relies on the inductive coupling of two antennae on NFC-enabled devices, such as a smartphone and a printer, which can communicate in either or both ways. NFC is working in short range with high frequency.
- Your smartphone, tablet, wearables, payment cards, and other devices become increasingly smarter thanks to near-field communication (NFC), a short-range wireless technology.
- Whether paying bills, exchanging business cards, or downloading coupons, NFC allows for quick, simple information transfer across devices with just a single touch.

3.2.1 History of Near Field Communication

- The first patent of RFID was granted to Charles Walton.
- In 2004 Forum was created by Nokia and Sony for NFC.
- Nokia 3161 was the NFC phone launched.
- In 2009 NFC forum release peer to peer standard. Samsung Nexus was the first android phone launched in 2010.
- Google launched NFC to share contacts, Video and Audio. Apple introduced apple pay using NFC in iPhone 6, 6s and advanced version.

(1C2)**Fig. 3.2.1 : Timeline of Near Field Communication**

3.2.2 Modes of Operation

- Near field communication is working in two modes of Operation: Active mode and passive mode
- In Active mode, electromagnetic field and exchange of data is generated by NFC chip in both devices. Example of active mode is Bluetooth, xender and share it.
- There is only one active devices and other uses field to share information in Passive mode.

3.2.3 Applications of Near Field Communication

(1C3)**Fig. 3.2.2 : Applications of RFID**

1. **Smart Cards :** Smart cards with NFC integration make it simpler to pay than the traditional, multi-step payment process. Customers can purchase NFC-embedded smart cards from reputable payment processors like Visa and MasterCard. Smart cards with NFC integration can be used to quickly pay for groceries, pay parking fines, accrue shopping points, and redeem coupons with just a single tap. Smart cards with NFC chips are offered by all the major banks worldwide.
2. **E-wallet :** Beginning this decade, mobile-based cashless payment systems gained popularity, and more businesses are now accepting them for the convenience of their customers. Payments can be

done using smart phone applications by just tapping or waving the card in close range. Using an implanted NFC tag, service providers can incorporate a payment option into smartphones. The most widely used mobile payment apps are Apple Pay, Google Wallet.

3. **Smart Ticketing :** Smart tickets that incorporate integrated smart chips can take the place of conventional airline, railroad, and bus ticketing systems, etc. Smart posters, movie tickets, concert tickets, ads, leaflets, and information links can all contain NFC tags. Customers will be able to tap NFC tags placed at designated locations to enter a reserved area or activate tickets. All it takes is a quick scan of the smart tag to get further details.
4. **Medicine and Healthcare :** The FC integrated system is applicable to medical and healthcare tasks. By adding NFC tags to patient documents, NFC makes it easier to check in, make payments, check a patient's status, and trace records. It also improves the accuracy and simplicity of providing medications. Devices with NFC integration can be quickly paired and set up. Access to medical equipment and gadgets is simple for medical practitioners.

3.3 BLUETOOTH LOW ENERGY (BLE)

- The Bluetooth Low Energy (LE) communication is built to operate at extremely low power levels.
- The Bluetooth LE radio offers manufacturers a significant amount of flexibility to create products that match the specific connection requirements of their market by transmitting data over 40 lines in the 2.4GHz ISM frequency spectrum.
- BLE supports a variety of communication topologies, ranging from point-to-point to broadcast and, most recently, mesh, making it possible to build dependable, extensive device networks using Bluetooth technology.
- One device can now use BLE features to determine the location, size, and direction of another device.
- Bluetooth Low Energy is designed to offer significantly lower power and cost while keeping a similar communication range as compared to Classic Bluetooth.Native support for Bluetooth Low Energy is provided by the mobile operating systems iOS, Android, Windows Phone, BlackBerry, Linux, macOS, Windows 8, Windows 10, and Windows 11.

3.3.1 Difference Between Bluetooth Low Energy and Bluetooth Classic is shown below

Parameter	Bluetooth Low Energy	Bluetooth Classic
Channels	40 channels with 2 MHz spacing	79 channels with 1 MHz spacing
Data Transports	• Asynchronous Connection-oriented • Isochronous Connection-oriented	• Asynchronous Connection-oriented • Synchronous Connection-oriented

	• Asynchronous Connectionless • Synchronous Connectionless • Isochronous Connectionless	
Communication Topologies	• Point-to-Point (including piconet) • Broadcast Mesh	Point-to-Point (including piconet)
Channel Usage	Frequency-Hopping Spread Spectrum (FHSS)	GFSK, π/4 DQPSK, 8DPSK
Power	0.01–0.50 W	1 W

3.3.2 Bluetooth Low Energy Architecture

(1C4)**Fig. 3.3.1 : Bluetooth Low Energy Architecture**

The Detail Architecture is explained as follows

Physical Layer and Link Layer

The analogue communications circuitry is actually present in the physical layer. The radio divides the 2.4 GHz ISM (Industrial, Scientific, and Medical) band into 40 channels. The component that has a direct interface with the physical layer is the link layer

Host Controller Interface (HCI)

In order for the controller and host to communicate with one another, HCI defines a set of instructions and events.

L2CAP

- In this layer, there are two functionalities. In the beginning, it performs the function of a protocol multiplexer, taking several protocols from the top levels and encapsulating them in the typical BLE packet structure.
- The Attribute Protocol (ATT) and the Security Manager Protocol are the two primary protocols for Bluetooth Low Energy that the L2CAP layer is responsible for routing (SMP)
- A straightforward client/server stateless protocol based on attributes provided by a device is known as the Attribute Protocol (ATT). Whether a device is a master or a slave in BLE, it can be a client, a server, or both. Data is sent to clients by servers in response to requests from clients for data. Each server houses data arranged as attributes, each of which is given a specific value.
- A protocol and a set of security algorithms make up the Security Manager (SM).

Generic Access Profile (GAP)

GAP establishes different sets of rules and concepts to regulate and standardize the low level operation of devices:

1. Roles and interaction between them.
2. Operational modes and transitions across those.
3. Security aspects, including security modes and procedures.

The Generic Attribute Profile

The specifics of how to exchange all profile and user data over a BLE connection are established in the Generic Attribute Profile (GATT).

3.4 LI-FI (LIGHT FIDELITY)

- Light Fidelity is referred to as Li-Fi. The German physicist Harald Haas first presented the concept in 2011 at the Visible Light Communication TED (Technology, Entertainment, Design) Global Talk (VLC).
- Light emitting diodes (LEDs) are used in the wireless optical networking technology known as Li-Fi to transmit data. The term "Li-Fi" refers to visible light communication (VLC) technology that complies with the IEEE standard IEEE 802.15.7 and employs light as a medium to deliver high-speed communication in a manner similar to Wi-Fi. Similar to Wi-IEEE Fi's 802.11 standard, the IEEE 802.15.7 is a high-speed, bidirectional, and fully networked wireless communication technology standard. The following Fig. 3.5 shows the concept of Li-Fi
- At the downlink transmitter, a light bulb is used to implement it. Since it only uses light, the light bulb typically glows at a constant current supply. However, quick and subtle variations in current can be made to produce optical outputs, making it easily applicable in places like airplanes, hospitals, and other places where radio frequency communication is frequently problematic.

- Simple operation involves transmitting a digital 1 when the LED is on and a digital 0 when it is off. Since the LED can be swiftly turned on and off, good opportunities for data transmission result. Therefore, all that is needed is a controller that encodes data into some LEDs so that they flicker in accordance with the data we wish to transmit. Your lamp can process more data the more LEDs there are in it.

(1C5)**Fig. 3.4.1 : Li-Fi Working**

3.4.1 Applications of Li-Fi

1. Health technologies
2. Airlines
3. Power Plant
4. Under sea working
5. GPS usage

3.4.2 Advantages of Li-Fi Over Wi-Fi

1. High speed connectivity of the rate of 500mbps.
2. Li- Fi uses light rather than radio frequency signals so are intolerant to disturbances.
3. VLC could be used safely in aircraft without affecting airlines signals.
4. Integrated into medical devices and in hospitals as this technology doesn't deal with radio waves, so it can easily be used in all such places where Bluetooth, infrared, Wi-Fi and internet are broadly in use.
5. Under water in sea Wi-Fi does not work at all but light can be used and hence undersea explorations are good to go now with much ease.
6. There are billions of bulbs worldwide which just need to be replaced with LED's to transmit data.
7. Security is a side benefit of using light for data transfer as it does not penetrate through walls.
8. On highways for traffic control applications like where Cars can have LED based headlights, LED based backlights, and they can communicate with each other and prevent accidents.

9. Using this Technology worldwide every street lamp would be a free data access point.
10. The issues of the shortage of radio frequency bandwidth may be sorted out by Li-Fi.

3.5 WPAN (WIRELESS PERSONAL AREA NETWORK)

- Wireless personal area networks, or WPANs, are types of personal area networks that connect devices to one another and are centred on the workspace of an individual.
- A wireless personal area network often employs a technology that allows communication over a relatively limited range of about 10 metres. One such technology is Bluetooth, which served as the foundation for the development of the new IEEE 802.15 standard.
- A WPAN could be used for a variety of purposes, including connecting all the common computing and communication devices that many people have on their desks or carry around with them today, or it could have a more niche use, such as enabling communication between the surgical team and the surgeon during an operation.
- Connecting is a crucial idea in WPAN technology. The ideal situation would be for any two WPAN-equipped devices to be able to communicate as if they were physically connected by a cable when they are in close proximity to one another (within a few metres) or when they are a short distance from a central server.
- The capability of each device to selectively lock out other devices, limiting superfluous interference or unwanted access to information, is another crucial aspect.
- The goal is to enable smooth operation across systems and appliances in the home or workplace. If two devices are physically close to one another, they can connect to each other and other devices in the same WPAN.
- Additionally, WPANs will be connected globally. So, for instance, an archaeologist working on location in Greece may use a PDA to immediately access databases at the University of Minnesota in Minneapolis and send research to those databases.

3.6 802.15 STANDARD BLUETOOTH:

- One of the most popular short-range wireless communication standard.
- Known as IEEE 802.15.1, now maintained by SIG (Special Interest Group)

Due to its numerous uses in audio devices like headsets, mobile phones, home stereos, MP3 players, laptops, desktop computers, tablets, and more, Bluetooth has now become a part of our daily life. If two devices are Bluetooth compliant, one can send data (meeting schedules, phone numbers), audio, graphic pictures, and video between them. The detailed Bluetooth standards are described in the IEEE 802.15.1 standard.

Bluetooth Network classified as follows

(1C6)**Fig. 3.6.1 : Bluetooth Architectures**

- Numerous bluetooth users make up the Bluetooth network. Piconet and scatternet are the two different types of network topologies used in Bluetooth. One master and one slave, as well as one master and several slaves, make up a piconet.
- The piconet can only have a maximum of 7 active slaves. A piconet, or small network, will therefore have a maximum of 8 devices talking with one another. Slaves can only transmit when the master Bluetooth device requests it. In parking state, there will be roughly 255 slaves.
- The master polls the active slaves to send data. Every station will receive an 8-bit parking address. In one piconet, 255 parked slaves are possible. In just 2 milliseconds, the parked station can connect.
- In just 2 milliseconds, the parked station can connect. The remaining stations have additional time to join. Within the range of the bluetooth radio, there are about 10 such piconets.
- Scatternet is a collection of many piconets. A device may take part in several piconets. It will timeshare and require synchronisation with the current piconet's master.
- Data rates based on various versions, ranging from 720 kbps to around 24 Mbps, are supported. Depending on the Bluetooth power class that is available, it will have a range of coverage of between 1 and 100 metres.

3.6.1 Bluetooth Applications

Following are few of the many Bluetooth applications :

1. One can receive and make call using Bluetooth compliant wireless headset.
2. Operate computer using mouse/keyboard and take print outs wirelessly eliminating cables.
3. Home automation

3.6.2 Bluetooth Protocol Stack

(1C7)Fig. 3.6.2 : Bluetooth Protocol Stack

3.6.3 Functions of the Core Protocols

- **Radio :** This is a physical layer equivalent protocol that lays down the physical structure and specifications for transmission of radio waves. It defines air interface, frequency bands, frequency hopping specifications and modulation techniques.
- **Baseband :** This protocol takes the services of radio protocol. It defines the addressing scheme, packet frame format, timing, and power control algorithms.
- **Link Manager Protocol (LMP) :** LMP establishes logical links between Bluetooth devices and maintains the links for enabling communications. The other main functions of LMP are device authentication, message encryption, and negotiation of packet sizes.
- **Logical Link Control and Adaptation Protocol (L2CAP) :** L2CAP provides adaption between upper layer frame and baseband layer frame format. L2CAP provides support for both connection-oriented as well as connectionless services.
- **Service Discovery Protocol (SDP) :** SDP takes care of service-related queries like device information so as to establish a connection between contending Bluetooth devices.

3.7 IEEE 802.15.4

A low-cost, low-data-rate wireless access technology for powered or battery-operated devices is IEEE 802.15.4. This explains the operation of low-rate wireless personal area networks (LR-WPANs).

3.7.1 IEEE 802.15. Protocol Stacks include:

1. **Standardization and alliances : It specifies low-data-rate PHY and MAC layer requirements for wireless personal area networks (WPAN).**

   (i) **ZigBee:** ZigBee is a low-rate task group 4 in the Personal Area Network task group. It is a form of home networking technology. A technological standard called ZigBee was developed for managing and sensing the network. Since ZigBee is the Personal Area Network of Task Group 4, it was developed by the Zigbee Alliance and is based on IEEE 802.15.4.

(ii) **6LoWPAN** : A number of applications, including wireless sensor networks, employ the 6LoWPAN standard. The term IPv6 over Low power Wireless Personal Area Networks is derived from the fact that this type of wireless sensor network distributes data as packets and utilises IPv6.

(iii) **Wireless HART** : A time-synchronized and self-organizing architecture is used in this wireless sensor network technology.

(iv) **Thread** : Thread is an IPv6-based networking protocol for low-power Internet of Things devices in IEEE 802.15. 4-2006 wireless mesh network. Thread is independent.

2. **Physical Layer**

- A wide variety of PHY possibilities in ISM bands, from 2.4 GHz to sub-GHz frequencies, are made possible by this standard. 20 kilobits per second, 40 kilobits per second, 100 kilobits per second, and 250 kilobits per second are among the data transmission speeds supported by IEEE 802.15.4.
- The main design presupposes a 10-meter range and a 250 kilobits per second data rate. Even lower data rates are feasible to further decrease power consumption.

3. **MAC layer**

By deciding which devices in the same area will share the given frequencies, the MAC layer establishes links to the PHY channel. At this layer, data packet scheduling and routing are also managed.

4. **Topology**

IEEE 802.15.4-based networks can be designed with a star, peer-to-peer, or mesh topology. Mesh networks link a lot of nodes together. This makes it possible for nodes that are out of communication range to communicate with one another and relay information using intermediary nodes.

5. **Security**

The Advanced Encryption Standard (AES) with a 128-bit key length is the primary encryption method used by the IEEE 802.15.4 standard to protect data.

3.8 ZIGBEE

- Zigbee is now very well-liked for wireless applications requiring modest data rates. Zigbee technology is utilised in a variety of applications, including smart energy, healthcare, and home automation.
- Zigbee devices are used in smart energy applications to monitor and regulate energy and water use, which benefits consumers by enabling them to conserve both resources and money. It can connect an infinite number of health monitoring devices in the medical area, among many others. Switches,

dimmers, occupancy sensors, and load controllers are all used in home automation to regulate domestic lights.

- It operates on two frequencies: 868/915 MHz and 2450 MHz. Data speeds in the 868/915 frequency range from 20 to 40 kbps, whereas those in the 2450 MHz spectrum are around 250 kbps. Additionally, because zigbee end devices include a security layer, they can enter a sleep state to conserve battery life and protect information security.
- Zigbee network is comprised of coordinator(C), router(R) and end devices (E). Zigbee supports mesh-routing.

(1C8)**Fig. 3.8.1 : Zig-bee Network**

Coordinator

- Always first coordinator need to be installed for establishing zigbee network service, it starts a new PAN (Personal Area Network), once started other zigbee components viz. router(R) and End devices(E) can join the network(PAN).
- It is responsible for selecting the channel and PAN ID.
- It can assist in routing the data through the mesh network and allows join request from R and E.
- It is mains powered (AC) and support child devices.
- It will not go to sleep mode.

Router

First router needs to join the network then it can allow other R & E to join the PAN. It is mains powered (AC) and support child devices.It will not go to sleep mode.

End Devices

It cannot allow other devices to join the PAN nor can it assist in routing the data through the network. It is battery powered and do not support any child devices. This may sleep hence battery consumption can be minimized to great extent.

3.9 Z-WAVE

- An important use for the wireless communications system Z-Wave is the automation of homes and businesses. It is a mesh network that communicates from device to device using low-energy radio waves, enabling wireless control of smart home appliances like smart lights, security systems, thermostats, sensors, smart door locks, and garage door openers.
- A Z-Wave system can be controlled remotely from a smart phone, tablet, or computer, as well as locally through a smart speaker, wireless keyboard, or wall-mounted panel, with a Z-Wave gateway or central control device acting as both the hub and controller, similar to other protocols and systems aimed at the residential, commercial, MDU, and building markets. Through its collaboration, Z-Wave enables application layer interoperability across home control systems made by various manufacturers.

3.9.1 Z-Wave Network

- Slaves and controllers make up the Z-wave network. There is one primary controller and multiple subordinate controllers. The nodes in a Z-wave network that send out control commands are known as controller devices.
- Additionally, it broadcasts the commands to other nodes. The slave devices are the nodes that respond to commands based on them and also carry them out. The directives are also forwarded to other network nodes by slave nodes. As a result, the controller can establish connection with nodes that are outside of the radio frequency range.

(1C9)**Fig. 3.9.1 : Z Network**

- **Controllers :** A controller device will have full routing table for this mesh network and it will host it. Hence controller can communicate with all the nodes of z-wave network.
- **Slaves :** The slave devices/nodes in z-wave network receive the commands and performs action based on the commands. These slave nodes are unable to transmit information directly to the other slave nodes or controllers unless they are instructed to do so in the commands. The slave nodes do not compute routing tables. They can store routing tables. They will act as a repeater.
- **Home ID :** The z-wave protocol uses Home ID field to separate the networks from each other. It is 32 bit unique identifier which will be pre-programmed in all the controller devices. At the start, all

the slave nodes will have Home ID value as zero. All the slave devices need Home ID value in order to communicate in the z-wave network. This will be communicated to all by the controller. Controllers exchange Home ID which makes it possible for more than one controller to control slave nodes.

- **Node ID :** This node ID is 8 bit value. Similar to Home ID, they are also assigned to slave nodes by controller. Node ID's are used in order to address individual nodes in a z-wave network. These Node ID's are unique within a network defined by a unique Home ID.

3.10 NARROW BAND IOT

- A large variety of new IoT devices and services are made possible by the standards-based low power wide area (LPWA) technology known as narrowband-Internet of things (NB-IoT). In deep coverage, NB-IoT dramatically increases spectrum efficiency, system capacity, and user device power consumption. A variety of use cases can accommodate battery life of more than 10 years.
- The challenging requirements of expanded coverage - rural and deep interiors - and ultra-low device complexity are met by new physical layer signals and channels. The NB-IoT modules' initial cost is anticipated to be similar to that of GSM/GPRS. Although the underlying technology is considerably simpler than GSM/GPRS today, its cost is anticipated to drop significantly as demand rises.
- All significant manufacturers of mobile devices, chipsets, and modules support NB-IoT, which can coexist alongside 2G, 3G, and 4G mobile networks. The security and privacy aspects of mobile networks, such as support for user identity secrecy, entity authentication, confidentiality, data integrity, and mobile equipment identification, are also advantageous to it. The initial NB-IoT commercial launches have been accomplished, and the global rollout is anticipated to begin in 2017 or 2018.

3.11 INTERNET PROTOCOL (IP)

The fourth version of the Internet Protocol is called Internet Protocol version 4 (IPv4) (IP). The Internet and other packet-switched networks use it as one of its primary core protocols for internetworking. The first production-ready release of IPv4 was made available on the SATNET in 1982 and the ARPANET in January 1983.

- Internet Protocol is connectionless and unreliable protocol. It ensures no guarantee of successfully transmission of data.
- In order to make it reliable, it must be paired with reliable protocol such as TCP at the transport layer.
- Internet protocol transmits the data in form of a datagram as shown in the following diagram:

(1C10)**Fig. 3.11.1 : IP Header Format**

- **Version :** The first header field in an IP packet is the four-bit version field. For IPv4, this is always equal to 4.
- **Internet Header Length (IHL) :** The IPv4 header is variable in size due to the optional 14th field (options). The IHL field contains the size of the IPv4 header; it has 4 bits that specify the number of 32-bit words in the header. The minimum value for this field is 5which indicates a length of 5 × 32 bits = 160 bits = 20 bytes. As a 4-bit field, the maximum value is 15; this means that the maximum size of the IPv4 header is 15 × 32 bits = 480 bits = 60 bytes.
- **Differentiated Services Code Point (DSCP) :** Originally defined as the type of service (ToS), this field specifies differentiated services (DiffServ) per RFC 2474.Real-time data streaming makes use of the DSCP field. An example is Voice over IP (VoIP), which is used for interactive voice services.
- **Explicit Congestion Notification (ECN) :** This field is defined in RFC 3168 and allows end-to-end notification of network congestion without dropping packets. ECN is an optional feature available when both endpoints support it and effective when also supported by the underlying network.
- **Total Length :** This 16-bit field defines the entire packet size in bytes, including header and data. The minimum size is 20 bytes (header without data) and the maximum is 65,535 bytes. All hosts are required to be able to reassemble datagrams of size up to 576 bytes, but most modern hosts handle much larger packets. Links may impose further restrictions on the packet size, in which case datagrams must be fragmented. Fragmentation in IPv4 is performed in either the sending host or in routers. Reassembly is performed at the receiving host.
- **Identification :** This field is an identification field and is primarily used for uniquely identifying the group of fragments of a single IP datagram. Some experimental work has suggested using the ID field for other purposes, such as for adding packet-tracing information to help trace datagrams with spoofed source addresses, but RFC 6864 now prohibits any such use.

- **Flags :** A three-bit field follows and is used to control or identify fragments. They are (in order, from most significant to least significant):
  - bit 0: Reserved; must be zero.
  - bit 1: Don't Fragment (DF)
  - bit 2: More Fragments (MF)

  If the DF flag is set, and fragmentation is required to route the packet, then the packet is dropped. This can be used when sending packets to a host that does not have resources to perform reassembly of fragments. It can also be used for path MTU discovery, either automatically by the host IP software, or manually using diagnostic tools such as ping or traceroute.

  For unfragmented packets, the MF flag is cleared. For fragmented packets, all fragments except the last have the MF flag set. The last fragment has a non-zero Fragment Offset field, differentiating it from an unfragmented packet.

- **Fragment offset :** This field specifies the offset of a particular fragment relative to the beginning of the original unfragmented IP datagram in units of eight-byte blocks. The first fragment has an offset of zero. The 13 bit field allows a maximum offset of (213 – 1) × 8 = 65,528 bytes, which, with the header length included (65,528 + 20 = 65,548 bytes), supports fragmentation of packets exceeding the maximum IP length of 65,535 bytes.

- **Time to live (TTL) :** An eight-bit time to live field limits a datagram's lifetime to prevent network failure in the event of a routing loop. It is specified in seconds, but time intervals less than 1 second are rounded up to 1. In practice, the field is used as a hop count—when the datagram arrives at a router, the router decrements the TTL field by one. When the TTL field hits zero, the router discards the packet and typically sends an ICMP time exceeded message to the sender.

  The program traceroute sends messages with adjusted TTL values and uses these ICMP time exceeded messages to identify the routers traversed by packets from the source to the destination.

- **Protocol :** This field defines the protocol used in the data portion of the IP datagram. IANA maintains a list of IP protocol numbers as directed by RFC 790.

- **Header checksum :** The 16-bit IPv4 header checksum field is used for error-checking of the header. When a packet arrives at a router, the router calculates the checksum of the header and compares it to the checksum field. If the values do not match, the router discards the packet. Errors in the data field must be handled by the encapsulated protocol. Both UDP and TCP have separate checksums that apply to their data.

  When a packet arrives at a router, the router decreases the TTL field in the header. Consequently, the router must calculate a new header checksum.

- **Source address :** This field is the IPv4 address of the sender of the packet. Note that this address may be changed in transit by a network address translation device.

- **Destination address :** This field is the IPv4 address of the receiver of the packet. As with the source address, this may be changed in transit by a network address translation device.

3.11.1 Transmission Control Protocol

TCP delivers end-to-end packet transmission and is a connection-oriented protocol. It serves as the foundation for relationship. It exhibits the following key features:

- Transmission Control Protocol (TCP) corresponds to the Transport Layer of OSI Model.
- TCP is a reliable and connection oriented protocol.
- TCP offers:
  - Stream Data Transfer.
  - Reliability.
  - Efficient Flow Control
  - Full-duplex operation.
  - Multiplexing.
- TCP offers connection oriented end-to-end packet delivery.
- TCP ensures reliability by sequencing bytes with a forwarding acknowledgement number that indicates to the destination the next byte the source expect to receive.
- It retransmits the bytes not acknowledged with in specified time period.

TCP Services

TCP offers following services to the processes at the application layer:

1. Stream Delivery Service	2. Sending and Receiving Buffers
3. Bytes and Segments	4. Full Duplex Service
5. Connection Oriented Service	6. Reliable Service

1. **Stream Deliver Service :** TCP protocol is stream oriented because it allows the sending process to send data as stream of bytes and the receiving process to obtain data as stream of bytes.
2. **Sending and Receiving Buffers :** It may not be possible for sending and receiving process to produce and obtain data at same speed, therefore, TCP needs buffers for storage at sending and receiving ends.
3. **Bytes and Segments :** The Transmission Control Protocol (TCP), at transport layer groups the bytes into a packet. This packet is called segment. Before transmission of these packets, these segments are encapsulated into an IP datagram.
4. **Full Duplex Service :** Transmitting the data in duplex mode means flow of data in both the directions at the same time.
5. **Connection Oriented Service :** TCP offers connection oriented service in the following manner:

   (i) TCP of process-1 informs TCP of process – 2 and gets its approval.

(ii) TCP of process – 1 and TCP of process – 2 and exchange data in both the two directions.

(iii) After completing the data exchange, when buffers on both sides are empty, the two TCP's destroy their buffers.

6. **Reliable Service :** For sake of reliability, TCP uses acknowledgement mechanism.

TCP Header Format

A TCP header consists of data bytes to be sent and a header that is added to the data by TCP as shown

(1C10)**Fig. 3.11.2 : TCP Header Format**

The header of a TCP segment can range from 20-60 bytes. 40 bytes are for options. If there are no options, a header is 20 bytes else it can be of upmost 60 bytes.

Header fields

- **Source Port Address –** A 16-bit field that holds the port address of the application that is sending the data segment.
- **Destination Port Address –** A 16-bit field that holds the port address of the application in the host that is receiving the data segment.
- **Sequence Number –** A 32-bit field that holds the sequence number, i.e, the byte number of the first byte that is sent in that particular segment. It is used to reassemble the message at the receiving end of the segments that are received out of order.
- **Acknowledgement Number –** A 32-bit field that holds the acknowledgement number, i.e, the byte number that the receiver expects to receive next. It is an acknowledgement for the previous bytes being received successfully.
- **Header Length (HLEN) –** This is a 4-bit field that indicates the length of the TCP header by a number of 4-byte words in the header, i.e if the header is 20 bytes(min length of TCP header), then this field will hold 5 (because 5 x 4 = 20) and the maximum length: 60 bytes, then it'll hold the value 15(because 15 x 4 = 60). Hence, the value of this field is always between 5 and 15.

- **Control flags** – These are 6 1-bit control bits that control connection establishment, connection termination, connection abortion, flow control, mode of transfer etc. Their function is:

  URG: Urgent pointer is valid

  ACK: Acknowledgement number is valid (used in case of cumulative acknowledgement)

  PSH: Request for push

  RST: Reset the connection

  SYN: Synchronize sequence numbers

  FIN: Terminate the connection
- **Window size** – This field tells the window size of the sending TCP in bytes.
- **Checksum** – This field holds the checksum for error control. It is mandatory in TCP as opposed to UDP.
- **Urgent pointer** – This field (valid only if the URG control flag is set) is used to point to data that is urgently required that needs to reach the receiving process at the earliest. The value of this field is added to the sequence number to get the byte number of the last urgent byte.

3.12 6LOWPAN

- Every node in the low power wireless mesh network known as 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) has a unique IPv6 address. This enables the node to establish a direct connection utilizing open standards with the Internet.
- The concept behind 6LoWPAN is that low-power, low-processing-power devices should be able to participate in the Internet of Things and that the Internet Protocol can and should be applied to even the smallest devices.

3.12.1 Advantages of 6LoWPAN

1. It works great with open IP standard including TCP, UDP, HTTP, COAP, MATT and web-sockets.
2. It offers end-to-end IP addressable nodes. There's no need for a gateway, only a router which can connect the 6LoWPAN network to IP.
3. It supports self-healing, robust and scalable mesh routing.
4. Offers one-to-many & many-to-one routing.
5. The 6LoWPAN mesh routers can route data to others nodes in the network.
6. In a 6LowPAN network, leaf nodes can sleep for a long duration of time.
7. It also offers thorough support for the PHY layer which gives freedom of frequency band & physical layer, which can be used across multiple communication platforms like Ethernet, WI-Fi, 802.15.4 or Sub-1GHz ISM with interoperability at the IP level.

(1C11)**Fig. 3.12 1 : LoWPAN Protocol Stack**

3.12.2 6LoWPAN Application Areas

- **Automation :** There are enormous opportunities for 6LoWPAN to be used in many different areas of automation.
- **Industrial monitoring :** 6LoWPAN has a lot of potential in automated industries and industrial plants. Automating routine tasks can result in significant savings. Furthermore, 6LoWPAN can link to the cloud, opening up a wide range of possibilities for data monitoring and analysis.
- **Smart Grid :** Smart grids enable smart meters and other devices to build a micro mesh network. They are able to send data back to the grid operator's monitoring and billing system using the IPv6.
- **Smart Home:** By connecting your home IoT devices using IPv6, it is possible to gain distinct advantages over other IoT systems.

3.13 WLAN AND WAN

The term "wireless LANs" refers to wireless computer networks that connect devices inside a constrained area without the usage of wires (Local Area Network). The restricted region includes places like homes, schools, campuses, office buildings, train platforms, etc. where users linked by wireless LANs can move about.

Most WLANs are based upon the standard IEEE 802.11 standard or WiFi.

Components of WLANs

The components of WLAN architecture as laid down in IEEE 802.11 are

- **Stations (STA)** – Stations comprises of all devices and equipment that are connected to the wireless LAN. Each station has a wireless network interface controller. A station can be of two types –
  - Wireless Access Point (WAP or AP)
  - Client

- **Basic Service Set (BSS)** – A basic service set is a group of stations communicating at the physical layer level. BSS can be of two categories :
  - Infrastructure BSS
  - Independent BSS
- **Extended Service Set (ESS)** – It is a set of all connected BSS.
- **Distribution System (DS)** – It connects access points in ESS.

(1C12)**Fig. 3.13.1 : Architectures of WLAN**

3.13.1 Types of WLANS

WLANs, as standardized by IEEE 802.11, operates in two basic modes, infrastructure, and ad hoc mode.

- **Infrastructure Mode :** Mobile devices or clients connect to an access point (AP) that in turn connects via a bridge to the LAN or Internet. The client transmits frames to other clients via the AP.
- **Ad Hoc Mode** : Clients transmit frames directly to each other in a peer-to-peer fashion.

3.13.2 Frame Format of IEEE 802.11

The main fields of a frame of wireless LANs as laid down by IEEE 802.11 are :

- **Frame Control :** It is a 2 bytes starting field composed of 11 subfields. It contains control information of the frame.
- **Duration :** It is a 2-byte field that specifies the time period for which the frame and its acknowledgment occupy the channel.
- **Address fields :** There are three 6-byte address fields containing addresses of source, immediate destination, and final endpoint respectively.
- **Sequence :** It a 2 bytes field that stores the frame numbers.
- **Data** : This is a variable-sized field that carries the data from the upper layers. The maximum size of the data field is 2312 bytes.
- **Check Sequence :** It is a 4-byte field containing error detection information.

(1C13)**Fig. 3.13.2 : 802.11 Frame Format**

3.13.3 Advantages of WLANs

(1) They provide clutter-free homes, offices and other networked places.

(2) The LANs are scalable in nature, i.e. devices may be added or removed from the network at greater ease than wired LANs.

(3) The system is portable within the network coverage. Access to the network is not bounded by the length of the cables.

(4) Installation and setup are much easier than wired counterparts.

(5) The equipment and setup costs are reduced.

3.13.4 Disadvantages of WLANs

(1) Since radio waves are used for communications, the signals are noisier with more interference from nearby systems.

(2) Greater care is needed for encrypting information. Also, they are more prone to errors. So, they require greater bandwidth than the wired LANs.

(3) WLANs are slower than wired LANs.

3.13.5 Wide Area Network

- A computer network known as a wide area network (WAN) is one that spans a substantial geographic area, such as an entire country, continent, or even the entire world. The technologies used by WAN allow data, picture, audio, and video to be transmitted over vast distances and between various LANs and MANs.
- **The distinguishing features of WAN are :**
  - WANs have a large capacity, connecting a large number of computers over a large area, and are inherently scalable.
  - They facilitate the sharing of regional resources.
  - They provide uplinks for connecting LANs and MANs to the Internet.

- Communication links are provided by public carriers like telephone networks, network providers, cable systems, satellites etc.
- Typically, they have low data transfer rate and high propagation delay, i.e. they have low communication speed.
- They generally have a higher bit error rate.

- **Example of WAN**
  - The Internet
  - 4G Mobile Broadband Systems
  - A network of bank cash dispensers.

3.14 LONG RANGE COMMUNICATION SYSTEM AND PROTOCOL:CELLULAR CONNECTIVITY LTE, LTE-A, LORA AND LORAWAN

3.14.1 Long Term Evolution

- The project name for the creation of a high performance air interface for cellular mobile communication systems is LTE (Long Term Evolution). It is the final phase of the transition to the fourth generation (4G) of radio technology intended to boost the speed and capacity of mobile phone networks. LTE is marketed as 4G while the earlier generations of mobile telecommunication networks are referred to as 2G or 3G.
- LTE (Long-Term Evolution) is a fourth-generation (4G) wireless standard that provides increased network capacity and speed for cellphones and other cellular devices compared with third-generation (3G) technology.
- LTE offers higher peak data transfer rates than 3G, initially up to 100 Mbps downstream and 30 Mbps upstream. It provides reduced latency, scalable bandwidth capacity and backward-compatibility with the existing Global System for Mobile communication (GSM) and Universal Mobile Telecommunications Service (UMTS) technology. The subsequent development of LTE-Advanced (LTE-A) yielded peak throughput on the order of 300 Mbps.
- LTE has a direct role in the development of the current 5G standard, called 5G New Radio. Early 5G networks, referred to as non-standalone 5G (NSA 5G), require a 4G LTE control plane to manage 5G data sessions. NSA 5G networks can be deployed and supported by the existing 4G network framework, lowering capital and operating expenses for operators rolling out 5G.

Why is LTE called 'Long-Term Evolution'?

- The 3rd Generation Partnership Project (3GPP) developed LTE. The standard was described as the next step in the progression of mobile telecommunications and follows the 2G GSM and 3G UMTS specifications. LTE is commonly marketed as 4G LTE.LTE did not originally qualify as true 4G.

- The International Telecommunication Union (ITU) initially defined 4G as a cellular standard that would deliver data rates of 1 Gbps to a stationary user and 100 Mbps to a user on the move. In December 2010, the ITU softened its stance, applying 4G to LTE, as well as several other wireless standards.

3.14.2 LTE-A

- LTE-A stands for LTE-Advanced. It is a standard for mobile communication that is one generation beyond LTE (Long Term Evolution). Whereas LTE was a 3G communication standard, LTE-A is a 4G or fourth generation communication standard.
- The benefits of a fourth generation communication network are many. Perhaps most simply, LTE-A offers faster speeds than 3G. As defined by the International Mobile Telecommunications-Advanced Standard, 4G must offer a nominal data rate of 100 Mbit/s when a user is physically moving at high speeds relative to a data station, and 1 Gbit/s when the user and station are fixed relative to one another. Additionally, LTE-A allows for global roaming, smooth handover between networks, and interoperability with existing wireless standards.
- For wireless communication, LTE-A or 4G LTE is the current state-of-the-art.

LTE-A features

- As one would expect, a 4G network offers several advantages over a 3G network. In order to be classified as 4G, a network must meet certain criteria, as governed by the International Telecommunications Union.
- Requirements for a 4G communication standard include:
  - All-Internet Protocol (IP) packet switched network (for increased network efficiency)
  - Interoperability with existing wireless standards
  - Specific nominal data rates for mobile and stationary users
  - Network resources are dynamically shared in order to support more simultaneous users per cell
  - Scalable channel bandwidth up to 40 MHz
  - Peak link spectral efficiency of 15 bit/s/Hz for downlink, and 6.75 bit/s/Hz for uplinks
  - Seamless connectivity with smooth handovers across networks
  - Global roaming with universal connectivity
  - Service sufficient for multimedia support
- Though 4G data speeds are greater than 3G data speeds, it is worth noting that Megabits (Mbit) are different than Megabytes (MB) – the two are commonly mistaken for one another.
- A Megabyte refers to the size of a digital file; Megabit refers to upload and download speeds of digital files. A Megabit is 1/8th as large as a Megabyte. To download a 1MB file in 1 second, your network connection must be 8Mbps (or Mbit/s).

3.14.3 LoRa

- LoRa is a radio modulation technique that is essentially a way of manipulating radio waves to encode information using a chirped (chirp spread spectrum technology), multi-symbol format. LoRa as a term can also refer to the systems that support this modulation technique or the communication network that IoT applications use.
- The main advantages of LoRa are its long-range capability and its affordability. A typical use case for LoRa is in smart cities, where low-powered and inexpensive internet of things devices (typically sensors or monitors) spread across a large area send small packets of data sporadically to a central administrator.
- LoRa (short for long range) is a spread spectrum modulation technique derived from chirp spread spectrum (CSS) technology. Semtech's LoRa is a long range, low power wireless platform that has become the de facto wireless platform of Internet of Things (IoT). LoRa devices and networks such as the LoRaWAN® enable smart IoT applications that solve some of the biggest challenges facing our planet: energy management, natural resource reduction, pollution control, infrastructure efficiency, and disaster prevention.
- LoRa devices have amassed several hundred known uses cases for smart cities, homes and buildings, communities, metering, supply chain and logistics, agriculture, and more. With hundreds of millions of devices connected to networks in more than 100 countries and growing, LoRa is creating a smarter planet.

Key Features of LoRa

- **Long Range**

  Connects devices up to 30 miles apart in rural areas and penetrates dense urban or deep indoor environments
- **Low Power**

  Requires minimal energy, with prolonged battery lifetime of up to 10 years, minimizing battery replacement costs
- **Secure**

  Features end-to-end AES128 encryption, mutual authentication, integrity protection, and confidentiality
- **Geolocation**

  Enables GPS-free tracking applications, offering unique low power benefits untouched by other technologies
- **High Capacity**

  Supports millions of messages per base station, meeting the needs of public network operators serving large markets

- **Low Cost**

  Reduces infrastructure investment, battery replacement expense, and ultimately operating expenses

3.14.4 LoRaWAN

- LoRaWAN is a low-power, wide area networking protocol built on top of the LoRa radio modulation technique. It wirelessly connects devices to the internet and manages communication between end-node devices and network gateways.
- Usage of LoRaWAN in industrial spaces and smart cities is growing because it is an affordable long-range, bi-directional communication protocol with very low power consumption devices can run for ten years on a small battery. It uses the unlicensed ISM (Industrial, Scientific, Medical) radio bands for network deployments.
- An end device can connect to a network with LoRaWAN in two ways:
  - **Over-the-air Activation (OTAA) :** A device has to establish a network key and an application session key to connect with the network.
  - **Activation by Personalization (ABP) :** A device is hardcoded with keys needed to communicate with the network, making for a less secure but easier connection.

LoRaWAN has three different classes of end-point devices to address the different needs reflected in the wide range of applications:

Class A – Lowest power, bi-directional end-devices

- The default class which must be supported by all LoRaWAN end-devices, class A communication is always initiated by the end-device and is fully asynchronous. Each uplink transmission can be sent at any time and is followed by two short downlink windows, giving the opportunity for bi-directional communication, or network control commands if needed. This is an ALOHA type of protocol.
- The end-device is able to enter low-power sleep mode for as long as defined by its own application: there is no network requirement for periodic wake-ups. This makes class A the lowest power operating mode, while still allowing uplink communication at any time.
- Because downlink communication must always follow an uplink transmission with a schedule defined by the end-device application, downlink communication must be buffered at the network server until the next uplink event.

Class B – Bi-directional end-devices with deterministic downlink latency

- In addition to the class A initiated receive windows, class B devices are synchronized to the network using periodic beacons, and open downlink 'ping slots' at scheduled times. This provides the network the ability to send downlink communications with a deterministic latency, but at the expense of some additional power consumption in the end-device. The latency is programmable up to 128 seconds to suit different applications, and the additional power consumption is low enough to still be valid for battery powered applications.

Class C – Lowest latency, bi-directional end-devices:

- In addition to the class A structure of uplink followed by two downlink windows, class C further reduces latency on the downlink by keeping the receiver of the end-device open at all times that the device is not transmitting (half duplex). Based on this, the network server can initiate a downlink transmission at any time on the assumption that the end-device receiver is open, so no latency. The compromise is the power drain of the receiver (up to ~50mW) and so class C is suitable for applications where continuous power is available.

For battery powered devices, temporary mode switching between classes A & C is possible, and is useful for intermittent tasks such as firmware over-the-air updates.

3.15 MULTIPLE CHOICE QUESTIONS

Q. 3.1 An interconnected collection of piconet is called _____

(a) Scatternet (b) Micronet
(c) Mininet (d) Multinet

✓**Ans. : (a)**

Q. 3.2 Coordinator ZigBee devices act as the bridge between _____.

(a) Different networks (b) Different edge devices
(c) Different fog devices (d) All of the above

✓**Ans. : (a)**

Q. 3.3 Which of the following IEEE standards is followed by the physical and MAC layer protocols in ZigBee?

(a) IEEE 801.15.4 (b) IEEE 802.15.4
(c) IEEE 803.15.4 (d) IEEE 804.15.4

✓**Ans. : (b)**

Q. 3.4 IEEE 802.11g, has a data rate of _____ Mbps.

(a) 1 (b) 2 (c) 11 (d) 22

✓**Ans. : (d)**

Q. 3.5 Which bluetooth version enables low energy?

(a) Bluetooth 3.0 (b) Bluetooth 4.0
(c) Bluetooth 2.0 (d) Bluetooth 1.0

✓**Ans. : (c)**

Q. 3.6 What is the Access technique used by an LTE or LTE-A network?

(a) WCDMA (b) FDMA
(c) PDMA (d) OFDMA

✓**Ans. : (d)**

Q. 3.7 IEEE has defined the specifications for a wireless LAN called _____, which covers the physical and data link layers.

(a) ieee 802.3 (b) ieee 802.5
(c) ieee 802.11 (d) ieee 802.2

✓**Ans. : (c)**

Q. 3.8 What is header of datagram in IPv4_____

(a) 20 to 60 bytes (b) 20 to 80 bytes
(c) 20 to 40 bytes (d) 0 to 20 bytes

✓**Ans. : (a)**

Q. 3.9 Transmission control protocol _____

(a) is a connection-oriented protocol

(b) uses a three way handshake to establish a connection

(c) receives data from application as a single stream

(d) all of the mentioned

✓**Ans. : (d)**

Q. 3.10 What is Narrow Band Internet of Things (NB-IoT)?

(a) Low Power Wide Area (LPWA)

(b) GPRS

(c) GSM

(d) Edge

✓**Ans. : (a)**

Chapter Ends...

❑❑❑

MODULE 4

Edge to Cloud Protocol

Syllabus

HTTP, WebSocket, Platforms. HTTP - MQTT - .Complex Flows: IoT Patterns: Real-time Clients, MQTT, MQTT-SN, Constrained Application Protocol (CoAP), Streaming Text Oriented Message Protocol (STOMP), Advanced Message Queuing Protocol (AMQP), Comparison of Protocols.

4.1 Introduction 4-3

4.1.1 Edge-to-cloud platform 4-3

4.1.2 Characteristics of - edge-to-cloud platform 4-4

4.2 HTTP (HyperText Transport Protocol) 4-4

4.2.1 Advantages of HTTP 4-5

4.2.2 Disadvantages of HTTP 4-5

4.3 Websockets 4-5

4.3.1 Difference between HTTP and WebSockets 4-6

4.4 Platforms 4-7

4.4.1 The most common categories of IoT platforms are: 4-7

4.4.1.1 Cloud platforms 4-7

4.4.1.2 IoT Connectivity Platforms 4-7

4.5 IoT Patterns 4-9

4.5.1 Types of IoT patterns 4-9

4.5.1.1 Edge Provisioning Pattern 4-9

4.5.1.2 Edge Code Deployment Pattern 4-10

4.5.1.3 Edge Orchestration Pattern 4-11

4.5.1.4 Diameter of Things (DoT) Pattern 4-12

4.5.2 Real Time Clients 4-13

4.6 Protocols used in IoT Systems 4-13
4.6.1 Message Queueing Telemetry Transport (MQTT) Protocol 4-13
4.6.1.1 MQTT Components 4-14
4.6.1.2 Message Types used in MQTT 4-14
4.6.1.3 Applications of MQTT 4-15
4.6.1.4 Advantages and Disadvantages of MQTT 4-17
4.6.2 MQTT-SN (MQTT for Sensor Networks) 4-18
4.6.2.1 MQTT-SN Architecture and Topology 4-18
4.6.2.2 Differences between MQTT and MQTT-SN 4-20
4.6.3 Constrained Application Protocol (CoAP) 4-20
4.6.3.1 Layers of CoAP Protocol 4-21
4.6.3.2 CoAP has four messaging modes: 4-21
4.6.3.3 CoAP Message Format 4-23
4.6.3.4 CoAP Security Aspects 4-23
4.6.3.5 Advantages of COAP 4-24
4.6.3.6 Disadvantages of COAP 4-24
4.6.3.7 CoAP vs MQTT 4-24
4.6.4 STOMP 4-25
4.6.5 Advanced Message Queuing Protocol - AMQP 4-26
4.6.5.1 Components of AMQP 4-27
4.6.5.2 Advantages of AMQP 4-27
4.6.5.3 Disadvantages of AMQP 4-28
4.7 Protocol Summary and Comparison 4-28
4.8 Multiple Choice Questions 4-28
- **Chapter End...** 4-29

4.1 INTRODUCTION

Edge Computing

- In edge computing, data is collected and analyzed and stored at the edge i.e. devices connected to the IoT systems, instead of the centralized storage or on the cloud.
- This is done to reduce the response time and to save bandwidth.

Cloud Computing

- In cloud computing, the data collected at the nodes (or sensors or nodes) is further sent to the cloud and then it will be processed and stored in the cloud
- Then the processed data (information) will be provided to the distributed applications and devices.

Fig. 4.1.1 : Edge Computing and Cloud Computing

4.1.1 Edge-to-cloud platform

- An edge-to-cloud platform is designed to bring the cloud experience to all of an organization's apps and data, regardless of their location.
- It provides good user experience with security and allows organizations to capture new business opportunities with simplicity and scalability.
- As shown in Fig. 4.1.2, most applications store, manage, and analyze data on a centralized storage like - on a public cloud or private cloud environment.
- Traditional infrastructure and cloud computing can't meet the requirements for many real life applications.
- For example, in the case of **IoT** (Internet of Things) and **IoE** (Internet of Everything), a **highly available network** with **minimal latency** is required to process large amounts of data in real time, which is not possible on a traditional IT infrastructure.

- In such a scenario, edge computing becomes more important.

Fig. 4.1.2 : Edge-to-Cloud Platform

Advantages of Edge-to-Cloud Platform

(1) Data is processed at the edge devices (where data is actually collected), so it is not necessary to transfer data to the cloud for processing and analysis.

(2) This approach will reduce the load on both network and servers.

(3) As edge-to-cloud platform processes data in real time, with its faster response time, it is highly applicable in the field of IoT, particularly industrial IoT (IIoT).

4.1.2 Characteristics of-edge-to-Cloud Platform

- **Self-service :** Organizations must arrange resources for new projects, such as new virtual machines (VMs) or services for containers or Machine Learning Operations.
- **Rapidly scalable :** a platform must provide scaling-up and scaling-down facility of resources whenever required.
- **Pay-per-use :** Billing to the customer should be based on the actual amount of resources used.

4.2 HTTP (HYPERTEXT TRANSPORT PROTOCOL)

- HTTP is the character based protocol, used in the application layer.
- It is used by web pages to exchange data.

- It is also used with IoT devices to exchange the data, using REST (RE presentational State Transfer) principles.
- REST allows IoT devices to communicate with each other using the standard API.
- The device or the server which is using these REST principles, is called a RESTful device.
- HTTP protocol is reliable & operates at the top of the TCP.
- HTTP is a unidirectional protocol - in HTTP, request is always initiated by the client and response is always given by the server.
- HTTP protocol provides half-duplex communication. I.e. either client or server can send the messages at the same time.

4.2.1 Advantages of HTTP

1. **Reliability** - guaranteed and acknowledged message delivery.
2. **Easy Implementation** - HHTP can be used simply by connecting a device to the internet.
3. **Less use of Resources** - HTTP uses less memory and less computational power, because of less simultaneous connections

4.2.2 Disadvantages of HTTP

1. **More Power Required** - to establish a connection, to retain this connection and to transfer the textual data between the devices, more power is required
2. **IoT Device Complexity** - device require more memory and processing power to work with TCP and HTTP RESTful APIs
3. **Less Secure** - because, it is not using any encryption mechanism to encrypt its http request and response

4.3 WEBSOCKETS

- WebSockets were invented in 2008, by the World Wide Web Consortium (W3C)
- Websockets are used with HTTP at the application layer, and Transmission Control Protocol (TCP) at the transport layer.
- Using Websockets, server and client can push the messages at any time.
- Websocket is a bidirectional protocol - i.e. either client or server can initiate the request to send the message
- Websocket protocol is providing full-duplex communication. I.e. both client and server can send messages to each other simultaneously.

☞ Advantages of WebSockets

1. Allows two way communication
2. Allows to send and receive data, faster than HTTP
3. Send and receive data faster than AJAX

☞ Disadvantages of WebSockets

1. HTML5 compliant web browser is required
2. Like HTTP, it can't provide intermediary or edge caching
3. Difficult than HTTP

(1D1)Fig. 4.3.1 : Connection using HTTP and web socket

4.3.1 Difference between HTTP and WebSockets

Sr. No.	Parameter	HTTP	WebSockets
1	Communication	Half Duplex	Full Duplex
2	Messaging Pattern	Request Response	Bi directional
3	Persistent	Partly	Fully
4	Service Push	Not Supported	Core Feature
5	Supported Clients	Broad Support	Modern Languages and Clients

4.4 PLATFORMS

- The Internet of Things (IoT) connects devices remotely.
- An IoT platform communicates device sensors and data networks.
- An IoT platform is a set of components which are used by developers to deploy the applications, collect the data remotely, secure connections, manage sensors and connections between them and allow developers to create new applications..
- Using IoT platforms, data is collected from the devices which can be used by businesses.
- The IoT platform helps to understand the customers' needs and helps to develop the products accordingly

4.4.1 The most common categories of IoT platforms are

4.4.1.1 Cloud platforms

Allows rapid development of applications, by hiding the complexities of developing an IOT solution. Some of the IoT Cloud platform are:

- Microsoft Azure IoT Suite
- ThingWrox 8 IoT Platform
- AWS IoT Platform
- Google Cloud's IoT Platform
- IBM Watson IoT Platform
- CISCO IoT Cloud Connect
- Saleforce IoT Cloud
- Kaa IoT Platform
- Oracle IoT Platform
- Thingspeak IoT Platform
- GE Predix IoT Platform
- Hitachi Vantara
- PTC

Fig. 4.4.1 : Cloud platforms in IoT

4.4.1.2 IoT Connectivity Platforms

- Connectivity is a very important part of the IoT stack.
- In IoT all the devices need to be connected to the cloud or to the central data repository.

- This can be done by using - bluetooth, wi-fi and other cellular technologies like - 4G, 5G or NB-IoT.
- Connectivity platforms provide a single interface for deploying, monitoring, and managing all your devices around the world.
- These platforms include functionalities like - traffic monitoring, connectivity management, geolocation, device management, over-the-air updates, and device provisioning.
- Examples of IoT connectivity platforms are -
  - Curiosity by Sprint,
  - Jasper by Cisco,
  - IoT Accelerator by Ericsson,
  - Pelion by ARM.

1. IoT Device Platforms

- Device platforms provide hardware building blocks for developing IoT devices.
- IoT devices must have - low cost, longer battery life, better supply chain agreements, improved serviceability, etc.

Types of IoT device Platforms are -

Pre-product-market fit

- The goal of this stage is innovation.
- Quick prototypes are created to share with your customers.
- Examples -
  - Single-board computers like - Arduino or Raspberry Pi can be used
  - For industrial hardware - PXI or Compact RIO from National Instruments.
  - A laptop can be used to attach the sensors via USB or a PCI card.
  - A smartphone or tablet also can be used, as they already have a lot of sensors.

Product-market fit

- Development of customized IoT devices that are similar in functionality and form-factor of the final product.
- For this stage, the hardware components like Arduino, Beagle Bone, Raspberry Pi, or the OEM version of CompactRIO can be used as the core of your device.

2. IoT Analytics Platforms -

- IoT devices are used to collect the data and send it to the cloud. This data is analyzed on the cloud.
- Most of the Cloud platforms include analysis tools, which are enough for many applications.

- But if your application has additional requirements like visualization, data processing, artificial intelligence (A.I), or machine learning (ML), an IoT analytics platform helps to accelerate your IoT development.
- Examples of IoT analytics platforms are -
  - Watson by IBM
  - C3 AI
  - SparkCognition
  - UpTake

4.5 IOT PATTERNS

- IoT Pattern is set of reusable and abstract solutions which are used to model and build IoT applications
- In IoT patterns abstract design principles of similar and frequent requirements are tailored for IoT applications
- IoT patterns are used to create valuable, interesting, usable, and compatible edge applications

4.5.1 Types of IoT patterns

There are 4 types of IoT patterns

1. Edge Provisioning Pattern
2. Edge Code Deployment Pattern
3. Edge Orchestration Pattern
4. Diameter of Things (DoT) Pattern

4.5.1.1 Edge Provisioning Pattern

- **Context**
  - IoT devices are usually geographically distributed, which are hard to manage.
  - It is necessary to reconfigure devices.
- **Motivation**
  - Suppose, a system is designed for some purpose.
  - At some point, it is necessary to replace the technology stack and provide a new environment remotely.
  - New devices can be added and their runtime environment and applications can be provided quickly.

- **Problem**
  - How do the developers ensure all of their edge devices are started with a reliable environment, when needed?
  - How can they provide all the devices automatically all at once?
- **Example - Container-based virtualization**
  - Containers can be used for provisioning resources.
  - Because, they contain the code and all the required software configurations and the runtime environment.
  - Using containers, devices can access only the required layers and not the whole image.
  - Containers allow the devices to rollback to the latest or any working version of the image.

Fig. 4.5.1 : Container Based Virtualization

4.5.1.2 Edge Code Deployment Pattern

- **Context**
  - Maintainability is also a main factor while deploying code to remote IoT devices.
  - As developers modify or repair the code or fix some bugs
  - Developers expect to deploy this modified code to remote IoT devices immediately.
- **Motivation**
  - Suppose a system is designed to display advertisements on some billboards at many places in the region.
  - It is necessary to update the text or visuals in advertisements frequently or change the duration of the advertisement display.
  - It is necessary to update & deploy the code to all devices at once.
- **Problem**
  - How developers can deploy their code to many IoT devices quickly
  - How can developers configure the code without worrying about the process of build, test and release?

- **Example - Device Continuous Delivery Pipeline**
  - Instead of changing the whole application, it is beneficial to deliver the required changes only
  - This can be done, by minimizing the connectivity issues, in the constrained network.
  - The pipeline includes - application building, deployment of it, testing, releasing and distributing it to edge devices.
  - Devices asks periodically to the central registry / hub for new versions or the server notifies to the devices about the release of a new image version

Fig. 4.5.2 : Device Continuous Delivery Pipeline

4.5.1.3 Edge Orchestration Pattern

- **Context**
  - By allowing the number of devices connected through the edge, the nodes in the cluster can - check their health state, their services state to reconfigure them.
  - This allows the IoT devices to accept the nodes remotely and quickly.
- **Motivation**
  - Consider, a system is designed to display advertisements on some billboards in a region.
  - And the devices are available to control each billboard.
  - Developers must be able - to check the state and health status of the devices, manage the devices, check their services, change the runtime configuration of the devices, and execute services in the cluster or on some devices.
- **Problem**
  - How can IoT devices handle their configurations as nodes of a cluster remotely?
  - How can edge cluster nodes discover services?
- **Example of - Automated IoT Edge Cluster Management**
  - The architecture should not have a single point of failure.
  - Nodes must describe their roles and services, and they must be able to discover each other.

- Any node can be the coordinator. If the coordinator node fails, a new coordinator can be elected in the cluster by the other nodes.

Fig. 4.5.3 : Automated IoT Edge Cluster Management

4.5.1.4 Diameter of Things (DoT) Pattern

- **Context**
  - the system of billing for the services can vary based on applied business models
  - These billing system can be - event-based, time-based, prepaid or pay-per-use
- **Motivation**
  - an IoT platform consisting of devices and services is available
  - Billing of using these devices and services, can be based on the real usage of the client.
- **Problem**
  - How can IoT service providers monitor and meter the actual usage of IoT services in real-time?
  - How the resource usage and service usage of an IoT application can be charged against a specific user balance?
  - How to collect information about usage of the resources for the billing ?
- **Example of - Diameter of Things (DoT) Pattern**

  Metering Patterns
  - **Timers :** measure the duration of a particular service used.
  - **Counters :** Measure number of calls and amount of data.

Fig. 4.5.4 : Metering Framework

4.5.2 Real Time Clients

- **An important pattern of IoT is -** to sense data through the sensors and make it available to users in real time, such as with smart home applications, security applications and alerts.
- **For example -** An Intrusion Detection System, is a real time application of IOT. Which senses the real time data and submits it to the system for further processing and generates the output signal as per the configuration done in the IoT system.

4.6 PROTOCOLS USED IN IOT SYSTEMS

4.6.1 Message Queueing Telemetry Transport (MQTT) Protocol

- MQTT is a machine to machine IoT connectivity protocol.
- It is defined by the standard body MQTT.org (mqtt.org), as ISO standard (ISO/IEC PRF 20922).
- It is a publish-subscribe-based lightweight and very simple, real time messaging protocol used with the TCP Protocol
- MQTT was introduced by IBM in 1999 and standardized by OASIS **(Organization for the Advancement of Structured Information Standards)** in 2013.
- It was designed for constrained devices and low-bandwidth, high-latency networks.
- It provides connectivity between applications & middle wares and networks & communications.
- In MQTT, the publish subscribe message pattern is controlled by a MESSAGE BROKER

- Messages are updated and distributed by the message broker when the client is subscribed to a TOPIC.
- It is designed for - Remote connections and Small-code footprint
- It provides faster data transmission, like WhatsApp or facebook messenger provides a faster delivery.

4.6.1.1 MQTT Components

As shown in Fig. 4.6.1, components of MQTT are -

- **Publishers** - These are lightweight sensors (for ex. Temperature Sensor), which transmits (sends) the messages
- **Subscribers** - These are the applications or receivers (for ex. Smartphone or PC), which receives the data from the publisher and uses this data.
- **Brokers** - these are used to connect - publishers and subscribers.

(102)**Fig. 4.6.1 : MQTT Components**

4.6.1.2 Message Types used in MQTT

- **Connect :** Used by the "Client A" or Sender to send connection request to the broker
- **Disconnect** : Waits for the MQTT client to finish any work it must do, and for the TCP/IP session to disconnect.
- **Publish :** Used by the "Client A" or Sender to publish messages to the Broker
- **Subscribe :** Used by the "Client B" or Receiver to receive messages from the Broker

(1D3)**Fig. 4.6.2 : Example of an MQTT connection with connect, publish/subscribe, and disconnect.**

4.6.1.3 Applications of MQTT

- MQTT is used by Microsoft Azure IoT Hub, as its main protocol for telemetry messages.
- MQTT is used in Facebook Messenger for online chat.
- MQTT is used in the EVRYTHNG IoT platform, as an M2M protocol for millions of connected products.
- With MQTT, Amazon Web Services uses Amazon IoT.
- Due to its lightweight properties MQTT works well for applications involving remote monitoring, including the following:
- synchronization of sensors, such as fire detectors or motion sensors for theft detection, to determine if a hazard is valid;
- monitoring health parameters using sensors for patients leaving a hospital; and sensors alerting people of danger.
- Another application is a text-based messaging application for real-time communication that capitalizes on MQTT's low data and energy usage.
- For example, Facebook uses MQTT for its Messenger app, because the protocol enables messages to be delivered efficiently in milliseconds, despite inconsistent internet connections across the globe.

- Most major cloud services providers, including Amazon Web Services (AWS), Google Cloud, IBM Cloud and Microsoft Azure, support MQTT.
- MQTT is well suited to applications using M2M and IoT devices for purposes like - real-time analytics, preventative maintenance and monitoring in environments, including smart homes, healthcare, logistics, industry and manufacturing.

How is MQTT used in IoT?

- MQTT clients require minimal resources and can be used on small microcontrollers, according to MQTT.org.
- To optimize network bandwidth, MQTT headers are small.
- Plus, MQTT "can scale to connect with millions of IoT devices," according to the organization.
- As a result, MQTT is one of the most commonly used protocols in IoT and IIoT infrastructure -- for example, for utilities industries to efficiently transmit data between their services and their customers and devices.
- Examples of MQTT use in IoT or IIoT in structure include the following:

Smart metering

- The MQTT protocol can be used to transmit data to provide accurate meter readings in real time.
- This helps make billing more accurate.

Gathering ambient sensor data

- Sensors used in remote environments are low-power devices,
- so MQTT is good for IoT sensor buildouts with lower-priority data transmission needs.

Machine health data

- provides a pub/sub messaging platform, giving the example of a wind turbine requiring "guaranteed delivery of machine health data to local teams even before that information hits a data center."

Billing systems

- MQTT helps eliminate duplicate or lost message packets in billing or invoicing.

There are three levels of quality of service in MQTT

- **QoS Level - 0 (non-assured transmission)**
  - This is also called called "fire and forget" or "at most once"
  - This is the minimal QoS level.
  - The recipient does not acknowledge receipt of the message to the sender
  - and the message is not stored and re-transmitted by the sender.
- **QoS Level - 1 (assured transmission)**
  - This is also called "at most once"
  - This level guarantees that a message is delivered at least one time to the receiver.

- The sender stores the message until it gets a PUBACK packet from the receiver, as an acknowledgement of the message.
- It is possible for a message to be sent or delivered multiple times.

- **QoS Level - 2 (assured service on applications)**
  - QoS 2 is the safest and slowest quality of service level
  - It ensures and informs both the sender and receiver that a message has been transmitted correctly.
  - The guarantee is provided by two request/response cycles (a four-part handshake) between the sender and the receiver.
  - The sender and receiver use the packet identifier of the original PUBLISH message to coordinate delivery of the message.
  - If a receiver gets a message set to QoS-2, it will respond with a PUBREC message to the sender.
  - the sender will respond with a PUBREL message.
  - The PUBREL is then acknowledged by the receiver with a PUBCOMP.
  - Until the PUBCOMP message is sent, the receiver will cache the original message for safety.

(1D4)**Fig. 4.6.3 : QoS Level 0, QoS Level 1 and QoS Level 2**

4.6.1.4 Advantages and Disadvantages of MQTT

Advantages

1. Low energy consumption
2. Reduced network bandwidth consumption
3. Highly scalable
4. Used and well suited for - remote sensing and control
5. Require lightweight resources to build an application stack
6. The message header can be only 2 bytes, it is beneficial for limited networks

7. Supports all messages on the network (like - Publish / Subscribe and Request / Reply)
8. Distribute information more efficiently
9. Used by oil-gas industry, Amazon, Facebook and other businesses
10. Saves development time

Disadvantages

1. Only the most recent message is stored in the broker, no message queue is maintained.
2. Header fields (such as TTL (Time To Leave)) sent to the user, are not supported

4.6.2 MQTT-SN (MQTT for Sensor Networks)

- The MQTT-SN is a broker based protocol.
- Many new features are added like - error status, concise message header etc.
- A derivative of MQTT is called MQTT-SN (sometimes called MQTT-S) for sensor networks.
- It is also a lightweight protocol for edge devices but is architected specifically for the nuances of a wireless PAN.
- MQTT-SN is so light that it can be run successfully over BLE and Zigbee.
- It does not require a TCP/IP stack.
- Alternatively, it can be used over UDP, which is less hungry than TCP.

4.6.2.1 MQTT-SN Architecture and Topology

There are four fundamental components in an MQTT-SN topology:

- **Gateways :** Gateway converts MQTT-SN to MQTT and vice versa
- **Forwarders**
  - A route between a sensor and an MQTT-SN gateway may take many paths and hop across several routers along the way.
  - Nodes between the source client and the MQTT-SN gateway are called forwarders
  - and simply re encapsulate MQTT-SN frames into new and unchanged MQTT-SN frames that are sent to the destination
  - until they arrive at the correct MQTT-SN gateway for protocol conversion.
- **Clients :** Clients of MQTT-SN behave in the same way as in MQTT, and are capable of subscribing and publishing data.
- **Brokers :** Brokers of MQTT-SN, behave in the same way as in MQTT.

(1D5)**Fig. 4.6.4 : Architecture of MQTT-SN**

- In Fig. 4.6.4, Wireless sensors (or MQTT-SN Clients) communicate with MQTT-SN Gateways and MQTT-SN Forwarders.
- MQTT-SN Gateways translates MQTT-SN to MQTT and communicates with Broker
- Forwarders that simply encapsulate MQTT-SN frames received into MQTT-SN messages and forwards to Gateways.

In MQTT-SN, the gateways are of two types

- **Transparent gateways** - manages many independent MQTT-SN streams from sensor devices and converts each stream into an MQTT message.
- **Aggregating gateways** - Coolects the number of MQTT-SN streams into a reduced number of MQTT streams sent to the cloud or MQTT broker. It is complex in design but reduces the amount of communication overhead and number of simultaneous connections left open on the server.

Fig. 4.6.5 : Types of Gateways in MQTT-SN (a) Transparent Gateway (b) Aggregating Gateway

4.6.2.2 Differences between MQTT and MQTT-SN

Sr. No.	Parameter	MQTT	MQTT-SN
1	Popularity	more	less
2	CONNECT messages	1	3
3	Topic Names	Long	Short
4	Predefined Topic Ids	No	Yes
5	Communication	TCP	UDP
6	Network	Ethernet, Wi Fi, 3G	Zigbee, Bluetooth, RF, etc.
7	Minimum Message Range	2 Bytes	1 Byte
8	Maximum Message Range	<=24 MB	< 128 Bytes
9	Battery	No	Yes
10	Sleep Mode Client	No	Yes
11	Connectionless Mode	No	Yes

4.6.3 Constrained Application Protocol (CoAP)

- CoAP replaces heavy HTTP abilities and usage, with a lightweight equivalent for IoT.
- The IETF Constrained RESTful Environments (CoRE) working group created the first draft of this protocol in June 2014, as as RFC7228. It is a web transfer (or Communication) protocol used for constrained nodes and networks. It is designed for Machine to Machine (M2M) communication between edge nodes. It is based on Request-Response model between end-points

(1D6)Fig. 4.6.6 : Architecture of CoAP Protocol

- CoAP works on minimum resources and low bandwidth.
- Representational State Transfer (REST) is the standard interface between HTTP client and servers.

- CoAP is designed to enable low-power sensors to use RESTful services while meeting their power constraints. This protocol is built over UDP, so it provides asynchronous interaction and it is less reliable.

4.6.3.1 Layers of CoAP Protocol

- CoAP architecture is divided into two sub-layers: **Messaging, Request/response.**
- **The messaging sub-layer** is responsible for reliability and duplication of messages,
- while the **Request/response** sub-layer is responsible for communication.

CoAP has two basic layers

1. **Request/response layer**
   - Responsible for sending and receiving RESTful-based queries.
   - REST queries are piggybacked on CON or NON messages.
   - A REST response is piggybacked on the corresponding ACK message.
2. **Transactional Layer (Messaging Layer)**
   - Handles single message exchanges between endpoints.
   - supports multicasting and congestion control

Fig. 4.6.7 : Layers of CoAP Protocol

4.6.3.2 CoAP has four messaging modes:

CoAP has four messaging modes: Confirmable, Non-confirmable, Piggyback, Separate

1. **Confirmable Message Mode**

   Represents the reliable transmission of the messages, between the client and the server.

Fig. 4.6.8 : Confirmable Message Mode

2. Non-confirmable Message Mode

Represents unreliable transmission of the messages, between the client and the server.

Fig. 4.6.9 : Non - Confirmable Message Mode

3. Piggyback Message Mode -

Used for the direct communication between the client and the server.

The server sends its response after receiving the message, within the acknowledgment message

Fig. 4.6.10 : Piggyback Message Mode

4. Separate Message Mode

This mode is used when the server sends the response message and the acknowledgment message separately.

Fig. 4.6.11 : Separate Message Mode

4.6.3.3 CoAP Message Format

- Fig. 4.6.12 shows the CoAP message format.
- The CoAP message is made by several parts like - ver, T, TKL, Code, Message ID, Token, Options and Payload.

Ver	T	TKL	Code	Message ID
Token				
Options (if exists...)				
Payload (if exists...)				

Fig. 4.6.12 : CoAP Message Format

Where,

Ver: It is a 2 bit unsigned integer indicating the version

T: it is a 2 bit unsigned integer indicating the message type: 0 confirmable, 1 non-confirmable

TKL: Token Length is the token 4 bit length

Code: It is the code response (8 bit length)

Message ID: It is the message ID expressed with 16 bit

4.6.3.4 CoAP Security Aspects

- Fig. 4.6.13 shows security aspects of CoAP protocol.
- CoAP uses UDP to transport information.
- CoAP relies on UDP security aspects to protect the information.
- As HTTP uses TLS over TCP, CoAP uses *Datagram TLS over UDP*.
- DTLS supports RSA, AES and so on.
- in some constrained devices some of DTLS cipher suites may not be available.

- Some cipher suites introduce some complexity and constrained devices may not have resources enough to manage it.

Fig. 4.6.13 : CoAP Security Aspects

4.6.3.5 Advantages of COAP

1. **Reduced power requirements** - this protocol operated on UDP, which is a connectionless protocol. So communication overhead is less. Due to this less power is required for this protocol.
2. **Smaller packet size** - UDP has small packet sizes, so, using COAP, faster communication cycles are possible between the IOT devices.
3. Designed to support IPV6

4.6.3.6 Disadvantages of COAP

1. **Unreliable message transfer** - as this protocol is used with UDP, there is no guarantee of the delivery of datagrams.
2. **Security** - COAP is an unencrypted protocol. This leads to unsecured communication.

4.6.3.7 CoAP vs MQTT

No	Parameter	CoAP	MQTT
1	Communication Model	Request-Response, Publish-Subscribe	Publish-Subscribe
2	RESTful	Yes	No
3	Transport Layer Protocol	Preferably UDP, TCP can be used	Preferably TCP, UDP can be used
4	Number of Message Types	4	16

No	Parameter	CoAP	MQTT
5	Messaging	Synchronous and Asynchronous	Asynchronous
6	Application Reliability	4 Levels	3 Levels
7	Security	IPSEC or DTLS	Not defined
8	Intermediaries	Yes	Yes (MQTT-S)
9	Power Consumption	Low	High
10	Header Size	4 bytes	2 Bytes
11	Architecture		

4.6.4 STOMP

- STOMP stands for Simple (or Streaming) Text Message-Oriented Middleware Protocol.
- STOMP specifies publish-subscribe mechanism
- It is a text-based protocol operating with message-oriented middleware.
- A broker developed in one programming language can receive messages from a client written in another.
- The protocol has similarities to HTTP and operates over TCP.
- STOMP consists of a frame header and a frame body.
- STOMP is optimized for human readability, fault-tolerant parsing, and self-described data.
- Clients subscribe to topics and they will be notified whenever messages have been sent to that particular topic by the broker.
- STOMP uses "SEND" semantics with the "destination" string.
- STOMP textual headers similar to HTTP.
- The broker must map "destination" string with something which understands
- Consumers subscribe to these destinations.

(107)**Fig. 4.6.14 : STOMP Architecture**

4.6.5 Advanced Message Queuing Protocol - AMQP

- AMQP is a flow-controlled, message-oriented communication system
- It is a wire-level protocol, above the physical layer and a low-level interface
- It processes billions of messages per day, and can collect many terabytes of data per day.
- It originated in 2003 at JP Morgan Chase, and later it was governed by the working group of 23 companies in 2006.
- In 2011, the working group was merged into the OASIS group.
- Today this protocol is used in banking and credit transaction industries and also in IoT.
- It is standardized by ISO and IEM as ISO/IEC 19464:2014. (www.amqp.org)
- The AMQP protocol is using the port 5672.
- Using AMQP, messages having a unique global id are broadcasted in unit frames.
- Each frame contains headers, footers, channel ids and payload information.
- AMQP decouples publishers from subscribers.
- Basic unit of data is Frame.
- Broker (or server) plays a crucial role in AMQP protocol
- It builds the connection for better data routing and queuing at the client-side.
- Consumer is responsible for - queues generation and message acknowledgement.
- Redirection of data is done through exchanges
- Producer places this data in queues.

(1D8)Fig. 4.6.15 : AMQP Protocol

4.6.5.1 Components of AMQP

- Exchanges (and their classification)
  - Fetches the messages and places them in the right queue.
  - Its 4 categories are: Fanout, Headers, Topic, and Direct.
- Channel
  - It is a multiplexed virtual connection among AMQP peers
  - which is built inside an existing connection.
- Message Queue
  - helps link messages with their resources or point of origin.
- Binding
  - denotes predefined instructions related to queues.
  - It administers the sending of messages and their delivery.
- Virtual Hosts
  - Offers the segregation facility inside the broker.

4.6.5.2 Advantages of AMQP

1. AMQP uses publish / subscribe architecture for data sharing
2. It ensures interoperability
3. Offers simpler peer to peer communication
4. This protocol can work with different standards
5. Offers secure connection for the users using SSL protocols like - COAP, MQTT, HTTP and XMPP.

4.6.5.3 Disadvantages of AMQP

1. This protocol is not backward compatible with old versions
2. It is very complex protocol than HTTP 1.0 or HTTP 1.1
3. It requires higher bandwidth than other protocols (like - COAP, MQTT, XMPP)

4.7 PROTOCOL SUMMARY AND COMPARISON

A summary and comparison of the various protocols is now given.

Sr. No.	Parameter	Protocols				
		MQTT	MQTT-SN	COAP	AMQP	STOMP
1	Header Size	2 Bytes	2 Bytes	4 Bytes	8 Bytes	8 Bytes
2	Broadcasting	Indirect	Indirect	Yes	No	No
3	QoS	Yes	Yes	CON Messages	Yes	No
4	TCP/UDP	TCP	TCP/UDP	UDP	TCP/UDP	TCP
5	Complexity	Low	Low	Low	High	Low
6	Access Control	No	No	No	Yes	No
7	Communication Overhead	Low	Very Low	Very Low	High	High
8	Resource required	Low	Very Low	Very Low	High	Medium
9	Power Usage	Lowest	Low	Medium	High	Medium

4.8 MULTIPLE CHOICE QUESTIONS

Q. 4.1 MQTT is based on which model?

(a) Publish-subscribe architecture

(b) Client-server Architecture

(c) Both A & B (d) None ✓**Ans. : (a)**

Q. 4.2 Which of the following is lightweight protocol?

(a) HTTP (b) MQTT (c) COAP (d) IP ✓**Ans. : (b)**

Q. 4.3 MQTT is mainly used for ?

(a) M2M Communication (b) Device Communication

(c) Internet Communication (d) Wireless Communication ✓**Ans. : (a)**

Q. 4.4 CoAP is specialized in _____

(a) Internet applications (b) Device applications

(c) Wireless applications (d) Wired applications

✓**Ans. : (a)**

Q. 4.5 CoAP is designed for use between devices on the same constrained network.

(a) True (b) False

✓**Ans. : (a)**

Q. 4.6 AMQP (Adv Message Queuing Protocol) is designed for connecting

(a) Constrained Networks (b) LANs and WANs

(c) Systems and Business Process (d) None of the above

✓**Ans. : (c)**

Q. 4.7 STOMP is a simple text-oriented messaging protocol used by browser to connect to brokers.

(a) True (b) False

✓**Ans. : (a)**

Q. 4.8 TCP supports reliable communication

(a) True (b) False

✓**Ans. : (a)**

Q. 4.9 UDP supports reliable communication

(a) True (b) False

✓**Ans. : (b)**

Q. 4.10 AMQP protocol requires more power than other protocols

(a) True (b) False

✓**Ans. : (a)**

Chapter Ends...

❑❑❑

MODULE 5

IoT and Data Analytics

Syllabus

Defining IoT Analytics, IoT Analytics challenges, IoT analytics for the cloud, Strategies to organize Data for IoT Analytics, Linked Analytics Data Sets, Managing Data lakes, The data retention strategy, visualization and Dash boarding – Designing visual analysis for IoT data, creating a dashboard, creating and visualizing alerts.

Self-learning Topics : AWS and Hadoop Technology.

5.1 Introduction to IoT Analytics 5-3
- 5.1.1 IoT Analytics different from Traditional Analytics 5-3
- 5.1.2 Devices that Power IoT Analytics 5-3
- 5.1.3 Process Flow in IoT Analytics 5-3
- 5.1.4 Types of IoT Analytics 5-4

5.2 Defining IoT Analytics 5-5

5.3 IoT analytics challenges 5-6
- 5.3.1 The Data Volume 5-7
- 5.3.2 Problems with Time 5-8
- 5.3.3 Problems with Space 5-10
- 5.3.4 Data Quality 5-11
- 5.3.5 Analytics Challenges 5-12
- 5.3.6 Business Value Concerns 5-13

5.4 IoT Analytics for the cloud 5-13
- 5.4.1 Cloud Infrastructure 5-14
- 5.4.2 Types of Cloud 5-14
- 5.4.3 Sectors that use Community Clouds are 5-15

5.5 Cloud Computing 5-17
- 5.5.1 Types of Cloud Computing 5-18
- 5.5.2 Advantages of Using Cloud 5-18

5.6 Elastic Analytics Concepts 5-21

5.7 Distributed Computing 5-22

5.8 Cloud Security and Analytics 5-23
- 5.8.1 Importance of Cloud Security 5-26
- 5.8.2 Security Benefits of Cloud Computing 5-27
- 5.8.3 Public/Private Keys 5-27
- 5.8.4 Public Versus Private Subnets 5-28
- 5.8.5 Access Restrictions 5-29
- 5.8.6 Securing Customer Data 5-29

5.9 The AWS Overview 5-30
- 5.9.1 AWS IoT Services 5-30

5.9.2 Services Included in IoT AWS 5-31
5.9.3 Benefits of the Amazon IoT Platform 5-32
5.9.4 AWS key Services for IoT Analytics 5-32
5.10 Microsoft Azure overview 5-34
5.10.1 Microsoft Azure 5-34
5.10.2 Various Azure Services 5-35
5.10.3 Services of Interest for IoT Analytics 5-36
5.11 The ThingWorx overview 5-38
5.11.1 ThingWorx Concepts 5-40
5.11.2 Strategies to Organize Data for Analytics 5-42
5.12 Managing data lakes 5-46
5.12.1 Data Refineries 5-47
5.12.2 Developing a Progression Process 5-48
5.13 The data retention strategy 5-49
5.13.1 Goals 5-49
5.13.2 Retention Strategies for IoT Data 5-49
5.13.3 The Retention Strategy Example 5-51
5.14 Common mistakes when designing visuals 5-52
5.15 The Hierarchy of Questions method 5-54
5.15.1 The Hierarchy of Questions Method Overview 5-55
5.15.2 Developing Question Trees 5-55
5.15.3 Pulling together the Data 5-58
5.15.4 Aligning views with Question Flows 5-58
5.16 Designing visual analysis for IoT data 5-58
5.16.1 Using Layout Positioning to Convey Importance 5-59
5.16.2 Use color to Highlight Important Data 5-59
5.16.3 Be Consistent Across Visuals 5-60
5.16.4 Make Charts Easy to Interpret 5-60
5.17 Creating a dashboard with Tableau 5-61
5.17.1 The dashboard walk-through 5-62
5.17.2 Hierarchy of Questions Example 5-62
5.17.3 Aligning Visuals to the thought Process 5-63
5.17.4 Creating Individual Views 5-63
5.17.5 Assembling views into a Dashboard 5-66
5.18 Creating and visualizing alerts 5-68
5.18.1 Alert Principles 5-68
5.18.2 Organizing alerts using a Tableau Dashboard 5-69
5.18.3 Types of Dashboards 5-71
5.18.4 Use Cases of Dashboards 5-72
5.18.5 Summary to Create a Data Dashboard 5-72
5.19 Example : Tableus DashBoard 5-72
5.19.1 Connection to Data Source 5-73
5.19.2 Creating Data Visualization Sheets 5-75
5.19.3 Building a Dashboard 5-79
5.19.4 Adding Filters 5-80
5.19.5 Adding Objects 5-81
5.19.6 Connecting Filters 5-83
5.19.7 Publishing 5-85
- **Chapter End**... 5-85

5.1 INTRODUCTION TO IOT ANALYTICS

- **Internet of Things (IoT) analytics** is a data analysis tool that assesses the wide range of data collected from IoT devices. IoT analytics assesses vast quantities of data and produces useful information from it.
- IoT analytics are usually discussed in tandem with Industrial IoT (IIoT). Data is collected from manufacturing infrastructure, meteorological stations, smart meters, delivery vans, and various sensors on all types of machines.
- IoT analytics can also be applied to retail and healthcare sectors. Data can be in different formats such as video feeds, geolocation data, social media data, or log files. Given the different types of information sources, data integration can be very difficult. This is exactly where IoT analytics makes a difference.
- In many ways, IoT data is similar to big data. The key difference between the two is not just the quantity of data but the range of sources it is obtained from. All this data has to be processed into one comprehensible, single stream of data.
- Considering the several kinds of sources of information, data integration becomes quite difficult, and this is where IoT analytics makes a difference, though it can be tough to develop and implement.

5.1.1 IoT Analytics different from Traditional Analytics

- Traditional analytics is done on structured data. IoT devices generate **unstructured data**. Data have different formats and are not standardized. Because sensors are affected by physical process and randomness, data may contain missing points, corrupted messages, and incorrect readings. This is unlike data entered by humans or scanned from forms.
- Traditional data systems don't change that often. For example, when filling a form many fields have a predefined range. IoT systems are more **dynamic**. Device IP addresses may change. Devices may be upgraded. New devices may be installed with better capability. Environmental conditions may affect sensor accuracy or precision.
- IoT data often comes in **real-time**. Many applications also require real-time analysis and insights. In fact, it's been said that much of IoT data becomes stale and useless if not analyzed immediately. Deciding whether to give someone a loan based on analysis of past behaviour is non-real-time traditional analysis. But to know available parking spaces requires IoT analytics in real-time.

5.1.2 Devices that Power IoT Analytics

There are a wide range of IoT devices that collect data:

(a) Wearables	(b) Smart Home	(c) Healthcare	(d) Voice-Activated Everything

(a) Wearables

- Dedicated trackers such as Fitbit or other smartwatches have gone beyond tracking steps. You can track your friends' fitness activities, compete with them, message, and even answer the phone by connecting your devices through the Internet.
- This information is tracked by fitness companies, enabling them to create customized packages if you sign up. This can include exercise routines, diet, goals, and more.
- The newest smart watches even monitor heart rates and rhythms and have accurately diagnosed heart problems in their wearers.

(b) Smart Home

- Smart homes have security systems you can access and control when you are away from home, to appliances you can turn on and off with digital assistance.
- There is a wide range of devices that you can incorporate into your home and a wide range of data that can be collected to assess usage patterns, the efficacy of systems, and more.

(c) Healthcare

- Healthcare has a wide range of IoT devices. Bluetooth technology creates hearing aids, records heart and blood pressure, and monitors pulse-based alarm systems that can call for help.
- This has helped enhance healthcare to a large extent.
- The data collected is invaluable in terms of creating newer and better technology.

(d) Voice-Activated Everything

- Digital assistants are a form of IoT devices. Alexa, Siri, and Google take notes, find information, play music, order cabs, tell the weather, set alarms, and everything else.
- The internet regularly updates these digital assistants to improve functionality.
- Their data helps companies tailor their services for you, based on your everyday interaction with digital assistants.

5.1.3 Process Flow in IoT Analytics

IoT analytics commonly adopts the following steps :

1. Data Collection

This step primarily comprises of data collected from various IoT sources including audio, image and light sensors. This heterogeneous nature of data raises the significance of IoT analytics technology.

2. Data Pre-processing

This step handles missing data, imports required libraries, encodes categorical data and does feature scaling.

3. **Analysis of Data**

This is mostly about exploratory data analysis that brings out summary statistics. It may suggest possible hypotheses and modelling approaches.

4. **Train and Test of data**

Data scientists build machine learning and deep learning models to suit business requirements. Models are trained from available data. With cross validation and online testing, the efficiency of the model is evaluated.

5. **Deployment and Improvement**

The tested model is deployed to deal with various real-world business problems.

5.1.4 Types of IoT Analytics

(a) Descriptive analytics on IoT data	(b) Diagnostic analytics on IoT data
(c) Predictive analytics on IoT data	(d) Prescriptive analytics on IoT data

(a) **Descriptive analytics on IoT data**

- Focuses on what's happening, by monitoring the status of IoT devices, machines, products and assets.
- Determines if things are going as planned, and notifies if anomalies occur.
- Descriptive analytics is generally implemented as dashboards that show current and historical sensor data, key performance indicators (KPIs), statistics and alerts.
- Addresses questions such as:
  - Are there any anomalies that demand attention?
  - What's the utilization and throughput of this machine?
  - How are consumers using our products?
  - Where do my assets reside?
  - How many components are we creating with this tool?
  - How much energy is this machine using?

(b) **Diagnostic analytics on IoT data**

- Answers the question: why is something happening? Analyzes IoT data to identify core problems and to fix or improve a service, product or process.
- Diagnostic capabilities are typically extensions to dashboards that permit users to drill into data, compare it, and visualize correlations and trends in an ad-hoc manner.
- Many organizations employ domain experts knowledgeable about a specific process, machine, device or product, rather than data scientists, to perform diagnostics on data.

- Addresses questions such as :
  - Why is this machine producing more defective parts than other machines?
  - Why is this machine consuming excessive energy?
  - Why aren't we producing enough parts with this tool?
  - Why are we getting a lot of product returns from American customers?

(c) Predictive analytics on IoT data

- Raises the question: what will happen? Assesses the likelihood that something will happen within a specific timeframe, according to historical data.
- The aim is to proactively take corrective action before an undesired outcome occurs, to mitigate risk, or to isolate opportunities.
- Typically implemented via machine learning models that are trained with historical data, and stationed on the cloud so that they can be accessed by end-user applications.
- Addresses questions such as :
  - What's the likelihood of this machine failing in the next 24 hours?
  - What is the anticipated useful life of this tool?
  - When should I service this machine?
  - What will be the demand for this feature or product?

(d) Prescriptive analytics on IoT data

- Poses the question: what action should I take? Suggests actions based on the result of a prediction or diagnosis, or provides some visibility to the rationale behind a prediction or diagnostic. Recommendations tend to be about how to optimize or fix something.
- Addresses questions such as :
  - This machine is 80 percent likely to fail in the next 12 hours. How should I prevent this?
  - The overall equipment effectiveness (OEE) of this machine is low. How can I improve it?
  - This machine is creating too many defective components. How can I avoid this?
  - This design is resulting in too many manufacturing issues. How can I improve it?

5.2 DEFINING IOT ANALYTICS

- In their best selling book *Competing on Analytics*, Tom Davenport and Jeanne Harris created a scale, which they called **Analytics Maturity**.
- Companies progress to higher levels in the scale as their use of analytics matures, and they begin to compete with other companies by leveraging it

- **The word analytics mean using techniques that fall in the range from query/drill down to optimization as shown in the following chart from Competing on Analytics :**

Fig. 5.2.1 : Analytical Maturity (Source : Competing on Analytics by Thomas H. Devenport & Jeanee G. Harris

- The notion of a company progressing through each level to get to the peak of maturity at the upper right withoptimization, Company reached to success at all levels in parallel.
- Company not being analytically mature till it is actively employing optimization models at every turn can be dangerous. A company is under pressure to focus time and resources where there may not be a return on investment (ROI) for them. Since resources are always limited, this could also cause them to under-invest in projects inother areas that have a higher ROI.
- The reason for the lack of ROI is often that a company simply does not have the right data to take full advantage of the more advanced techniques. This could be no fault of their own as the signal in the noise may be just too weak to tease out. This could stem from the state oftechnology, not yet at the point where the key predictive data can even be monitored. Or even if this is possible, it may be far too expensive to justify capturing it. The goal will always be to maximizeROI at all levels of the maturity model.

5.3 IOT ANALYTICS CHALLENGES

- The Internet of Things is an evolutionary step in internet-based computing. It already made a tremendous impact in a large number of application domains such as smart cities, sustainable living, manufacturing, and healthcare.
- IoT analytics is the analysis of data from disparate data sources that include sensors, actuators, and other objects connected to the internet. It is the key element of IoT's disruptive power.

- However, a McKinsey survey states that less than 1% of IoT data is used to make business decisions. This is a serious setback to maximize IoT business value. Today, most IoT applications are used for anomaly detection rather than optimization and prediction.
- There are some special challenges that come along with IoT data.
  - The data was created by devices operating remotely, sometimes in widely varying environmental conditions that can change from day to day.
  - The devices are often distributed widely geographically.
  - The data is communicated over long distances, often across different networking technologies.
  - It is very common for data to first transmit across a wireless network, then through a type of gateway device to be sent over the public internet–which itself includes multiple different types of networking technology working together.

5.3.1 The Data Volume

- A company can easily have thousands to millions of IoT devices with several sensors on each unit, each sensor reporting values on a regular basis.
- The inflow of data can grow quitelarge very quickly. Since IoT devices send data on an ongoing basis, the volume of data in total can increase much faster than many companies are used to.
- Consider a company that manufactures small monitoring devices. It produces 12,000 devices a year, starting in 2010 when the productwas launched.
- Each one is tested at the end of assembly and the values reported by the sensors on the device are kept for analysis for five years. The data growth looks like the following image :

Fig. 5.3.1 : A chart showing data storage needs for production snapshot of 200 KB and 1,000 units per month. Five years of production data is kept

- Also this device had internet connectivity to track sensor values, and each one remains connected for two years. Since the data inflow continues well after the devices are built, data growth is exponential until it stabilizes when older devices stop reporting values.
- This looks more like the shaded (IoT Data) area in the following chart :

Fig. 5.3.2 : Chart shows the addition of IoT data at 0.5 KB per message, 10 messages per day. Devices are connected for two years from production

How Large data is Generated

- Consider the following example. If one capture10 messages per day and the message size is half of a full production snapshot, by 2017, data storage requirements would be over 1,500 times higher than production-only data.
- For many companies, this introduces some problems. The database software, storage infrastructure, and available computing horsepower is not typically intended to handle thiskind of growth.
- The licensing agreements with software vendors tends to be tied to the number of servers and CPU cores. Storage is handled by standard backup planning and retention policies.
- The data volume rapidly leads to computing and storage requirements well beyond what can be held by a single server. It gets cost prohibitive very quickly under traditional architectures to distribute it across hundreds or thousands of servers.
- To do the best analytics, one need lots of historical data, and since you are unlikely to know ahead of timewhich data is most predictive, you have to keep as much as you can on hand.
- Analytic needs are very elastic. Traditional server planning ratchets up on premise resources with the anticipated number of servers needed to meet peak needs determined in advance.
- Doubling compute power in a short amount of time, if even possible, is very expensive.

- IoT data volumes and computing resource requirements can quickly outpace all the othercompany data needs combined.

5.3.2 Problems with Time

- Time depends on geographical position and the date on the calendar.
- The international standard way of tracking a common time is using Coordinated Universal, those are
  - Time (UTC). UTC is geographically tied to 0 longitude, which passes through Greenwich, England, in the UK. Although it is tied to the location, it is actually not the same as Greenwich Mean Time (GMT).
  - GMT is a time zone, while UTC is a time standard. UTCdoes not observe Daylight Savings Time (DST):

Fig. 5.3.3 : Standard time zones of the World. Source: CIA Factbook

(a) Different Local Time

- Data used for analytics is recorded at different location may be Central control room, site etc. everything happens at the same place and time zone.
- IoT devices are spread out across theglobe. The event happens at the absolute same time but they may have different local time.
- This difference in time of recording affects the integrity of the resulting analytics.

(b) Clock Synchronization

- There can also be issues with clock synchronization. Devices set their internal clock to be in sync with the time standard being used.

- If it is local time, it could be using the wrong time zone due to a configuration error.
- It could also get out of sync due to a communication problem with the time standard source.

(c) **Day Light Saving**

- If local time is being used, daylight savings time can cause problems. Daylight savings time changes is different from country to country.
- In the United States, daylight savings time is changed at 02:00 local time in each time zone.
- In the European Union, it is coordinated so that all EU countries change at 01:00 GMT for all timezones at once.
- This keeps time zones always an hour apart at the expense of it changing at different local times for each time zone.
- In practice, though, the time available for analytics can be the time the event occurred, the time the IoT device sent the data, the time the data was received, or the time the data was added to your data.

Fig. 5.3.4 : In early 2008, Central Brazil was one, two, or three hours head of eastern U.S., depending on the date

5.3.3 Problems with Space

IoT devices are located in multiple geographic locations. Different areas of the world have different environmental conditions. Temperature variations can affect sensor accuracy.

There may be less accurate readings in some part were temperature is hot, also cold impacts your device.

(a) **Different Locations**

- Taking an example of diesel engines, elevation may effect reading, also if location and elevation is not taken into consideration, you may falsely conclude from IoT sensor readings of delivery trucks are poorly managing fuel economy compared to a different locations.

- Since mountain roads can burn up some fuel!

Fig. 5.3.5 : US elevation profile from LA to NYC. Source: reddit.com

(b) Remote Location

- Remote locations may have weaker network access.
- The higher data loss could cause data values for those locations to be underrepresented in the resulting analytics.

(c) Battery and Solar Powered

- Many IoT devices are solar powered. The available battery charge can affect the frequencyof data reporting.
- A device in different location in India, where it is often cloudy and rainy will be more impacted than the same device installed in location where it is mostly sunny.

(d) Political Consideration

- There are also political considerations related to the location of the IoT device. Privacy laws in India affect how the data from devices can be stored and what type of analytics is acceptable.
- You may be required to anonymize the data from certain countries, which can affect what you can do with analytics.

5.3.4 Data Quality

- Constrained devices means lossy networks. For analytics, it often results in either missing or inconsistent data. The missing data is often not random. It can be impacted by the location. Devices run on a software, called firmware, which may not be consistent across locations. This could mean differences in reporting frequency or formatting of values. It can result in lost or mangled data.

- Data messages from IoT devices often require the destination to know how to interpret the message being sent. Software bugs can lead to garbled messages and data records.
- Messages lost in translation or never sent due to dead batteries result in missing values. The conservation of power often means not all values available on the device are sent at the same time. The resulting datasets often have missing values, as the device sends some values consistently every time it reports and sends some other values less frequently.

5.3.5 Analytics Challenges

Analytics often requires deciding on whether to fill in or ignore the missing values. Either choice may lead to a dataset that is not a representative of reality.

(a) Inaccurate Data

- For example how this can affect results, consider the case of inaccurate political poll results in recent years. Many experts believe it is now in near crisis due to the shift of much of the world to mobile numbers as their only phone number. For pollsters, it is cheaper and easier to reach people on landline numbers. This can lead to the over representation of people with landlines. These people tend to be both older and wealthier than mobile-only respondents.
- The response rate has also dropped from near 80% in the 1970s to about 8% (if you are lucky) today. This makes it more difficult (and expensive) to obtain a representative sample leading to many embarrassingly wrong poll predictions.

(b) Environmental Data

- There can also be outside influences, such as environment conditions, that are not captured in the data. Winter storms can lead to power failures affecting devices that are able to report back data.
- You may end up drawing conclusions based on a non-representative sample of data without realizing it. This can affect the results of IoT analytics – and it will not be clear why.

(c) Historical Data

- Since connectivity is a new thing for many devices, there is also often a lack of historical data to base predictive models on. This can limit the type of analytics that can be done with the data.
- It can also lead to a recency bias in datasets, as newer products are over represented in the data simply because a higher percentage are now a part of the IoT.

5.3.6 Business Value Concerns

- Many companies are struggling to find value with IoT data. The costs to store, process, and analyze IoT data can grow quickly. With future financial returns uncertain, some companiesare questioning if it is worth the investment.

- According to McKinsey & Company, a consulting agency, most IoT data is not used. From their research, less than 1% of data generated by an oil platform was used for decision- making purposes. We can accept that 1% of the data has value, but which 1% is it? This can vary depending onthe question.
- The business value challenge is how to keep costs low while increasing the ability to createsuperior financial returns. Analytics is a great way to get there.

5.4 IOT ANALYTICS FOR THE CLOUD

- Cloud IoT Analytics is a fully-managed service that makes it easy to run and operationalize sophisticated analytics on massive volumes of IoT data without having to worry about the cost and complexity typically required to build an IoT analytics platform.
- It is the easiest way to run analytics on IoT data and get insights to make better and more accurate decisions for IoT applications and machine learning use cases.
- IoT data is highly unstructured which makes it difficult to analyze with traditional analytics and business intelligence tools that are designed to process structured data.
- IoT data comes from devices that often record fairly noisy processes (such as temperature, motion, or sound). The data from these devices can frequently have significant gaps, corrupted messages, and false readings that must be cleaned up before analysis can occur.
- Also, IoT data is often only meaningful in the context of additional, third-party data inputs. For example, to help farmers determine when to water their crops, vineyard irrigation systems often enrich moisture sensor data with rainfall data from the vineyard, allowing for more efficient water usage while maximizing harvest yield.
- Cloud IoT Analytics automates each of the difficult steps that are required to analyze data from IoT devices. Cloud IoT Analytics filters, transforms, and enriches IoT data before storing it in a time-series data store for analysis.
- You can setup the service to collect only the data you need from your devices, apply mathematical transforms to process the data, and enrich the data with device-specific metadata such as device type and location before storing the processed data.
- Then, you can analyze your data by running ad hoc or scheduled queries using the built-in SQL query engine, or perform more complex analytics and machine learning inference. Cloud IoT Analytics makes it easy to get started with machine learning by including pre-built models for common IoT use cases.

5.4.1 Cloud Infrastructure

The National Institute of Standards and Technology defines five essential characteristics :

1. **On-demand self-service :** You can provision things such as servers and storage as needed and without interacting with someone.

2. **Broad network access :** Your cloud resources are accessible over the internet (if enabled) by various methods, such as web browser or mobile phone.
3. **Resource pooling :** Cloud providers pool their servers and storage capacity acrossmany customers using a multi-tenant model. Resources, both physical and virtual, are dynamically assigned and reassigned as needed. The specific location of resources is unknown and generally unimportant
4. **Rapid elasticity :** Your resources can be elastically created and destroyed. This can happen automatically as needed to meet demand. You can scale out rapidly. You can also contract rapidly. The supply of resources is effectively unlimited from your viewpoint.
5. **Measured service :** The resource usage is monitored, controlled, and reported by the cloud provider. You have access to the same information, providing transparency to your utilization. Cloud systems continuously optimize resourcesautomatically.

5.4.2 Types of Cloud

(a) Public cloud	(b) Private cloud
(c) Hybrid cloud	(d) Community cloud

(a) Public Cloud

- Public clouds are managed by third parties which provide cloud services over the internet to the public, these services are available as pay-as-you-go billing models.
- They offer solutions for minimizing IT infrastructure costs and become a good option for handling peak loads on the local infrastructure. Public clouds are the go-to option for small enterprises, which are able to start their businesses without large upfront investments by completely relying on public infrastructure for their IT needs.
- The fundamental characteristics of public clouds are multitenancy. A public cloud is meant to serve multiple users, not a single customer. A user requires a virtual computing environment that is separated, and most likely isolated, from other users.

Fig. 5.4.1 : Public cloud

(b) Private cloud

- Private clouds are distributed systems that work on private infrastructure and provide the users with dynamic provisioning of computing resources.
- Instead of a pay-as-you-go model in private clouds, there could be other schemes that manage the usage of the cloud and proportionally billing of the different departments or sections of an enterprise.

Fig. 5.4.2 : Private Cloud

Advantages of using a private cloud are

1. **Customer information protection :**

   In the private cloud security concerns are less since customer data and other sensitive information do not flow out of private infrastructure.

2. **Infrastructure ensuring SLAs :**

   Private cloud provides specific operations such as appropriate clustering, data replication, system monitoring, and maintenance, and disaster recovery, and other uptime services.

3. **Compliance with standard procedures and operations :**

   Specific procedures have to be put in place when deploying and executing applications according to third-party compliance standards. This is not possible in the case of the public cloud.

(c) Hybrid cloud

- A hybrid cloud is a heterogeneous distributed system formed by combining facilities of public cloud and private cloud. For this reason, they are also called heterogeneous clouds.
- A major drawback of private deployments is the inability to scale on-demand and efficiently address peak loads. Here public clouds are needed.
- Hence, a hybrid cloud takes advantage of both public and private clouds.

Fig. 5.4.3 : Hybrid Cloud

(d) Community cloud

- Community clouds are distributed systems created by integrating the services of different clouds to address the specific needs of an industry, a community, or a business sector.
- In the community cloud, the infrastructure is shared between organizations that have shared concerns or tasks. The cloud may be managed by an organization or a third party.

Fig. 5.4.4 : Community Cloud

5.4.3 Sectors that use Community Clouds are

1. **Media industry :** Media companies are looking for quick, simple, low-cost ways for increasing the efficiency of content generation.

   Most media productions involve an extended ecosystem of partners. In particular, the creation of digital content is the outcome of a collaborative process that includes the movement of large data, massive compute-intensive rendering tasks, and complex workflow executions.

2. **Healthcare industry :** In the healthcare industry community clouds are used to share information and knowledge on the global level with sensitive data in the private infrastructure.
3. **Energy and core industry :** In these sectors, the community cloud is used to cluster a set of solution which collectively addresses management, deployment, and orchestration of services and operations.
4. **Scientific research :** In this organization with common interests in science share a large distributed infrastructure for scientific computing.

5.5 CLOUD COMPUTING

Cloud Computing can be defined as the practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer. Companies offering such kinds of cloud computing services are called cloud providers and typically charge for cloud computing services based on usage. Grids and clusters are the foundations for cloud computing.

5.5.1 Types of Cloud Computing

- Most cloud computing services fall into five broad categories :

(a) Software as a service (SaaS)
(b) Platform as a service (PaaS)
(c) Infrastructure as a service (IaaS)
(d) Anything/Everything as a service (XaaS)
(e) Function as a Service (FaaS)

- These are sometimes called the **cloud computing stack** because they are built on top of one another. Knowing what they are and how they are different, makes it easier to accomplish your goals.
- These abstraction layers can also be viewed as a **layered architecture** where services of a higher layer can be composed of services of the underlying layer i.e, SaaS can provide Infrastructure.

(a) Software as a Service(SaaS)

- Software-as-a-Service (SaaS) is a way of delivering services and applications over the Internet. Instead of installing and maintaining software, we simply access it via the Internet, freeing ourselves from the complex software and hardware management.
- It removes the need to install and run applications on our own computers or in the data centers eliminating the expenses of hardware as well as software maintenance. SaaS provides a complete software solution that you purchase on a **pay-as-you-go** basis from a cloud service provider.

- Most SaaS applications can be run directly from a web browser without any downloads or installations required. The SaaS applications are sometimes called **Web-based software, on-demand software, or hosted software.**

Advantages of SaaS

1. **Cost-Effective :** Pay only for what you use.
2. **Reduced time :** Users can run most SaaS apps directly from their web browser without needing to download and install any software. This reduces the time spent in installation and configuration and can reduce the issues that can get in the way of the software deployment.
3. **Accessibility :** We can Access app data from anywhere.
4. **Automatic updates :** Rather than purchasing new software, customers rely on a SaaS provider to automatically perform the updates.
5. **Scalability :** It allows the users to access the services and features on-demand.

- The **various companies** providing Software as a service are Cloud9 Analytics, Salesforce.com, Cloud Switch, Microsoft Office 365, Big Commerce, Eloqua, dropBox, and Cloud Tran.

(b) Platform as a Service

- PaaS is a category of cloud computing that provides a platform and environment to allow developers to build applications and services over the internet. PaaS services are hosted in the cloud and accessed by users simply via their web browser.
- A PaaS provider hosts the hardware and software on its own infrastructure. As a result, PaaS frees users from having to install in-house hardware and software to develop or run a new application. Thus, the development and deployment of the application take place **independent of the hardware**.
- The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.
- To make it simple, take the example of an annual day function, you will have two options either to create a venue or to rent a venue but the function is the same.

Advantages of PaaS

1. **Simple and convenient for users :** It provides much of the infrastructure and other IT services, which users can access anywhere via a web browser.
2. **Cost-Effective :** It charges for the services provided on a per-use basis thus eliminating the expenses one may have for on-premises hardware and software.
3. **Efficiently managing the lifecycle :** It is designed to support the complete web application lifecycle: building, testing, deploying, managing, and updating.
4. **Efficiency :** It allows for higher-level programming with reduced complexity thus, the overall development of the application can be more effective.

- The various companies providing Platform as a service are Amazon Web services Elastic Beanstalk, Salesforce, Windows Azure, Google App Engine, cloud Bess and IBM smart cloud.

(c) Infrastructure as a Service

- Infrastructure as a service (IaaS) is a service model that delivers computer infrastructure on an outsourced basis to support various operations.
- Typically IaaS is a service where infrastructure is provided as outsourcing to enterprises such as networking equipment, devices, database, and web servers.
- It is also known as **Hardware as a Service (HaaS).** IaaS customers pay on a per-user basis, typically by the hour, week, or month. Some providers also charge customers based on the amount of virtual machine space they use.
- It simply provides the underlying operating systems, security, networking, and servers for developing such applications, and services, and deploying development tools, databases, etc.

Advantages of IaaS

1. **Cost-Effective :** Eliminates capital expense and reduces ongoing cost and IaaS customers pay on a per-user basis, typically by the hour, week, or month.
2. **Website hosting :** Running websites using IaaS can be less expensive than traditional web hosting.
3. **Security :** The IaaS Cloud Provider may provide better security than your existing software.
4. **Maintenance :** There is no need to manage the underlying data center or the introduction of new releases of the development or underlying software. This is all handled by the IaaS Cloud Provider.

- The various companies providing Infrastructure as a service are Amazon web services, Bluestack, IBM, Openstack, Rackspace, and Vmware.

(d) Anything as a Service

- It is also known as Everything as a Service.
- Most of the cloud service providers nowadays offer anything as a service that is a compilation of all of the above services including some additional services.

Advantages of XaaS

As this is a combined service, so it has all the advantages of every type of cloud service.

(e) Function as a Service

- FaaS is a type of cloud computing service. It provides a platform for its users or customers to develop, compute, run and deploy the code or entire application as functions. It allows the user to entirely develop the code and update it at any time without worrying about the maintenance of the underlying infrastructure.

- The developed code can be executed with response to the specific event. It is also **as same as PaaS**.
- FaaS is an event-driven execution model. It is implemented in the serverless container. When the application is developed completely, the user will now trigger the event to execute the code. Now, the triggered event makes response and activates the servers to execute it.
- The servers are nothing but the Linux servers or any other servers which is managed by the vendor completely. Customer does not have clue about any servers which is why they do not need to maintain the server hence it is **serverless architecture.**
- Both PaaS and FaaS are providing the same functionality but there is still some differentiation in terms of Scalability and Cost.
- FaaS, provides auto-scaling up and scaling down depending upon the demand. PaaS also provides scalability but here users have to configure the scaling parameter depending upon the demand.
- In FaaS, users only have to pay for the number of execution time happened. In PaaS, users have to pay for the amount based on pay-as-you-go price regardless of how much or less they use.

Advantages of FaaS

1. **Highly Scalable :** Auto scaling is done by the provider depending upon the demand.
2. **Cost-Effective :** Pay only for the number of events executed.
3. **Code Simplification :** FaaS allows the users to upload the entire application all at once. It allows you to write code for independent functions or similar to those functions.
4. Maintenance of code is enough and no need to worry about the servers.
5. Functions can be written in any programming language.
6. Less control over the system.

- The various companies providing Function as a Service are Amazon Web Services – Firecracker, Google – Kubernetes, Oracle – Fn, Apache OpenWhisk – IBM, OpenFaaS,

5.5.2 Advantages of Using Cloud

The advantages of using the cloud are :

1. **Speed :** You can bring cloud resources online in minutes.
2. **Agility :** The ability to quickly create and destroy resources leads to ease of experimentation. This increases the agility of analytics organizations.
3. **Variety of services :** Cloud providers have many services available to support analytics workflows that can be deployed within minutes. These services manage hardware and storage needs for you.
4. **Global reach :** You can extend the reach of analytics to the other side of the world with a few clicks.

5. **Cost control :** You only pay for the resources you need at the time you need them. You can do more for less.

5.6 ELASTIC ANALYTICS CONCEPTS

- Elastic Analytics means designing analytics processess so that scale is not a concern. Main focus should be on analytics not technology used.
- Analytics should be able to scale. It should go from supporting 100 IoTdevices to 1 million IoT devices without requiring any fundamental changes. All thatshould happen is that the costs increase as demand increases.
- This reduces complexity and increases maintainability. This translates into lower costs, which enables you to do more analytics. More analytics increases the probability of findingvalue. Finding more value enables even more analytics.

Core elastic analytics concepts

- Separate compute from storage
- Laptop or desktop are with fixed memory and storage.
- Cloud infrastructure abstracts this away. Doing analytics in the cloud is like renting a laptop or Desktop where you can change memory as per need and it is chargeable only for that much memory used.
- Same way hard drive can grow and shrink independently of the memory specification.
- Depending on the requirement one can choose a good balance between them and match compute needs with requirements.

Build for scale from the start

- Use software, services, and programming code that can scale from 1 to 1 million without changes.
- Each analytic process you putin production has continuing maintenance efforts that will build up over time as you add more and more processes.

Make your bottleneck wetware not hardware

- By wetware (human brain cells). *Not enough memory to run the job* will never be theproblem.
- Since requirement can be changed as per need in cloud computing.

Manage to a spend budget, not to available hardware

- Use as many cloud resources which fits within the budget. There is no need to limit analytics to fit within a set number of servers, analytics can be runed on the cloud.
- The traditional enterprise architecture purchases hardware ahead of time, which incurs a capital expense.

- Managing to spend means keeping track on costs, not on resource limitations. Expand when needed and make sure to contract quickly to keep costs down.

Experiment, experiment, and experiment:

- Create resources, try things out, and kill them off if they do not work. Then, try something else. Iterate till correct resources are available.
- Scale out resources to run experiments. Stretch it, cut down back down when done.
- If elastic analytics is done correctly, then biggest limitations are time andwetware can be overcome, not hardware and capital.

5.7 DISTRIBUTED COMPUTING

- Distributed computing (or distributed processing or cluster Computing) is the technique of linking together multiple computer servers over a network into a cluster, to share data and to coordinate processing power. Such a cluster is referred to as a "distributed system."
- Distributed computing is a computing concept that, in its most general sense, refers to multiple computer systems working on a single problem. In distributed computing, a single problem is divided into many parts, and each part is solved by different computers. As long as the computers are networked, they can communicate with each other to solve the problem. If done properly, the computers perform like a single entity
- Distributed computing offers advantages in scalability (through a "scale-out architecture"), performance (via parallelism), resilience (via redundancy), and cost-effectiveness (through the use of low-cost, commodity hardware).
- The ultimate goal of distributed computing is to maximize performance by connecting users and IT resources in a cost-effective, transparent and reliable manner. It also ensures fault tolerance and enables resource accessibility in the event that one of the components fails.

(a) **Avoid containing analytics to one server**

- The advantage to this for IoT analytics is in scale. You can add resources by adding nodes to the cluster; no change to the analytics code is required.
- The most common framework in use today is Hadoop. Try and avoid containing analytics to one server (with a few exceptions). This puts a ceiling on scale.

(b) **When to use distributed and when to use one server**

- There is a complexity cost to distributed computing though. It is not as simple as single server analytics.
- Even though the frameworks handle a lot of the complexity for you, youstill have to think and design your analytics to work across multiple nodes.

- Guidelines on when to keep it simple and on one server :
  - **There is not much need for scale :** Your analytics process needs little change even if the number of IoT devices and data explodes. For example, the analytics process runs a forecast on data already summarized by month. The volume of devices makes little difference in that case.
  - **Small data instead of big data :** The analytics run on a small subset of data without much impact from data size. Analytics on random samples is an example.
  - **Resource needs are minimal :** Even at orders of magnitude more data, you are unlikely to need more than what is available with a standard server. In this case,keep it simple.

(c) Assuming that change is constant

- The world of IoT analytics moves quickly. The analytics you create today will change many times over as you get feedback on results and adapt to the changing business conditions.
- Your analytics processes will need to change. Assume this will happen continuously and design for change. This brings us to the concept of **continuous delivery**.
- Continuous delivery is a concept from software development. It automates the release of code into production. The idea is to make change a regular process.
- Bring this concept intoyour analytics by keeping a set of simultaneous copies that you use to progress through three stages :
  1. **Development :** Keep a copy of your analytics for improving and trying out newthings.
  2. **Test :** When ready, merge your improvements into this copy where the functionality stays the same, but it is repeatedly tested. The testing ensures it isworking as intended. Keeping a separate copy for test allows development to continue on other functionality.
  3. **Master :** This is the copy that goes into production. When you merge things fromtest to the master copy, it is the same as putting it into live use. Cloud providers often have a continuous delivery service that can make this process simpler.
- For any software developer readers out there, this is a simplification of the **git flow** method, which is a little outside the scope of this book. If the author can drop a suggestion, it is worth some additional research to learn git flow and apply it to your analytics development in the cloud.

(d) Leverage managed services

- Cloud infrastructure providers, such as AWS and Microsoft Azure, offer services for things such as message queues, big data storage, and machine learning processing.
- The services handle the underlying resource needs such as server, storage provisioning, and also network requirements. User need not to bother how this happens under the hood, and it scales as big as per requirement.

- Cloud providers also manage global distribution of services to ensure low latency. The following image shows the AWS regional data center locations combined with under water internet cabling:

Fig. 5.7.1 : AWS regional data center locations and underwater internet cables

Source : http://turnkeylinux.github.io/aws-datacenters/

- This reduces the amount of things you have to worry about for analytics. It allows you to focus more on the business application and less on the technology.
- An example of a managed service is Amazon **Simple Queue Service (SQS)**. SQS is a message queue where the underlying server, storage, and compute needs are managed automatically by AWS systems. Only thing needed is to set it up and configure it, which takes justa few minutes.

(e) **Use Application Programming Interfaces (API)**

- API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you're using an API.
- APIs are a way for other processes, software, or services to access analytics code that you have created. It allows you to easily reuse your code in other applications. You can also allow your customers to directly access this functionality through web-based APIs.
- An API builds on of encapsulation principle. It is a defined list of supported actions and information that another system can call and retrieve.
- The calling system does not need to know how the action is performed or the information created and retrieved. The complexityis hidden.

- The API defines how to interact with an encapsulated set of analytic processes. It abstracts away the details. It also supports low friction change as the analytics processes can be improved without requiring other systems to alter what they are doing. As long as the API definition is held constant, the other systems will not know the difference.
- APIs are a great way to create building blocks for more complex analytics. You can use multiple APIs to build rich, fully-functional analytic applications in a short period of time. Itis also far easier to adapt to changing business conditions by reconfiguring the assembled applications to use a different mix of APIs.

Example Web API

- A Web API uses the internet to handle the communication between systems. Cloud providers offer this as a managed service. It can help a great deal in handling security andscale.
- Using this service, you can quickly implement new analytics functionality securely and at scale. The API can be either public or private. Private is only accessible to your internal applications. Public opens it up to systems outside your company.

Fig. 5.7.2 : Architecture using Web API gateway. Source: Amazon Web Services

- Cloud providers can even handle usage tracking and billing if you decide to offer the functionality behind your API as a paid service to your customers

5.8 CLOUD SECURITY AND ANALYTICS

Definition : Cloud Security is a branch of cybersecurity that is concerned with protecting the cloud from cyberattacks and any other form of threat. It protects the important data, applications and secure the cloud infrastructure to keep the cloud computing environment safe.

- Cloud Computing provides **services on demand**. It means storing the data over the Internet. As long as the business will have internet access it will be able to access the cloud storage and servers' resources.
- It provides resources on demand, from applications to storage to computing power even networking, and software availability, cloud has got your back. Variety of IT services are delivered via cloud.
- Cloud is helping businesses to increase productivity and cut down costs as services can be accessed with ease, most businesses are opting for Private and Hybrid Cloud Computing networks.
- Cloud Security keeps the cloud storage safe from data breaches, data loss and account hijacking. So for businesses that opt for cloud, cloud security can keep all the IT services away from data attacks.

5.8.1 Importance of Cloud Security

- Security in cloud computing is crucial to any company looking to keep its applications and data protected from bad actors. Maintaining a strong cloud security posture helps organizations achieve the now widely recognized benefits of cloud computing.
- Cloud security comes with its own advantages as well, helping you achieve lower upfront costs, reduced ongoing operational and administrative costs, easier scaling, increased reliability and availability, and improved DdoS protection.

5.8.2 Security Benefits of Cloud Computing

1. Lower upfront costs
2. Reduced ongoing operational and administrative expenses
3. Increased reliability and availability
4. Centralized security
5. Greater ease of scaling
6. Improved DdoS protection

1. Lower upfront costs

- One of the biggest advantages of using cloud computing is that you don't need to pay for dedicated hardware.
- Not having to invest in dedicated hardware helps you initially save a significant amount of money and can also help you upgrade your security.
- CSPs will handle your security needs proactively once you've hired them. This helps you save on costs and reduce the risks associated with having to hire an internal security team to safeguard dedicated hardware.

2. Reduced ongoing operational and administrative expenses

- Cloud security can also lower your ongoing administrative and operational expenses. A CSP will handle all your security needs for you, removing the need to pay for staff to provide manual security updates and configurations.
- You can also enjoy greater security, as the CSP will have expert staff able to handle any of your security issues for you.

3. Increased reliability and availability

- You need a secure way to immediately access your data. Cloud security ensures your data and applications are readily available to authorized users.
- You'll always have a reliable method to access your cloud applications and information, helping you quickly take action on any potential security issues.

4. Centralized security

- Cloud computing gives you a centralized location for data and applications, with many endpoints and devices requiring security.
- Security for cloud computing centrally manages all your applications, devices, and data to ensure everything is protected.
- The centralized location allows cloud security companies to more easily perform tasks, such as implementing disaster recovery plans, streamlining network event monitoring, and enhancing web filtering.

5. Greater ease of scaling

- Cloud computing allows you to scale with new demands, providing more applications and data storage whenever you need it.
- Cloud security easily scales with your cloud computing services. When your needs change, the centralized nature of cloud security allows you to easily integrate new applications and other features without sacrificing your data's safety.
- Cloud security can also scale during high traffic periods, providing more security when you upgrade your cloud solution and scaling down when traffic decreases.

6. Improved DdoS protection

- Distributed Denial of Service (DdoS) attacks are some of the biggest threats to cloud computing. These attacks aim a lot of traffic at servers at once to cause harm.
- Cloud security protects your servers from these attacks by monitoring and dispersing them.

5.8.3 Public/Private Keys

- Cloud providers use asymmetric cryptography throughout their services. The public and private keys are generated. Person keeps the private key, so the service does not have a copy. The service holds the public key. Communication using public/private key is secure and hasnever been broken.

- The cloud provider could publish the public key in tomorrow's newspaper and it would notmatter; the encryption cannot be broken with just the public key. It may seem counterintuitive that a public key is used to encrypt data but cannot be used to decrypt it.
- But it works.
- Every time you visit a website starting with HTTPS:, a public/private key encryption isbeing used. It is the basis of SSL and TLS encryption, which is employed for HTTPS communications.
- The public/private keys are often used for IoT analytics when you build secure processes. Think of it like a username and password for your analytics, but better.

5.8.4 Public Versus Private Subnets

- When you set up a cloud environment for analytics, Own networking environment can be created, also define the networkingstructure. A basic component of this is the concept of subnets.
- Subnets are logical subdivisions of the overall network in your cloud environment. You launch resources into a subnet where it will follow the internet addressing rules defining forthe subnet. Subnet can be
  - Public subnet and
  - Private subnet
- A **public subnet** has resources that can be addressable from the outside internet. This does not mean that all resources in the subnet can be found from the outside; It is needed to assign a public IP address to it first.
- A **private subnet** is not addressable from the outside internet. There are methods to allow internet communication through a gateway device, usually a **Network Address Translation(NAT)** device, but an outside object cannot initiate communication directly with something in a private subnet.
- Most analytic processing should happen in a private subnet for security reasons, which adds some complexity in connecting with resources.

5.8.5 Access Restrictions

- Restrict access to your analytic resources to named users only. Avoid the trap of keeping asingle user ID and password that everyone uses. For resources accessed by public/private keys, make sure to keep the private key in a safe, secure place.
- For networking security, only allow network traffic through to your resources that are needed. Block all others. Lock everything down from the start and then open things up onlywhen needed.
- In a distributed computing environment, such as used for IoT analytics, networkingis a key element in both security and problem solving. There may be situations where your analytic jobs will not run correctly and find that the source was a network or security related setting.

5.8.6 Securing Customer Data

- Keep your IoT data secure, use access controls, and encrypt data files. IoT data can be used to infer things about your customers. Unintentionally exposing company's to legal risk by not properly securing customer data.
- Encryption can be enabled in two ways,
  - In transit and
  - At rest.
- Encryption in **transit** refers to network transmission such as SSL or TLS. Most cloud services require their use. Make sure all thenetwork communication in support of your analytic processes is encrypted in transit.
- Encryption at rest **refers** to data storage. This could be data files on a server or data inside a database. Cloud providers can encrypt your data and make it transparent to you if you are using public/private key pairs to secure it.
- This is used when data could be accessed by a wide set of users, such as all employees of your company. If the data is only accessible by a few people due to tight access controls and it is in a private subnet, can be a little less
- Cloud providers has built in operating inside the public internet. Security is built in at everylevel. Take advantage of it and use public/private keys, private subnets, access control, andencryption to secure your valuable data

5.9 THE AWS OVERVIEW

- In 2006, Amazon Web Services (AWS) started to offer IT services to the market in the form of web services, which is nowadays known as cloud computing. With this cloud, It is not needed to plan for servers and other IT infrastructure which takes up much of time in advance.
- Instead, these services can instantly spin up hundreds or thousands of servers in minutes and deliver results faster. Its paid only for what used with no up-front expenses and no long-term commitments, which makes AWS cost efficient.
- Today, AWS provides a highly reliable, scalable, low-cost infrastructure platform in the cloud that powers multitude of businesses in 190 countries around the world.

5.9.1 AWS IoT Services

- AWS IoT Services is a cloud platform that works with thousands of connected devices and is capable of processing trillions of requests simultaneously.
- To store communication files and enable features, AWS IoT Services offers cloud infrastructure the information is stored on Amazon Web Services servers.

Solutions offered by the AWS IoT platform Offer for IoT Devices

- The Amazon Internet of Things platform essentially connects IoT devices to the Cloud. Each device transfers its information to the device shadow. The shadow service will respond to requests and work with the app's functionality.
- X.509 certificates project the communications between the actual device and its shadow service. This is a basic idea behind the AWS IoT platform let's take a closer look at its leading solutions.

1. **IoT Device Management :** A service that allows registering, organizing, securing, monitoring, and managing remotely connected devices and sensors. The service provides real-time statistics on the application's performance and allows uploading multiple devices in one go.
2. **IoT Device Defender :** Amazon IoT platform makes sure to collect security reports from all connected devices these metrics are sent to the Device defender, which audits whether metrics show abnormalities. If there's a strange access attempt or unnatural □ehaviour, Defender updates AWS Cloud Watch, IoT Console, and Device Management.
3. **AWS Lambda :** A software development environment where developers can write and edit code, aggregate projects from other AWS services (including IoT), and run code as soon as it's written. It's a platform for continuous deployment developers can release code to the service one at a time, avoiding tech debt and bug cluttering.
4. **AWS IoT Greengrass :** Physical devices that generate information (equipment, transport, etc.) are connected to the Greengrass Connectors. The data from connectors is sent to Lambda and device shadows by the Greengrass Core. This is how the data from the outside world makes its way to the software.
5. **AWS IoT Analytics :** The service creates analytics of IoT data. It's responsible for the collection, processing, storage, real-time machine learning analytics, and code-based reports.

5.9.2 Services Included in IoT AWS

- AWS IoT Services is an ambitious IoT management platform with dozens of features. If we were to focus on the complete functionality, you'd be reading a 40-page ebook right now.
- To keep the guide brief, we highlighted the vital AWS IoT services that are essential to most IoT projects.

1. **Device Gateway :** All devices in Amazon Web Services for IoT are connected to the Gateway. The service is responsible for maintaining connections between devices and a server even in low-latency conditions. The Device Gateway is the entrance to using the AWS IoT platform.
2. **Message Broker :** This service enables connected devices to exchange messages with each other and an application server. This tool is responsible for connectivity it can process, store, and organize thousands of messages simultaneously.

3. **Device Shadow :** All AWS IoT devices in the AWS IoT platform has a virtual version a shadow. It stores information on the state of physical equipment that you can access remotely. Here, you set the performance parameters for IoT devices, and even plan the settings a year in advance.
4. **Rules Engine :** This tool puts restrictions and imposes guidelines on data usage. The rule defines how devices process data. For instance, you can specify a threshold and set a standard for values that are above the limit. AWS IoT rules will trigger the execution of a certain functionality from AWS Lambda, connecting hardware updates with software reactions.

- The primary purpose of Amazon and IoT services is to connect IoT hardware and software code. The platform creates an environment for secure data exchange, organization, and management. Changes in the state of the device are saved to the system, where rules can trigger changes in code.

5.9.3 Benefits of the Amazon IoT Platform

The AWS IoT Platform makes IoT development faster: the code is available in AWS Lambda, where it can be executed right away, the hardware is managed in shadows, etc. This is not the only advantage a lot of benefits come into play after the product release.

1. **Device management :** Solutions responsible for device management, data organization, and cloud integration, make it easy for developers and project managers to track changes in the product.
2. **Data security and connection safety :** AWS has strict access control algorithms you can set up multi-step authentication and define user roles. The system continually monitors the software performance and detects suspicious patterns. You'll get an alert if something seems unusual.
3. **Improved data processing :** Amazon IoT platform uses AI to set up models of data storing and processing. You can set up scenarios that will be automatically executed in the cloud. Data processing can improve its efficiency and speed with Machine Learning add-ons.
4. **Scaling of IoT projects :** AWS IoT is connected to a productive Amazon Web Services infrastructure. The service acts together with robust services, which allows you to add new features. If you want to add machine learning to your IoT, use Amazon SageMaker. To increase data storage, you have Amazon S3. The fact that the Amazon IoT Platform is a part of a significant infrastructure is an advantage because you will not have to migrate to another service to add a new feature.

5.9.4 AWS key Services for IoT Analytics

The most important services for IoT Analytics are :

(a) Amazon Simple Queue Service (SQS)
(b) Amazon Elastic Map Reduce (EMR)
(c) AWS machine learning
(d) Amazon Relational Database Service (RDS)
(e) Amazon Redshift

(a) **Amazon Simple Queue Service (SQS)**

This is the AWS managed message queue service' is to be paid.

(b) **Amazon Elastic Map Reduce (EMR)**

- EMR is a fully managed Hadoop framework that can be launched in minutes. It handles the tasks of node provisioning, cluster setup, configuration, and cluster tuning for you. It operates using EC2 instances and can scale from one node to thousands.
- The number of instances can be increased or decreased manually or use auto scaling to do it dynamically, even while the cluster is running. The EMR service monitors cluster; it can handle retries for failed tasks and will replace poor performing instances automatically.
- Even though it is managed, you have complete control over the cluster, including root access. EMR has the option to choose from several Hadoop distributions and applications such as Apache Spark, Presto, and Hbase.
- Data storage can be linked to S3 using the **EMR File System (EMRFS)**. Data can be stored in Amazon S3 and use multiple EMR clusters to process the same dataset.

(c) **AWS machine learning**

- This service has wizards and visualization tools that guide you through the process of creating machine learning models.
- The trained can be deployed models to make predictions on new data without having to set up a separate system.
- Pay as you go, there are no upfront costs. It is highly scalable and can handle billions of predictions in a day.

(d) **Amazon Relational Database Service (RDS)**

- RDS is a set of managed relational databases and includes database engines such as Oracle, Microsoft SQL Server, PostgreSQL, MySQL, and MariaDB. There is full control over the database but do not have any access to the underlying server it is hosted on.
- RDS handles this part and also provides monitoring on the health of the database management system. You can connect to and manage databases on RDS using the same tools that you would use on a non-RDS Oracle or SQL Server database, for example.

(e) **Amazon Redshift**

- Redshift is a fully managed, SQL-compliant data warehouse service. Storage up to a petabyte in a Redshift cluster. It has its own JDBC and ODBC drivers but also supports standard PostgreSQL drivers.
- This means most Business Intelligence (BI) tools can connect directly to it. Redshift is useful for commonly queried data and is a common place to put processed and summarized IoT data for wider enterprise consumption.

5.10 MICROSOFT AZURE OVERVIEW

- Today, cloud computing applications and platforms are rapidly growing across all industries, serving as the IT infrastructure that drives new digital businesses.
- These platforms and applications have revolutionized the ways in which businesses function, and have made processes easier.
- In fact, more than 77 percent of businesses today have at least some portion of their computing infrastructure in the cloud.
- While there are many cloud computing platforms available, two platforms dominate the cloud computing industry. Amazon Web Services (AWS) and Microsoft Azure are the two giants in the world of cloud computing.
- While AWS is the largest cloud computing platform, Microsoft Azure is the fastest-growing and second-largest.

5.10.1 Microsoft Azure

- Azure is a cloud computing platform and an online portal that allows you to access and manage cloud services and resources provided by Microsoft.
- These services and resources include storing your data and transforming it, depending on your requirements.
- To get access to these resources and services, all you need to have is an active internet connection and the ability to connect to the Azure portal.
- Things that you should know about Azure:
    - It was launched on February 1, 2010, significantly later than its main competitor, AWS.
    - It's free to start and follows a pay-per-use model, which means you pay only for the services you opt for.
    - Interestingly, 80 percent of the Fortune 500 companies use Azure services for their cloud computing needs.
    - Azure supports multiple programming languages, including Java, Node Js, and C#.
    - Another benefit of Azure is the number of data centers it has around the world. There are 42 Azure data centers spread around the globe, which is the highest number of data centers for any cloud platform. Also, Azure is planning to get 12 more data centers, which will increase the number of data centers to 54, shortly.

5.10.2 Various Azure Services

Azure provides more than 200 services, are divided into 18 categories.

These categories include :

- computing,
- IoT,
- analytics,
- integration,
- security,
- media identity, and
- networking,
- migration,
- containers,
- management tools,
- databases,
- web services.
- storage,
- mobile,
- artificial intelligence, and other machine learning,
- developer tools,
- DevOps,

(a) **Compute Services**

- **Virtual Machine**

  This service enables you to create a virtual machine in Windows, Linux or any other configuration in seconds.

- **Cloud Service**

  This service lets you create scalable applications within the cloud. Once the application is deployed, everything, including provisioning, load balancing, and health monitoring, is taken care of by Azure.

- **Service Fabric**

  With service fabric, the process of developing a microservice is immensely simplified. Microservice is an application that contains other bundled smaller applications.

- **Functions**

  With functions, you can create applications in any programming language. The best part about this service is that you need not worry about hardware requirements while developing applications because Azure takes care of that. All you need to do is provide the code.

(b) **Networking**

- **Azure CDN**

  Azure CDN (Content Delivery Network) is for delivering content to users. It uses a high bandwidth, and content can be transferred to any person around the globe. The CDN service uses a network of servers placed strategically around the globe so that the users can access the data as soon as possible.

- **Express Route**

  This service lets you connect your on-premise network to the Microsoft cloud or any other services that you want, through a private connection. So, the only communications that will happen here will be between the enterprise network and the service that you want.

- **Virtual network**

  The virtual network allows you to have any of the Azure services communicate with one another privately and securely.

- **Azure DNS**

  This service allows you to host your DNS domains or system domains on Azure.

(c) Storage

- **Disk Storage**

  This service allows you to choose from either HDD (Hard Disk Drive) or SSD (Solid State Drive) as your storage option along with your virtual machine.

- **Blob Storage**

  This service is optimized to store a massive amount of unstructured data, including text and even binary data.

- **File Storage**

  This is a managed file storage service that can be accessed via industry SMB (server message block) protocol.

- **Queue Storage**

  With queue storage, you can provide stable message queuing for a large workload. This service can be accessed from anywhere in this world.

5.10.3 Services of Interest for IoT Analytics

Following are the services of azure for IoT

(a) Azure Data Lake Store	(b) Azure Analysis Services
(c) HDInsight	(d) The R server option

(a) Azure Data Lake Store

- Data stored in Data Lake Store can be analyzed using analytic frameworks within the Hadoop ecosystem, such as MapReduce and Hive.
- Microsoft Azure HDInsight clusters canbe provisioned and configured to directly access data stored in Data Lake Store.
- You can store a variety of data types for analytics and storage size is effectively unlimited. Individual files sizes can be from kilobytes to petabytes.
- Storage is durable as multiple copies are created and managed automatically.

(b) Azure Analysis Services

- Azure Analysis Services is built on Microsoft **SQL Server Analysis Services (SSAS)** and is compatible with SQL Server 2016 Analysis Services Enterprise Edition. It supports tabular models. Functionality includes DirectQuery, partitions, row-level security, bidirectional relationships, and translations.
- The same tools can be used for SSAS to create data models for Azure Analysis Services. Tabular data models can be created and deployed using **SQL Server Data Tools (SSDT)** or using some Azure templates in **SQL Server Management Studio (SSMS)**.

- Data sources can be connected to the Azure cloud and also to on-premise data in your organization. Connecting to cloud data sources is fairly seamless as it is essentially in the same local network from the perspective of the cloud server.
- Connecting to on-premises data sources is supported by an on-premises data gateway, which enables secure connections to your Analysis Services server in the cloud.

Fig. 5.10.1 : Some data sources are not yet supported in preview as of Microsoft December 2016. Image source : Microsoft Azure

- Data from Microsoft tools such as Power BI Desktop and Excel can be connected. Connectors can also be used to link custom applications and some browser-based tools.

Fig. 5.10.2 : Power BI Embedded is not yet supported in preview as of Microsoft December 2016. Image source : Microsoft Azure

(c) **HDInsight**

- Azure HDInsight uses the Hadoop components from the **Hortonworks Data Platform (HDP)** distribution. It deploys managed clusters in the cloud with a focus on high reliability and availability. Microsoft Active Directory is used for security and governance.
- HDInsight includes the implementations of Hadoop ecosystem tools, such as Apache Spark, HBase, Kafka, Storm, Pig, Hive, Interactive Hive, Sqoop, Oozie, and Ambari. It also integrates with **Business Intelligence (BI)** tools, such as Power BI, Excel, SQL Server Analysis Services, and SQL Server Reporting Services.

- Default storage for the HDFS used by HDInsight clusters can be either an Azure storage account or a Azure Data Lake store.

(d) The R server option

- In 2015, Microsoft bought a company called revolution analytics, which maintained a managed version of the open source R distribution. Since then, Microsoft has been integrating R into several of its software products and Azure has been no exception.
- HDInsight includes an option to integrate an R Server into your HDInsight cluster when the cluster is created. This allows R scripts to use Hadoop to run distributed computations.
- Microsoft includes a big data analytics R package called ScaleR.
- You can connect to your cluster and run R scripts on an edge node. You have the option of running parallelized distributed functions across the cores of the edge node server using ScaleR.
- You can also run them across the nodes of the cluster by using the ScaleR Hadoop MapReduce or Spark compute contexts.

5.11 THE THINGWORX OVERVIEW

- The company PTC, which has a long history in creating software for the world of machinestalking to machines, developed ThingWorx.
- It is an application development environment for building IoT solutions. It is a software platform that abstracts IoT devices and related components and services into model-based development objects.
- The platform makes it easy to model your devices, the data, and has the ability to quickly create dashboards through a web-based application. No code is required.
- ThingWorx is also extensible to third-party components through its marketplace. This makes it easy to add in a third-party functionality without special configuration. It can also integrate with both AWS and Azure IoT hub services.
- There are multiple components of ThingWorx.
- ThingWorx Foundation is the center of theplatform. It is divided into three areas, as shown in the following image :

(a) ThingWorx Core	(b) ThingWorx Connection Services	(c) ThingWorx Edge

(a) ThingWorx Core

- ThingWorx Core is a software platform environment that allows you to design, run, and implement analytics for IoT applications that control and report data from remote devices. These devices could be sensors, consumer electronics, or industrial equipment.

- ThingWorx uses a representational object-based design. This means that you create software objects to represent your IoT devices and other assets. The representation includes relevant properties and related data items.
- You then use the objects to create applications, which can monitor and manage your IoT devices. You can create dashboards, implement response logic, and integrate third-party applications.
- The ThingWorx Core is the hub of your ThingWorx environment. You logically define behavior and relationships between IoT devices or remote assets that are set up in your environment. Once the actual devices have been modeled in the software, they can register and communicate with the Core. You can then collect data and manage the physical devices.
- ThingWorx Core includes two main tools for you to create IoT solutions:
- **ThingWorx Composer**: This is a modeling environment where you set up theremote assets, business logic, data storage, and security.
- **ThingWorx Mashup Builder**: This is a drag and drop tool where you can quickly create dashboards and mobile interfaces without needing to write code. This is where you can do things like show a location on a map and chart sensor value trends.

(b) **ThingWorx Connection Services**

- ThingWorx Connection Services handle communication and connectivity between the Core and remote assets. Components can handle connectivity over different protocols and with different device clouds. They handle message routing to and from the remote devices and also message translation when required.
- Connection Services have connection adapters to link up devices that are using AWS IoT SDK or Azure IoT SDK. They link into the cloud providers and allow you to translate thedata into ThingWorx.

Fig. 5.11.1 : ThingWorx

- The Connector Services components have a core connection server and also an adapter. ThingWorx documentation refers to these two combined as a **Connector**. They are both packaged and installed together.
- Each Connector supports a specific protocol where inbound messages are translated into ThingWorx format and sent on to the Core. The reverse occurs for outbound messages from the Core to the remote device.

(c) ThingWorx Edge

- The third component of ThingWorx Foundation consists of a couple of software products that operate out in the edge of the IoT network.
- The first one operates as a small server and hub for communication back to the Core component at a centralized system. It is called **ThingWorx Websocket-Based Edge MicroServer (WS EMS)**. The second is a **Software Development Kit (SDK)** that your developer would install on your IoT device as part of your device software. It is called ThingWorx Edge SDK.
- WS EMS is a standalone application that is installed on a remote device. It uses a ThingWorx protocol called AlwaysOn to communicate with the ThingWorx Core. The WS EMS supports several operating systems and has a small footprint. It can work with a large number of devices to provide a way to establish communication between an edge device and the ThingWorx Core.
- Depending on what language your developers are using for your IoT device code, there are several versions of **ThingWorx Edge SDKs** that allow you to add connectivity to your device. There are SDKs that support C, .NET (C#), and Java languages, along with the ones for the Android and iOS platforms.

5.11.1 ThingWorx Concepts

The main objects and concepts used on the ThingWorx core to model environment are :

(a) Thing templates

- A **thing template** sets up and defines the base functionality that you build multiple things from. This defines a general category with a common set of properties, services, events, and subscriptions. These will be included in any thing definition that is based upon the template.
- The thing will *inherit* from the thing template.

(b) Things

- A **thing** is used to represent an object. It is based on a thing template but will often include additional properties, services, or events unique to the implementation of the more generic base thing, which was defined in the template.
- There are several types of thing:
  - **Things :** This represents a real-world asset, device, or system.

- **Remote thing :** A remote thing is a special type of thing that represents an asset in a remote location. When using a ThingWorx Edge SDK, the edge devices where your application is running needs to be created in ThingWorx Composer using the RemoteThing template.
- **Custom things :** Extensions provide custom thing templates that you can use to create the custom things to represent your devices.

(c) **Properties**

A property is a variable that represents a behavior of a thing. Properties have either of the following :

- **Remote bindings :** Remote bindings support egress. They can be written to the edge device when it connects to ThingWorx. The edge device must know of any Properties with Remote bindings.
- **Non-remote bindings :** These are not sent to the edge device when written and are not read from the edge device when a value is requested.

(d) **Services**

A service represents a function that a thing can perform. It can be defined as part of a thing, a thing template, or a thing shape.

(e) **Events**

An event is triggered from a change in the state or value of a property. It sends data to an object that subscribes to it.

(f) **Thing shapes**

A thing shape is an abstraction of concrete things. Typically, the thing shape is used by a thing template, which itself is used for thing definitions.

(g) **Data shapes**

- Data shapes define the structure of tables in ThingWorx, which represent informational data structures or the output of ThingWorx services. They are composed of field definitions.
- Some types of data shapes are here :
  - **Data table structures**: These are storage tables that have a primary key that can support indexing
  - **Stream structures**: These can access data continuously
  - **Value stream structures**: Data stored from a property bound to a thing
  - **Event data**: This stores data linked to events

(h) **Entities**

- Entities is the general term in ThingWorx for all the types of objects that you can create. They include things, thing templates, thing shapes, and data shapes.
- You can also importentities into ThingWorx.

5.11.2 Strategies to Organize Data for Analytics

- Data organization is the practice of categorizing and classifying data to make it more usable. Similar to a file folder, where we keep important documents, you'll need to arrange your data in the most logical and orderly fashion, so you and anyone else who accesses it can easily find what they're looking for.
- Good data organization strategies are important because your data contains the keys to managing your company's most valuable assets.
- An analytics strategy is part of a comprehensive strategic vision to specify how data is collected and used to inform business decisions. It is meant to provide clarity on key reporting metrics by: Specifying the sources and types of data that are collected and used for reporting

(a) Linked Analytical Datasets

- Linked Data is a set of design principles for sharing machine-readable interlinked data on the Web. When combined with Open Data (data that can be freely used and distributed), it is called Linked Open Data (LOD).
- An RDF database(Resource Description Framework) such as Ontotext's GraphDB is an example of LOD. It is able to handle huge datasets coming from disparate sources and link them to Open Data, which boosts knowledge discovery and efficient data-driven analytics.
- Linked Data is one of the core pillars of the Semantic Web, also known as the Web of Data. The Semantic Web is about making links between datasets that are understandable not only to humans, but also to machines, and Linked Data provides the best practices for making these links possible. In other words, Linked Data is a set of design principles for sharing machine-readable interlinked data on the Web.
- The concept of LAD ties together the well-established ideas of analytical datasets and relational databases. Combining them together accelerates how quickly data scientists can get to the part both you and they care about most—the analytics.

(b) Analytical datasets

- An analytic database, also called an analytical database, is a read-only system that stores historical data on business metrics such as sales performance and inventory levels.
- Business analysts, corporate executives and other workers run queries and reports against an analytic database. The information is regularly updated to include recent transaction data from an organization's operational systems.
- An analytic database is specifically designed to support business intelligence (BI) and analytic applications, typically as part of a data warehouse or data mart.
- This differentiates it from an operational, transactional or online transaction processing database, which is used for transaction processing, such as order entry and other business applications.

- While databases that do transaction processing can also support data warehouses and BI applications, analytic database vendors claim their products offer performance and scalability advantages over conventional relational database software.

(c) Building analytic datasets

- Analytical datasets are semi-denormalized tables. By semi-denormalized, this means including not just the ID code of a field but the description for it as well. Categories based on value can be created in ranges and include these as separate features.
- Analyst can focus on efficiently storingvalues, as it would be with purely relational database design. Prebuilding the transformed datasets that an analyst would be building using SQL to preprocess a dataset in preparation to train an ML model anyway.
- As data wrangling takes 80-95% of a typical analysts time, having most of this already built, tested, and the business logic incorporated saves a tremendous amount of time. And this happens not just once for one person, but many times, for many people. This compounding effect dramatically increases the value of prebuilding the analytical datasets.
- The following process will help in deciding how to build an analytic dataset for a particular topic. For example of GPS positional data from an IoT device. Each instance in the dataset will be a combination of time and location:

1. Determine the resolution of the data

- What level of aggregation is needed?
- Will an individual record be at a device level, a reported instance level, or a time period level?
- This should be determined by what makes the most sense based on the incoming data resolution, and how it will probably be used for business purposes. In our example, the resolution is at a GPS position reporting frequency,which is every 10 seconds.

2. List out all the variations, categories, calculations, and descriptions that are added or transformed from the data

- This should be what your team finds useful for modeling. This can also result from discussions with business experts on whatcould be useful to them.
- In our GPS position example, it is as follows :
  - Latitude
  - Longitude
  - The day of the week
  - Time since previous GPS position record
  - Speed (calculated from previous rolling set of records)The current time zone offset
  - Daytime or nighttime
  - GPS grid identifier

- The exact time at UTCThe exact local time
- The current state of the device (driving, idling, or parked)

3. **Review each item for how often they are likely to be used in analysis, ML modeling, or reporting**
   - Decide if the cost of creating and storing the informationis worth it for how often it is used. In our example, the exact local time and the day of the week were eliminated.
   - This is based on the expected frequency of use versus the storage and computational costs of keeping the information.
   - This is a balancing act and your decision may shift over time as different fields become more or less valuable to your business.
4. **Create data transformation code that automatically creates and maintains theinformation in one table**

   The goal is to do this in an automated fashion so thatthe data scientists do not have to recreate it each time they need it.
5. **Create a unique identifier for each record, if it does not already exist**
   - In this case, it would be the exact UTC time combined with the unique device identifier,so there is a need to create a separate ID field for the combination.
   - This is done to make life easy for the data scientist, who should not have to worry about complicating their analytics by connecting datasets using two different fields. Combine it into one to simplify things for them:

UTC|DeviceID	DeviceID	UTC Time	Latitude	Longitude	Time since previous (sec)	Speed (km)	Current time zone offset	Day/Night	GPS grid identifier	Current State
2017061903154[illegible]	0879991B1	6/19/2017 3:15:04	41.861832	-87.623177	15	0	-5	N	U2345	Parked
2017061903154[illegible]	0879991B1	6/19/2017 3:15:24	41.861834	-87.623181	20	5	-5	N	U2345	Driving
2017061903154[illegible]	0879991B1	6/19/2017 3:15:44	41.861828	-87.623182	20	20	-5	N	U2345	Driving
2017061903160[illegible]	0879991B1	6/19/2017 3:16:04	41.861841	-87.623167	20	25	-5	N	U2344	Driving

Fig. 5.11.2 : Analytical dataset example in table form

(d) Linking together datasets

- Building the analytical dataset in a way that makes it easy to connect to others. This is typicallydone by creating a new or using an existing identifier key. The identify key would be the same in both datasets.
- It is also closely related to the star schema design in Online Analytical Processing (OLAP) datawarehousing. The goal is, however, somewhat different.
- With relational database design, the goal is to minimize or even eliminate data duplication by denormalizing the data into multiple linked tables. Denormalizing separates the identifier from its description in different tables, so that the description is only stored once.
- With star schema design, the goal is to make drill down, drill through, and predefined metric calculations easy and fast. Datasets are stored and linked along dimensions such as time, category, fiscal year, and company divisions.

- However, with LAD design, the goal is to minimize joins while still allowing easy creation of hybrid datasets, not previously conceived. The goal is to minimize data transformation work for the IoT analyst, who is building training sets for ML modeling.
- The tradeoff is in data size, the initial ETL complexity when developing the analytical datasets, and the duplication of data.
- The following diagram shows a simple example:

Fig. 5.11.3 : Linked Analytic Dataset design

- The cost of storing large datasets has dropped dramatically when using big data system, such as HDFS or S3.
- The cost of missing an IoT analytics business opportunity due to your data scientists being tied up with data munging can be very high.
- It is a worthy tradeoff and can greatly accelerate the iteration time for new ML model development.
- Follow these steps to identify and build links between analytical datasets:

1. **Identify fields that can create a bridge to other analytical datasets**
   - These datasets may either be already created or are being considered for creation.
   - In GPS data example, the combination of latitude and longitude identifies a location. Certain locations, such as rest stops, distribution centers, and fueling stations, have useful and possibly predictive data tied to them.
2. **If necessary, combine multiple fields to create a single field that identifies the linkage**
   - Big data systems handle single field joins much better than multiple fieldjoins. It also makes it much simpler for the data scientist to use and therefore less likely to make a mistake in combining datasets.
   - In the example it can be combine a slightly rounded latitude and longitude value into a single identification field and store it in a separate field in your dataset.

- The rounding is to adjust for extremely precise GPS values that are at the location, but just in different areas of the parking lot.
  1. Repeat for all linkable fields in the dataset: The GPS grid identifier is anothercandidate.
  2. Add the dataset and its links to a master diagram to use as reference.
- The following diagram shows GPS example can link to otheranalytical datasets:

Fig. 5.11.4 : Simple LAD example

5.12 MANAGING DATA LAKES

- A data lake is a central location that holds a large amount of data in its native, raw format. Compared to a hierarchical data warehouse, which stores data in files or folders, a data lake uses a flat architecture and object storage to store the data.
- A Data Lake is a storage repository that can store large amount of structured, semi-structured, and unstructured data. It is a place to store every type of data in its native format with no fixed limits on account size or file. It offers high data quantity to increase analytic performance and native integration.
- Data Lake is like a large container which is very similar to real lake and rivers. Just like in a lake you have multiple tributaries coming in, a data lake has structured data, unstructured data, machine to machine, logs flowing through in real-time.
- Object storage stores data with metadata tags and a unique identifier, which makes it easier to locate and retrieve data across regions, and improves performance. By leveraging inexpensive object storage and open formats, data lakes enable many applications to take advantage of the data.
- Data lakes were developed in response to the limitations of data warehouses. While data warehouses provide businesses with highly performant and scalable analytics, they are expensive and proprietary and can't handle the modern use cases most companies are looking to address.

- Data lakes are often used to consolidate all of an organization's data in a single, central location, where it can be saved "as is," without the need to impose a schema (i.e., a formal structure for how the data is organized) up front like a data warehouse does.
- Data in all stages of the refinement process can be stored in a data lake: raw data can be ingested and stored right alongside an organization's structured, tabular data sources (like database tables), as well as intermediate data tables generated in the process of refining raw data.
- Unlike most databases and data warehouses, data lakes can process all data types — including unstructured and semi-structured data like images, video, audio and documents — which are critical for today's machine learning and advanced analytics use cases.

5.12.1 Data Refineries

- Data refining is a concept that was introduced a few years back to address the challenges in big data in terms of refining data and monetize the data in the repository providing fast information to enable real-time decisions and actions.
- Data refinery is a process which collects data, enriches it and create an integrated refined data repository which can be used for analysis to take actions. In the age of big data, companies are looking for ways to take on the ambiguous data. Data Refinery plays a crucial role in removing this ambiguity and bringing in more informed decisions and accuracy.
- Data science solutions for a company depend on the size, the business problem at hand, available expertise and the volume of data that needs to be refined.
- Organizations, large and small are looking for solutions to meet their specific business problems. Instead of having their own data science unit, many are choosing to collaborate with external bodies that can provide the necessary expertise.

The refining steps to meet the analytics demands

- **Collecting process** is where data is gathered and information is measured on variables of interest in a systematic way that enables one to answer stated research questions, test hypotheses and evaluation.
- In this process, issues are promptly identified and goals are set. Analysts should already have planned the methods and approach despite the massive and data ambiguity.
- **Data enrichment** is merging third-party data from an external source with an existing database. Technically, it is the process of taking external information and overlaying it into your current files. Despite these large amounts of data, efficient parallel algorithms are needed to maintain clean data.
- **Analysis.** A refined data now come to analysis stage where it enables organizations to provide real time insights with a measured assurance for success to be able to compete in the fierce big data market.
- Furthermore, using real-time analytics, companies can then generate new and bigger revenue streams and identify risk factors and pain points as well as predicting success.

5.12.2 Developing a Progression Process

- Setting up a progression process will help manage this dataset. Data science is highly iterative, which makes it difficult to find a clear point that signals a change in state, as with normaldatabase development projects.
- A way to handle this is by setting up regular review periods where a team determines which datasets are ready to progress to the next stage of development. Decide how often this should happen, define the stages and the corresponding requirements for each, all based on your unique situation.
- A suggested progression path is reviewed that can be tailored to the needs of unique IoT analytics environment.
- Data lake is segmented into three general areas :
  1. **Sandbox :** A data scientist has full access to read and write in this area. They may have a sandbox to themselves in addition to one for the team. This is for initial experimentation and model development.
  2. **Mature :** A data scientist has full access to this area but does not have their own mature environment, the team shares it. All code and scripts used to generate datasets kept in this area should be under source code control.
  3. **Production :** A data scientist has full read access but no write capability. Dat asetsin this area have been fully tested, code and scripts used to generate datasets areunder source code control, and a change control process is in place.
- The suggested progression process to move datasets between areas are
  - **Establish a regular recurring review of your datasets :** This could be every month, every quarter, or semi-annually. Use whichever makes the most sense forhow quickly your IoT analytics is developing. But the review should be regular and enforced. It is too easy to delay due to the small iterations that occur with analytics. Avoid this trap and force your team to do it.
  - **Review the datasets in all three areas for any that should be deleted :** These could be development datasets that never made it far, or old versions of ones that are now in production. Take a cue from Java and have a regular garbage collection event to keep your data lake optimized.
  - **Review Sandbox datasets that are ready to move to the Mature area :** These are the ones that are either project specific and ready for team testing, or are useful tothe team for their general work. The latter should be set up as regularly recurringand scheduled jobs to build the datasets. This type of datasets will help acceleratemany projects in the future.
  - **Review Mature datasets for ones that are ready to move to Production :** These are typically more project-specific and have passed all the testing. Once it is moved into production, future changes should be minimal. At this point, control of the dataset should be handed over to a separate group to maintain and provideservice-level support.

5.13 THE DATA RETENTION STRATEGY

- A data retention policy is a company's established protocol for keeping records for a set period of time. It may also be called a records retention policy or backup retention policy. The goal is to secure your data and ensure compliance with particular business needs, industry guidelines, or legal requirements.
- A comprehensive data retention policy and records management plan detail the reasons a company wants to retain specific records and where it will store or archive this data. The policy should also include information about who is responsible for each type of data and how it will be deleted (or purged) at the end of its retention period.
- A data retention policy should also specify backup storage procedures to help a company recover if they experience data loss.
- Even big data eventually gets too big and costly to maintain. Remembering the goal of minimizing costs while still maximizing value, make sure to develop a retention strategy for IoT data. Data could be simply deleted after it is retained for a certain amount of time.
- There are other options that allow you to retain value of the data while minimizing the costs.

5.13.1 Goals

The goals of a retention strategy for IoT analytics are two fold :

1. **Maintain Value**
   - Advanced modeling techniques, such as deep learning, need lots of history to maximize prediction effectiveness.
   - It is also difficult to know ahead of time which fields will be valuable for a future unknown project.
   - The traditional data retention strategies of storing records for a fixed period of time and then deleting the full dataset could result in lost profitable opportunities.
2. **Minimize Costs**
   - IoT data can get big quickly. Even with cloud-based HDFS storage, costs can get large. Keeping everything forever could easily cost morethan the value it provides.
   - The more accessible data is, the more costly it becomes.
   - There should be some compromises to keep your costs low.

5.13.2 Retention Strategies for IoT Data

- Three strategies to reduce the storage size of data.
- These can be used individually or in combination.

(a) Reducing accessibility (b) Reducing the number of fields (c)Reduce the number of records

(a) Reducing accessibility

Costs significantly reduced without deleting any data by reducing the relative accessibility of the data.

Compression

- Compressing data retains all the information at the (typically) but time to access it is increased.
- Using compression formats such as Avro and Parquet can significantly reduce storage size (and therefore costs) in Hadoop clusters and S3 folders, while oftenimproving performance.
- The performance can be improved with proper design of the file format.
- HDFS supports other compression formats such as GZIP and Snappy. This should be the first thing you do to reduce the file size.

Changing the storage technology to lower-cost options

- Keep the data, but moveto lower-cost methods.
- If the data stored and not accessed often. Then storage can be changed from SSD- to hard disk-backed storage or even to tapes were cost of storage is low.

Changing accessibility service levels

- This method suggests cloud storage services and is analogous to changing storage technology (although you can do that also in the cloud).
- For Amazon S3, this could be a change to Standard–Infrequent Access level service or to Amazon Glacier for very infrequently accessed data.
- S3 allows automated scheduling of when files should be moved into lower service levels based on rules, such as the age of the file.

Changing redundancy levels

- HDFS keeps multiple copies of files for durability.
- The standard setting is three copies, but this is configurable.
- The redundancy level can be changed for less valuable files and save some costs. Amazon S3 alsohas a reduced redundancy option.

(b) Reducing the number of fields

- In IoT data there may be some fields which are more valuable than others.Using *Data Science for IoT Analytics*, those fields (features)can be found and that keep coming up as statistically significant, and some that just never seem to matter.
- For older files, some methods can be used to keep the useful fields, and removing those do not have an impact of the data.
- Following are the way to remove the unwanted fields.
  - **Transform older data to only keep useful fields :** Move older data to either a new file or table but only keep those useful fields. Then, delete the old records.

- **Split out useful fields and treat them differently :** For older records, you could keep the useful fields in hot areas that are easily accessible, while move the less useful fields off into cold storage
- **Summarize and remove large data fields such as text or binary files (such as image or sound) :** Reduce lengthy free-form text field to the count of key word occurrences.

(c) **Reduce the number of records**

- Delete old records which are not need much. There are some other options where you can at least retain some possibly useful information.
- Some other options are listed next:
  - **Deleting raw data but keeping the refined versions of it :** After a certain amount of time, having both the raw data and the refined version probably does not add much value. older raw data files can be deleted without losing much value.
  - **Summarizing old data :** Instead of deleting the data completely, summarize individual records into weekly or monthly summary values. The appropriate summary statistic may be an average or a sum, or preferably, an entire set of summary statistics. For example, you may want to save the mean, the standard deviation, the maximum, the minimum, and the record count. This can reduce thousands of rows into one, so it is perfectly fine to have a much wider resulting dataset.
  - You will be losing some value in the data, as you will not have full resolution. But you are still retaining some informational value in the summary statistics. You could even use the statistics to simulate individual records later ifyou had to.
  - **Deleting old data :** This should be the final and last option. Remember, analyticsis built from the data and once that data is gone, the analytics you could have done goes with it.

5.13.3 The Retention Strategy Example

Let set up a retention policy for GPS position data.

- The raw data is landed into an HDFS (**Hadoop Distributed File System**) table and kept uncompressed. A series of data refineries transforms the data into cleaned and useful analytical datasets. All of the analytical datasets are compressed in Parquet format–still in HDFS.
- Upon the review of the raw data, it was determined that any problems in the initial transformation are typically discovered within a week. No problems were found more thana month later.
- Due to this, the raw data was retained for the latest month, and then moved into S3 Standard. In S3, a ruleset was created to move the data into Standard-Infrequent Access after another month, and then into Amazon Glacier a month after it.
- After another three months, the data is scheduled to be deleted from Amazon Glacier.
- After several months of statistical analysis and modeling, several key fields were identifiedin each analytical dataset. It was also discovered that data records from more than two years prior were rarely accessed. Records from more than three years prior were never accessed.

- The data from two years prior was moved out of HDFS into S3 Standard. Data between two and three years old was transformed to only keep the key fields.
- The older data is moved to Amazon Glacier. After four years, the data is summarized at a daily aggregation with metrics on record counts, starting, ending, and average position for each device along with distance traveled and the travel time.
- This data is kept in S3 Standard as it is 10,000 times smaller in size. The full data is then removed.

5.14 COMMON MISTAKES WHEN DESIGNING VISUALS

- Designing visual is challenging thing, because it need to catch audience. Charts and dashboards tend to be done as an after analytics.
- The quality of analytics is determined by what they see first – ie visuals. For someone to be willing to use it, they must understand it and be engaged by it.
- Following are the common mistakes when designed are made
- **Assuming others know the data intimately :** Following are some ways to address this :
  - **Always, always, always (always) label chart axes so your audience know what is being measured**: Some charting softwaremakes this difficult as the default setting inexplicably does not show axis labels (Eg. *Microsoft Excel*).
  - **Avoid using abbreviations and acronyms**: Make this clear in order to minimize opportunities for misinterpretation. The acronym ETDcan mean Estimated Time of Departure or Explosive Trace Detection, both would mean very different things when analyzing airport data.
- **Focusing on the analytics instead of the conclusions :** Following are some ways to help them :
  - **Make the conclusions obvious :** Highlight, circle, and point a big arrow at the important information. Do not make them have to figure it out for themselves. In a dashboard, the important information should draw the eye first.
  - **Hide the complexity:** If it does not add information useful todrawing a conclusion, hide it.
  - **Start with the end :** Show your conclusions first, then follow up with the explanation of what led you there. The same is relevantwith a dashboard; make the most prominent visual the one that will drive a user to make an action. Follow it up with the supporting information.
- **Not considering how the analytics will be used :** Following are somethings to think about :
  - **Simplify for management reporting :** Executives have a lot of information thrown at them every day, from a variety of sources.Make yours simple to understand with clear trend lines and red, yellow, and green symbols indicating status.
  - **Add more detail for daily operations users :** They will need as much relevant information as you can display in single view without it being too busy to interpret. Find the right balance, butlean toward showing more useful data.

- **Making it difficult to answer follow-up questions :** Following are some suggestions to avoid this mistake :
  - **Anticipate the follow-up questions :** Have the answer either already visible or just a simple click away.
  - **Layer the views :** Link simpler summary visuals to more detailed visuals, such as tying a top 10 Pareto chart to monthly trends.

Example : To illustrate how to transform a bad visual into a good one

Fig. 5.14.1 : Example of Bad Visual - putrid chart

Following are problems associated with this visual chart :

- Dark backgrounds are distracting unless the entire theme of a presentation or dashboard is dark. Neither of the axes are labeled.
- Which line is more important is not clear, Since chart gives the impression to the audience that all are equally important.
- A character code is used in the legend instead of a more descriptive name. If someone is not intimately familiar with the code, they will have to look up the descriptions for them. 3D was used in a way that does not provide any informational value. It distracts attention away from the key data.
- Labels are applied to values without any rounding. This makes the chart cluttered and implies a precision that may not be accurate. If this is a chart reviewed monthly, the historical trend is important, but the exact values are not - with the exception of the currentmonth.
- **Conclusion :** The chart is both difficult to interpret and visually confusing.

Following is improved version of the same chart:

Fig. .5.14.2 : Example of Good Visual – Putrid Chart (Much better visual, the key trend is obvious and the current month value is clearly stated)

Changes as compared to previous Chart

- This visual is much more clear about which trend line is the most important one. Lines are also clearly labeled so the audience does not have to reference a legend.
- The current monthvalue, which is the number reported on each month, is clearly visible without an implied extreme precision.
- Both axes are labeled so that the audience does not have to guess at what is being measured.
- The axis labels for the month are perpendicular, which visually aligns with the value for themonth.
- It takes a little more work to format a visual in this way, but the audience will understand the information faster and more clearly.

5.15 THE HIERARCHY OF QUESTIONS METHOD

- When designing visuals and dashboards, instead of just replicating the same charts and tables that you created for yourself when exploring the data, take a minute to think about things from the point of view of the audience.
- As an analyst,
  - The data and environment already well known;
  - There is no need of labels and descriptions on your charts.
  - What needed is as much information which can be fit into one place, and patterns can be easily found.

- But the people who will be either interacting with dashboards or viewing your presentation have different needs.

- What People or audience need is
  - They will want to be able to orient themselves visually with minimal effort.
  - They want the key conclusions to be obvious.
  - They do not want to have to spend a lot of time trying to figure it out, or have to ask a lot of questions just to understand what is being shown.
  - Theywant simple and familiar, but with enough detail that they feel confident that the data is backing up their interpretation.
  - It is useful to follow a framework when designing analytics interactions for others. This will help to organize your thoughts and plan out what visualizations and dashboards are needed to support analysis by other people.

5.15.1 The Hierarchy of Questions Method Overview

- A process to plan out visualizations, also called as **Hierarchy of Questions method**. Tis method stated that instead of starting with requirements and mockups, start with mapping out the thought process of the audience.
- Find out the answer the questions identified in the mapped thought process which will be data for us. The data should be in a form that can be queried efficiently and in multiple different ways.
- Finally, visuals are created which are aligned to the mapped thought process. The same data will probably be used multiple times in different ways.
- The benefit of following this process is that it puts the needs of the audience first, without anchoring to a starting visualization - which is likely to be whatever was easiest to create.
- The tendency, in that case, is to do what is simplest for the developer (you) to create and maintain, as opposed towhat is best for the audience.
- Developer is one person while the audience/user are many;, each person will have to use dashboard many times. The benefit of getting it right can be large, especially for customer-facing dashboards, which could have thousands of users.

Fig. 5.15.1 : General process for the hierarchy of questions method

5.15.2 Developing Question Trees

- **Step 1 :** Process to Prepare List of Questions
- The first step in the process is to prepare a list of questions that the visualizations in a presentation or dashboard should answer. This can be done with a small group of end users.

- If it is not possible or you are preparing a small presentation, thinks as audience what they expect.
- Prepare list of questions keeping minds audience and List them out.

▶ **Step 2 :** generalize and consolidate the list of questions

- The next step is to generalize the list of questions into the underlying concepts of what is really being asked for each one.
- Some questions may get consolidated together. The idea is to get to a concept that can be applied over many situations versus a specific situation.

▶ **Step 3 :** Review and Indentify the list of questions

- Then, review the list and identify the questions that are the starting points.
- Starting points are the reasons that would motivate one to initially go to a dashboard or sit through a presentation in the first place.
- Then, identify which questions are really follow-up questions to the starting point question.

▶ **Step 4 :** Arrange/Organize the list of questions

- Arrange the questions in a hierarchy with the starting points at the left linked to their follow-up questions out to the right.
- If the same follow-up question relates to multiple starting point questions, duplicate it for each. It may be subtly different in each scenario.
- We will refer to the resulting diagram as a **question tree**. This can then be used as a reference when building the datasets and visualizations.

Example : To demonstrate how this process works, using movie time information to walk-through the process. Most people can easily relate to it.

▶ **Step 1 :** The first step is to list out the questions that would lead you to want to know your movie time information:

- When is *Satyamev Jayate* playing?
- What theaters are playing *PM Modi Picture Show*?
- Is the cinema sold out for *PM Modi vs. silent Priminister* at 8:10 pm tonight?How many seats are left?
- How much does a ticket cost at that theater?How far away is Cinemax?
- How long will it take to get to Cinema Cheap?What movies are at Luxury Cinema tonight?
- What movies are playing at around 9 pm tomorrow night?

▶ **Step 2 :** The next step is to generalize the questions and consolidate:

- What times can I see the movie I want to see? What theaters are playing the movie I want to see?
- Will I be able to get good seats at a specific theater and time?
- Will I be able to get good seats at a specific theater and time? (consolidate)What is the price of a ticket at a specific theater and time?

- What is the travel time to a specific theater at a specific time?
- What is the travel time to a specific theater at a specific time? (consolidate)What movies are at the theater I want to go to?
- What movies can I see at a certain time?

Step 3 : Identify the main starting point questions. It is helpful to start documenting your reasoning for choosing the questions. In this example, there are three variables: movie, place, and time. You generally have decided on one of those already and need information to help you decide on the other two:

- What times can I see the movie I want to see?

I know the movie I want. I'm dying to see Satyamev Jayate ; I just need to know which times I can see it. Then, I'll want to know where it is playing and whether I can still get good seats for me and my friends. Ticket price and how long it will take to get to the theater would be helpful too.

- What movies are at the theater I want to go to?

I know the place I want. I love going to Luxury Cinema; it is such a great experience. I just want to know if there are any movies there that I want tosee. Then, I'll want to know the times and whether I can get good seats; I don't want to pay top dollar and not even be able to sit next to my girlfriend.

- What movies can I see at a certain time?

I know the time I want. We're going out to dinner with reservations at 7pm, and I want to see if there are any movies in the area I can see afterwards. I'll want to know the price and whether I can get good seats,and how long it will take to get there from the restaurant.

Step 4 : The final step is to organize the questions into a question tree. The following diagram shows a question tree for the movie time example:

Fig. 5.15.2 : Movie time information question tree example

Summary of the steps in creating your question tree hierarchy

1. List the questions the audience wants to ask of the analytics.
2. Generalize the questions and consolidate.
3. Identify starting point questions.
4. Organize the questions into a question tree diagram.

5.15.3 Pulling together the Data

- IoT data is typically stored using big data technology such as Hadoop (HDFS specifically).
- Joining together tables in SQL statements for dashboards typically decreases performance, sometimes significantly in these systems.
- Pull together the data into as few tables as possible, so the table joins are only done during batch processing instead of every time a dashboard is used by someone.
- When consideringwhat information to include in the dataset, reference the question tree and include what is needed to answer the questions.

5.15.4 Aligning views with Question Flows

- When designing visuals, align them with the question tree hierarchies.
- The first and most visible visual should answer the starting point question.
- The follow-up questions should beaddressed in the same visual, if possible, but less prominently.
- If this is too unwieldy, then the follow-up should be a simple click away for a user to answer his question. The same is applicable with the follow-up question to the first follow-up question, and so on.
- You may need to do a different dashboard for each hierarchy tree if it is too cumbersome to answer in the same dashboard. The test here is to think what is easiest and most natural to auser.
- The goal is for the dashboard or presentation order to follow the thought processes of the audience.

5.16 DESIGNING VISUAL ANALYSIS FOR IOT DATA

- To visualize the data means to make charts out of numbers. The goal of data visualization is to provide a graphical representation of the data so analysts can identify patterns and trends.
- Data visualization shows great efficiency when it comes to large series of data. Its methods include graphs, bar charts, pie charts, status tables, maps, line graphs, scatter plots, and much more.
- These charts contain a lot of condensed information that allows analysts to see the data trends easily and make the right conclusions. Such conclusions can be priceless and bring new opportunities to businesses.
- Some important considerations are reviewed when designing visual analysis with specialattention for IoT data.

5.16.1 Using Layout Positioning to Convey Importance

- Layout position implies what is important and primary; taking advantage of this to help communicate more effectively with your audience.
- Generally we start reading from upper-left position of a view. Hence put the key message here: the answer to your starting point question.
- As we move towards right and lower in a view, put the visuals (including tables and text) in order of its position in the question tree.
- In other cultures, apply the same concept but follow the position order that aligns with reading order. For example, in Arabic the language Urdu is written right to left - so the most important information should be in the upper right.

Fig. 5.16.1 : Recommended layout order when showing multiple visuals in same view

5.16.2 Use Color to Highlight Important Data

- Color is also a powerful way to communicate to an audience what they should interpret as important.
- For example generally to high some text many times Red Color is used which is very catchy
- But using many colors in the same view are confusing and diminish the impact.

The impact of using a single color to communicate importance

- In order to reiterate how powerful color (or shades of gray if only grayscale is available) can be,
- For example in the following image, look through the numbers and counthow many 6s you see.
- Its difficult to count and also it will take long time and even we may miss some count.

83291381337794749
91422691539122696
17946686489967647
35594183184243557
62144473513428797
96375323638591958

Fig. 5.16.2 : Random set of numbers. Count the number of times 6 is shown

- Now, let's take the same image and put the important information (the 6s) in the foreground using color and the less important information (the other numbers) in thebackground by lightening to a shade of gray.

- Looking at the changed image, know its easy to count 6s m it takes less time and chance of missing is also less. This is impact of color.

83291381337794749
91422691539122696
17946686489967647
35594183184243557
62144473513428797
96375323638591958

Fig. 5.16.3 : The same random set of numbers with the number 6 highlighted in color. Adapted from Cole Nussbaumer Knaflic's book *Storytelling with Data*.

- If u see of advertisement of Coca-cola , MacDonald, in their logo and Signages colour play very important role.

5.16.3 Be Consistent Across Visuals

- In a presentation or in a series of dashboards, make sure to keep your colors and formatting consistent.
- If the India Population is shown as a solid Red line in the first chart, make sure it is also a solid red line in all following charts. Keep colors consistent even if the typeof chart changes.
- If a Humidity sensor trend in a line chart is Green, make sure that in a following scatter plot, the points are also green.
- This consistency saves your audience from having to reorient themselves between views. It also helps prevent mis interpretations as it is easy to assume the same color has the same meaning.

5.16.4 Make Charts Easy to Interpret

- For IoT data, time series analysis is very common. Analyzing trends in IoT sensor values over time is useful for people in roles such as technical support, marketing, quality, and engineering. Time series charts communicate trend data more effectively than pareto chartsor pie charts.
- When creating charts, keep some things in mind to help make it easy for your audience to draw the conclusions that you intended.
- In most cases, See to that the data in far more ways than you will be making available to audience.
- Pick the charts that best convey what you have understood from your analysis of the data. Make it easy for your audience to grasp.

Some of the key points which will be help full:

- **Accentuate the key data**

  Make the key trend line in a chart bright and bold. In a table, bold the row you found to be the most important. Make it obvious. You are not insulting their intelligence, It saves them time and minimize misinterpretations.

- **Label chart items clearly**

Give the chart a clear title and make the font big enough to read easily. Make sure chart axes are labeled and avoid abbreviations.

- **Point out key information**

Circle an area on a chart you want to make sure the audience notices. Draw an arrow pointing to it and add text that states how they should interpret it. If there is a spike in average temperature in December due to record high regional temperatures, and not due to a system issue, circle it and add a note. It will save the audience from having to ask the question or investigate it themselves.

5.17 CREATING A DASHBOARD WITH TABLEAU

- A car dashboard provides real-time information about a car's speed, fuel volume, RPM, and other engine-related indicators. Similarly, a data dashboard provides information about company historical sales, key performance indicators (KPIs), sales growth, operational indicators, and customer feedback. This information is presented in a precise manner so that managers or executives can understand the situation and make appropriate decisions.
- There are hundreds of moving parts in your business and a dashboard summarizes these events into an easy-to-understand, real-time data visualization. These visualizations and charts can be used to make fast and effective decisions.
- Tableau makes it easy to assemble a dashboard from the visual analysis tabs you have already created.

There are several benefits of dashboard reporting :

- **Usability :** a typical company generates gigabytes of raw data daily. Understanding the data can help companies create value from it and make better decisions. Dashboards provide access to all key metrics on a single screen, turning raw data into valuable insights.
- **Access to data :** a single dashboard has access to multiple data sources to provide detailed reports of the inner workings of a company.
- **Decision making :** managers or executives can view anomalies, forecast sales, and review historical data to come up with business strategies. The information is available in an interactive visual form, where we can dive deep into historical data or filter out critical parameters.
- **Accountability :** it provides an unbiased picture of how well your company is performing. The dashboard can show you the difference in growth percentage and how you may have failed at a certain marketing campaign. Accountability is necessary to keep companies away from bankruptcy.
- **Interactivity :** the gamified and dynamic experience of the dashboard makes it easy to use and understand various factors of organizations. You can filter, isolate a single metric, zoom into a map or time series line plot, search for terms or even use third-party tools to generate anomaly alarms.

- **Analysis :** you can use these dashboards to come up with detailed analytical reports. The dashboard simplifies data analysis tasks as you are monitoring key permanence metrics and making sense of past events.

5.17.1 The Dashboard Walk-Through

- There is also web server software for Tableau, called **Tableau Server**, that allows you to easily publish a dashboard from the desktop software so that it is viewable through a browser.
- Other users can then easily interact with it. Although Tableau Server is outside the scope of this book, you would most likely want to publish the dashboards you create to it.

5.17.2 Hierarchy of Questions Example

- Let's take example assuming the audience is a government water use planning group for the State of Maharashtra.
- The group wants to understand how many weather stations are reporting precipitation information and a sense of how well the information isbeing captured.
- The following steps show how this can be made into a question tree:

Step 1 : List out questions the audience wants to ask of the analytics:

- Did the number of stations reporting precipitation change from lastmonth?
- How many stations in total have sent precipitation numbers?Where are the stations located?
- How many have reported a significant rainfall?
- Is there anything weird in the daily sums for each station thatmight indicate a problem?
- Which areas of the state had some good rain last month?

Step 2 : Generalize the questions and consolidate:

- What is the trend of stations reporting usable precipitation data?
- How many stations report data?
- Where are the stations that reported usable data in the period?Did a station report significant rainfall in the period?
- Are there abnormal precipitation values for a specific station? Where are the stations that reported usable data in the period?(consolidate with the similar question earlier in the list)

Step 3 : Identify starting point questions:

- Based on conversations with the water planning group, you determine their first thought is to simply check the number ofstations that report 15-minute precipitation data.
- Starting point question: How many stations report data?

Step 4 : Organize the questions into a question tree diagram:

Fig. 5.17.1 : Weather station question hierarchy

5.17.3 Aligning Visuals to the thought Process

- What visuals are needed for and how to arrange them before creating them in Tableau. The most important thing, the starting point, should be the first thing the audience views. If there are to be multiple visuals in the same dashboard, it should be in the upper- left part of the screen.
- Since a simple count is all that is really needed to answer the question, this can just be a number. No chart is necessary. You should make it large, so it is instantly noticed. Make it what may feel uncomfortably large, and it will be obvious to the audience without needingto point out that it is the most important piece of information in the view.
- A trend of the number of reporting stations is needed so the audience can understand if more or less are reporting usable data over time versus the total number. This aligns with the first follow-up question. A monthly trend should work.
- Then, for any given period, the water planning group wants to know where the reporting stations are located and whether there was significant precipitation for the period at the stations. They also want some idea of locations in the state with significant reported precipitation.
- A map is a good way to fit in a lot of spatial information in a way that can beeasily comprehended. This aligns with the secondary follow-up question.
- For any station or group of stations in an area, the planning group wants to be able to see daily trends, so anything unusual can be detected quickly for additional investigation.
- Thisaligns with the last follow-up question. Now, we can start building the views one at a time,then assemble them into a dashboard.

5.17.4 Creating Individual Views

- The starting point question can be answered by counting the number of unique stations in the dataset regardless of the validity of the values. Set up a calculated field called **Number of Stations** using the formula, `COUNTD([Station])`, which will aggregate based on a distinct count of unique stations. Drag the **Number of Stations** aggregation to the text box in the **Marks** shelf. Makethe number big and bold, so it will be seen first.

Fig. 5.17.2 : Total stations count view

- The next view should align to the follow-up question on the trend in the Number of Stations reporting valid values. Since the audience is interested in longer-term trends, a monthly grouping is appropriate. At this point, let's rename Qpcp to a better description. From the dataset documentation, the value represents the amount of precipitation measured in inches. We will rename Qpcp to Amount of Precipitation (inches). Also filter out records with the extreme values in either Qgag or Qpcp (the latter now called Amount of Precipitation (inches)).
- Set up the measurement date to show month and year. Drag the **Number of Stations** field to **Rows** shelf to show the trend. There is an implicit comparison to the total **Number of Stations**, so make the values visible by dragging **Number of Stations** to the **label** box in the**Marks** shelf.

Fig. 5.17.3 : Monthly trend view

- The next follow-up question in the question hierarchy is about the location of the stations. Use the mapping functionality to convey location information. You can also answer the other follow-up question about which stations reported significant precipitation for the period in the same view using color. Create a calculated field using the following formula, and drag it to the color box in the **Marks** shelf. The value to use as the breakpoint for what is considered significant should be determined with the audience. We will use **0.2** inches forthis demonstration:

```
IF SUM([Amount of Precipitation (inches)]) >= 0.2 THEN"Yes"

ELSE

"NO"

END
```

- The resulting view should look similar to the following screenshot. Make sure to use the same principles of color and arrangement for the tooltips in each view so that important information is clear:

Fig. 5.17.4 : Map view

- Finally, answer the last question in the hierarchy on abnormal precipitation values by showing daily summed values for each station. The unusual values can be found easily in a daily bar graph by station. The resulting visual should look similar to the following screenshot:

Fig. 5.17.5 : A daily trend by a station view

5.17.5 Assembling views into a Dashboard

- The next step is to create a new dashboard tab. Set the size to **Generic Desktop (1366 x 768)**.Add the four views in the priority order discussed previously in the chapter.
- The initial dashboard should look like the following screenshot:

Fig. 5.17.6 : The initial dashboard

- With Tableau, you can add dashboard actions that allow a user to filter other views based on what they click in a source view. Following the question hierarchy, a click on a monthlystation count bar in the monthly trend view should filter the map view.
- A selection of stations on the map view should filter the daily trend view.
- Go to dashboards in the menu bar and select **Actions...** to add a couple of actions to the dashboard view. Add a filter from the the monthly trend view to the map view linked by the **MONTH([Measurement date])** field when a bar is selected. Do the same for the **Map** view targeting the daily trend view linked by the **Station** field:

Fig. 5.17.7 : Add dashboard actions

- Now, a user of the dashboard can investigate a measurement month to quickly understand which weather stations are reporting data.
- They can also view one station or groups of stations by the daily summed precipitation values to identify unusual results.
- The example of following screenshot shows the mapped location for November 2013 reporting stations and the daily trend for a few stations in the Denver area:

Fig. 5.17.8 : Example of dashboard interaction

5.18 CREATING AND VISUALIZING ALERTS

- IoT data is inherently noisy. There are often cases of invalid and missing values. This requires constant vigilance to identify and correct data issues when they occur. The correction could then be handled in the transformation of the raw data or in the softwareand design of the device.
- Either way, the faster an issue is detected, the quicker it can be resolved. For IoT data,consider bad data as lost money that can rarely be recovered. Minimize the loss by identifying and correcting issues quickly.
- Dashboards can also be created for this purpose by following the same process.

5.18.1 Alert Principles

There are some principles to follow when designing an alert system, even a simple one that will be part of a dashboard :

- **Balance alert sensitivity to minimize false positives:** People will learn quickly toignore alerts if they rarely identify an actual problem. This has to be balanced against missing too many real issues though. A cost benefit calculation can be used to help in this decision.

  [cost of investigating a problem] * [number of alerts] < [cost of actual problem] *[probability of true positive]
- The left side of that equation should be noticeably less than the right side.

- **Be wary of alert fatigue :** A long list of continuous alerts is daunting, and human brains tend to become conditioned over time to recognize it as a normal situation. Alerts will start to become background noise and will not get a response. Anyonewith an Android smartphone knows how this feels - too many alerts are exhausting. There does not need to be an alert for every problem. Keep it to a manageable amount for the big ones.
- **Make alerts like a to-do list :** A user should not have to search for where the problem is; it should be listed out for them if there are any. If there are no issues discovered, a blank list can be comforting - like leaving work an hour early!
- **Incorporate a tracking system for alert responses:** More frustrating than spending a couple of hours investigating a problem someone else has already discovered and corrected.

5.18.2 Organizing Alerts using a Tableau Dashboard

- Example to highlight how to use a dashboard for alerting purposes. Tableau server has an email functionality that a user can subscribe to that will email an image of the dashboard at a regular interval with an embedded link back to the interactive dashboard. This can then be read and digested like a morning newspaper.
- It is learned while investigating the 15 minute precipitation dataset that there were measurement values that represented a state instead of an actual reading. In these cases, the value was either **-9999** or **999.99**.
- What we would want to see, if we were relying on this data on a daily basis, is that a station reports valid precipitation measurements and little to no non-measurement values. We can organize a simple question hierarchy to represent our thinking process:

Fig. 5.18.1 : Alert question hierarchy for stations reporting invalid precipitation data

- We can create an alert dashboard to monitor this on a daily basis. This will allow quick identification of a problem. We will start by creating four additional calculated fields in the following order:

Name: Precip Accumul Value ValidityCalculation:

```
CASE [Qgag]
WHEN -9999 THEN
"Invalid"
WHEN 999.990 THEN
"Invalid"
ELSE
```

```
IF [Qgag] >= 0 AND [Qgag] <20 THEN
"Valid"
ELSE
"Invalid"
END
END
```

Name: Precip Value Validity Calculation:

```
CASE [Amount of Precipitation (inches)]
WHEN -9999 THEN
"Invalid"
WHEN 999.99 THEN
"Invalid"
ELSE
IF [Amount of Precipitation (inches)] >= 0 AND [Amount of Precipitation (inches)]
<20 THEN
"Valid"
ELSE
"Invalid"
END
END
```

Name: Invalid Count Calculation:

```
IF [Precip Accumul Value Validity] = "Invalid" AND [Precip Value Validity]
= "Invalid" THEN1
ELSE
0
END
```

Name: Valid Count Calculation:

```
If [Invalid Count] = 0 THEN1
ELSE
0
END
```

- As a little bonus, here is a trick in Tableau to create a dynamic bar chart within a tooltip. You can use the ASCII character for a fixed in square to create what looks like a dynamic length bar. Create the following calculated field and drag it to the **Tooltip** box in the **Markscard**:
- Name: Valid Count BarCalculation:
- LEFT("[redacted]", SUM([Valid Count]))
- Now, arrange the calculated fields into a view that looks like the following screenshot. Make sure to edit the tooltip card to arrange information in the appropriate order. You canmove the bar chart formula to be after the valid count field. Add a measurement date filter,so the list is for a single day only:

Fig. 5.18.2 : Alert view

Now, you have an alert view to put into a dashboard. If this were a live feed, you can change the date filter to be a relative date and set it to yesterday. Then, in Tableauserver, set up the users to receive daily emails of the alert view. Now you have a list of stations that need reviews sent to their inbox every morning.

5.18.3 Types of Dashboards

There are several ways to customize the dashboard, and they all fall into one of three categories - **iDashboards**:

1. **Operational Dashboards:** these dashboards show the real-time performance of day-to-day business operations. They are connected to multiple data sources and contain hundreds of metrics, indicating various functionalities of the business.

2. **Analytical Dashboards:** these dashboards use historical data to identify trends. They are mainly used by data analysts to write detailed reports about a company's past performance and what steps they can use to improve current systems.
3. **Strategic Dashboards:** these dashboards are mainly used to track current performance compared to key performance indicators and align actions with strategy.

5.18.4 Use Cases of Dashboards

In data science, dashboards are used for machine learning operations, data streaming, database management, and monitoring applications in production. However, in a typical business, the dashboard has a broader use case:

- Customer metrics
- Financial information
- Sales information
- DevOps
- Web analytics
- Manufacturing information
- Human resources data
- Marketing performance
- Logistics information

5.18.5 Summary to Create a Data Dashboard

Before creating a Tableau dashboard, It is necessary to understand a few rules. These rules will help you build an effective dashboard that will fulfill your target audience's demands and goals.

1. Know your audience and understand what they need.
2. Choose relevant and relatively clean data that will meet your goals.
3. Make sure your data is correct and clean.
4. Select the best visualizations that represent your data.
5. If you are developing a dashboard for the first time, choose a template that reduces complexity and saves time.
6. Do not clutter your dashboard space with too many plots. Use simple colors and allow for space in your design to showcase the most significant parts of the data.
7. Ask for feedback from the audience and use it to improve on the current design. Iterate and improve.

5.19 EXAMPLE : TABLEUS DASHBOARD

A simple Tableau dashboard consists of three elements:

- objects,
- visualization, and
- filters.

Fig. 5.19.1 : Tableus DashBoard

5.19.1 Connection to Data Source

- Create a simple Tableau dashboard and publish it on a public server. The primary purpose of this tutorial is to learn the basics of creating and customizing data visualization, adding objects and filters, connecting filters, and sharing the finalized version of the dashboard with the wider public.
- Before we connect the data source, we need to download **Power consumption in India(2019-2020)** from the Kaggle platform and then extract the data.

Fig. 5.19.2 : Power consumption in India(2019-2020) from the Kaggle platform

- To connect the data source, we need to launch Tableau Desktop Public Edition (2022.1.1). On the main screen, it will show the multiple options to load a file or connect to a database server. Tableau supports all kinds of data connections; to access unavailable connections, download the specific **drivers**.
- For example, a MySQL connection is not available by default. We can install the drivers for the database and then connect to the SQL server. Take the **Connecting Data in Tableau** course to better understand connecting and managing various data sources.

- In our case, the data source is a CSV file, and it can be accessed by clicking on the "Text file" tab on the left panel. Next, select the CSV file (long_data_.csv) and press the open button.

Fig. 5.19.3 : Upload Dataset

- The dataset has been successfully loaded and you can explore the columns and attributes before jumping into creating a visualization.
- As we can observe, Tableau has automatically assigned data types to each column based on properties. This dataset consists of:
  - States (String): 33 states
  - Regions (String): 5 regions
  - Latitude (Geography): Geolocation coordinates
  - Longitude (Geography): Geolocation coordinates
  - Dates (Date & Time): Jan 2, 2019 to May 23, 2020
  - Usage (Decimal Numbers): Power consumption in MegaWatts

Fig. 5.19.4 : DataSet details

5.19.2 Creating Data Visualization Sheets

- Create the first bar chart by switching tabs from the "Data Source" to the "Sheet 1" (orange highlighted tab on the bottom left corner).
- Sheet 1 has multiple sections, tabs, and buttons. To make things simple, we will ignore all the sections and buttons and focus on the "Tables" section on the top left.
- The table consists of the data fields (columns) separated by the data types: Discrete and Continuous.

Fig. 5.19.5 : DataField

- To create a bar chart of total power consumption per state, simply drag "States" to Columns and "Usage" to Rows.
- Tableau will automatically assemble a bar chart with labels and values, as shown below.

Fig. 5.19.6 : Bar Chart in Tableau

- To change the color of bars to green, click on the "Color" button under the Marks section and then select the color green.
- We can customize the title by "double-clicking" on it and making changes to the font and color to match the theme.

Fig. 5.19.7 : Change the color of bars

- Before we create another visualization, we need to understand the bottom panel. We can create multiple Worksheets, Dashboards, and Stories by clicking on buttons with a plus (+) sign, as shown below.
- Similar to Google Chrome tabs, we can switch between Data Source, Worksheets, Dashboards, and Stories tabs by clicking on them.
- In our case, we will create a second worksheet by clicking on the "New Worksheet" button.

Fig.5.19.8 : Creating Worksheet

- On the second worksheet, we will plot "power usage per state" on a map. First, drag and drop Longitude to Columns and Latitude to Rows.
- Second, change both fields to "Dimension" by clicking on the arrow and selecting the "Dimension" option.
- Third, double click on the Usage and States data fields, and they will automatically appear in the "Marks" section. You can also do this manually by dragging and dropping them into the "Marks" section and changing them to "Size" and "Detail" marks.
- Finally, click on the "Show Me" button and select symbol maps. The "Show Me" button gives you recommended visualizations based on your inputs. You can switch from one chart to another by clicking on the available options.

Fig. 5.19.9 : Customizing Map

- After customizing the map and title, the final version of the map looks clean. On the top right, you can see filters based on the size of the circle.
- The circle represents the states in India, and the size of the circle is based on power usage.
- You can zoom in and interact with the map by clicking on any red circle and reading tooltips (Coordinates, States, Usage).

Fig. 5.19.10 : Plot of Total Power Consumption

- In the subsequent worksheet, we are going to plot a historical time series line chart with predictions. Move "Usage" to Rows and "Dates" to Columns. Then, change the "Dates" from "Year" to "Month", as shown below.
- Finally, we are going to plot the forecast for the next 11 months by right-clicking on the plot and selecting Forecast > Snow Forecast. Tableau will automatically plot forecasts with a 90% confidence highlighted area.

Fig. 5.19.11 : Plot the forecast

- After setting up the title and colors, we can see two distinct graphs:
- Actual: Due to the Covid19 lockdown, the power consumption dropped because the majority of industries were closed.
- Estimated: The forecast chart is a constant line, and it is not reliable due to missing values in historical data. The 90% confidence area is also too wide for us to take this prediction seriously.

Fig. 5.19.12 : Power Usage History and Forecast

- For the fourth worksheet, we will be creating a Region versus Usage BoxPlot. We can do this by clicking on "box-and-whisker plots" in the "Show Me" section, as shown below. After customizing the visualization, all four sheets are ready to be added to a dashboard.

Fig. 5.19.13 : Creating a Region versus Usage BoxPlot

5.19.3 Building a Dashboard

- To create a dashboard, we need to click on the "New Dashboard" button at the bottom. The button has a box shape with a plus sign. The Dashboard window has a dashboard panel on the left side that consists of three sections:
  1. **Canvas :** to set the size of the dashboard based on various devices (Mobile, Tablet, etc).
  2. **Sheets :** for adding and removing worksheets on canvas.
  3. **Objects :** for adding text, images, extensions, webpage, and buttons to canvas.
- Now, we will double-click on all sheets one by one to add them to the dashboard canvas with related filters. We can adjust the locations of visualizations and filters and adjust the size to give it a clean look.

Fig. 5.19.14 : Building a Dashboard

Fig. 5.19.15 : Adjust the locations and size

5.19.4 Adding Filters

Next, we are going to add relevant filters and remove the previous filter.

1. Remove the "Usage" size filter by clicking on X.
2. Select the map visualization, and click on the down arrow to select Filters > Sum of Usage.
3. The filter will appear in the top right corner.
4. Move the filter towards the right, so that it looks like part of the dashboard.

Fig. 5.19.16 : Adding Filters

Similarly, we are going to add filters for "States" and "Regions".

1. Select the Boxplot visualization and add the "States" filter.
2. Go to the filter and click on the down arrow to access the options and select Multiple Values (dropdown). It will change the filter from a list to a dropdown.
3. Move the filter so that it shows in the middle of the "Forecast indicator" and "Usage". Tableau will adjust the filter automatically if you pick and hover over the filter in the center.
4. In the same way, add a "Regions" filter and adjust it between the "States and "Usage" filters.

Fig. 5.19.17 : Add filters for "States" and "Regions

5.19.5 Adding Objects

- To add a title object by dragging the "Text" and placing it at the top of the dashboard canvas. A window prompt will then ask us to type the text. In our case, we will write the title of the dashboard.

Fig. 5.19.18 : Adding Object

- Next, we are going to add an Image object to the top left section and then select the Indian national flag JPEG file.

Fig. 5.19.19 : Add an Image object

- After adjusting the position and size of the flag and title, our dashboard is almost complete. Now, we just need to connect filters and visualization so that they are in sync.

5.19.6 Connecting Filters

- The filters that we have just added are unconnected and only affect a single visualization.
- For example, if you change Regions from "(All)" to "ER", the filter will only be applied to the Boxplot, as shown below.

Fig. 5.19.20 : Connecting Filter

- To make the "Regions" filter global, we will change the option from "Only This Worksheet" to "All Using This Data Source".
- You can access this option by selecting the filter and clicking the down arrow > Apply to Worksheets > All Using This Data Source.
- We are also going to connect the "States" filter so that if we make changes, they will be applied to all four worksheets.
- To increase interactivity, we can also use the worksheet visualization as a filter.
- We can achieve this by selecting the worksheet and clicking on the funnel button "Use as Filter".
- Repeat this step for all visualizations, and the dashboard will become dynamic and fully connected.

Fig. 5.19.21 : Connect filters and visualization

Fig. 5.19.22 : Use as Filter

Now, if we change the Regions to "ER", the filter is applied to all four workspace visualizations. Our dashboard is now complete.

Fig. 5.19.23 : Complete Dashboard

5.19.7 Publishing

- To publish the dashboard and share it with a broader audience, we can simply click on File > Save to Tableau Public and then write a workbook title to save it.
- Uploading the changes to the public server takes a few seconds. We will then be redirected to an online version of the dashboard. Make sure you are logged into your public account.

Fig. 5.19.24 : Publishing a Dashboard to Tableau Public

Chapter Ends...

❑❑❑

MODULE 6

IoT Application Design

Syllabus

Prototyping for IoT and M2M, Case study related to : Home Automation (Smart lighting, Home intrusion detection), Cities (Smart Parking), Environment (Weather monitoring, weather reporting Bot, Air pollution monitoring, Forest fire detection, Agriculture (Smart irrigation), Smart Library. Introduction to I-IoT, Use cases of the I-IoT, IoT and I-IoT – similarities and differences, Introduction to Internet of Behavior (IoB).

Self-learning Topics: Internet of Behaviors (IoB) and its role in customer services.

6.1 Prototyping For Iot And M2M 6-2
6.1.1 Difference between M2M and IoT 6-2
6.2 Case Studies 6-3
6.2.1 Home Automation 6-3
6.2.2 Home Automation - Smart Lighting 6-4
6.2.3 Home Automation - Home Intrusion Detection 6-5
6.2.4 Smart Cities - Smart Parking 6-6
6.2.4.1 Smart Parking 6-7
6.2.5 Smart Environment-Monitoring 6-9
6.2.5.1 Weather Monitoring System 6-10
6.2.5.2 Weather Reporting Bot 6-11
6.2.5.3 Air Pollution Monitoring 6-11
6.2.5.4 Forest Fire Detection 6-13
6.2.6 Smart Agriculture 6-14
6.2.6.1 Smart Irrigation 6-16
6.2.7 Smart Library 6-17
6.3 Introduction To IIoT 6-18
6.4 Use Cases Of The I-IoT 6-21
6.5 IoT and IIoT - Similarities And Differences 6-22
6.5.1 Similarities between IIoT and IoT 6-22
6.5.2 Differences Between IIoT and IoT 6-23
6.6 Internet of Behaviour (IOB) 6-23
6.7 Self-Learning Topics 6-24
6.7.1 Internet of Behaviours (IoB) and its role in customer services 6-24
6.8 Case studies based in IoB 6-26
- **Chapter End** 6-26

6.1 PROTOTYPING FOR IOT AND M2M

- M2M (Machine-to-Machine communication) supports communication between two or more wireless devices.
- M2M communication is free from any human intervention.
- M2M is designed for cross-platform integration.
- M2M, means : two machines "communicating," or exchanging data, without human interfacing or interaction.
- Devices could be sensors, actuators, embedded systems or other connected elements.
- M2M technology can be used in our homes, offices, shopping malls and other places.
- Controlling electrical appliances RF or Bluetooth from your smartphone is a simple example of M2M applications at home.
- The Internet of Things (IoT) is the network of physical devices embedded with sensors, software and electronics.
- These devices can communicate with each other and exchange data over a computer network.
- The things in the IoT refer to hardware devices which are having an unique id through a network platform within the Internet infrastructure.

(1F1)Fig. 6.1.1 : M2M and IOT

- M2M and the IoT are two of the technologies that form the basis of the new world.
- The physical entities such as buildings, farmland, etc. and natural resources like air.
- M2M is about machines, smartphones and appliances,
- whereas the IoT is about sensors, cyber-based physical systems, Internet and so on

6.1.1 Difference between M2M and IoT

Sr. No.	Parameter	M2M	IoT
1.	Communication	machine to machine communication	communication using cloud
2.	internet connection	not required	required

Sr. No.	Parameter	M2M	IoT
3.	technology	mostly hardware based	hardware and software based
4.	scalable	less	more
5.	used for	B2B	B2B and B2C
6.	Sharing of collected data	Data collected is not shared with other applications	Data is shared with other applications (like weather forecasts, social media etc.) improve end user experience
7.	Example	Remote monitoring, fleet control	Smart Cities, smart agriculture etc.
8.	Open APIs	Not supported	Supported

6.2 CASE STUDIES

6.2.1 Home Automation

Fig. 6.2.1 : Home Automation

- A smart home automation system enables individuals to control electric and electronic appliances like - TV, Air Conditioners, washing Machines, CCTVs, etc. smartly and automatically within a home environment, from anywhere in the world.

- As shown in Fig. 6.2.1, all the appliances of the home are connected to the internet through the gateway.
- User's smartphone or laptop is also connected to the internet using the mobile communication networks.
- Using the app installed in the smartphone, users can control the devices which are present in the home, remotely from anywhere in the world.

6.2.2 Home Automation - Smart Lighting

- Smart lighting uses IoT-enabled sensors, bulbs, & adapters to manage the home or office lighting with a user's smartphone or smart home management platform.
- Smart lighting solutions can be controlled through a smartphone or smart assistant, set to operate on a schedule, or triggered by sound or motion.
- Smart lighting systems automatically detect whether a building is occupied or not, using CCTV Cameras or an IoT-enabled thermostat or other security systems,
- If no one is present in the office or in the room, the smart lightning system will automatically power off the appliances

How it Works

- As shown in Fig. 6.2.2, Smart lightning devices like - Smartphone, Tablet, Laptop etc. are WiFi-enabled that can be controlled by a smart assistant or mobile app, using the internet.
- Light switches operate as an adapter to control groups of lights.

(1F2)Fig. 6.2.2 : Smart Lighting

Key Benefits of IoT-Enabled Smart Lighting

- Save money and energy.
- According to external whether ,Adjust the colour or dimness of lights in different rooms
- Sensors used in Smart Lighting System
  - Photosensitive Sensor
  - Ultrasonic Sensor
  - Voice Sensor
  - Illuminous Sensor
  - Infrared Sensor
  - Temperature Sensor
  - Microwave Sensor

- Wireless communication protocols used are –
  - 802.15.4 ○ ZigBee ○ WiFi ○ 868 ○ 900 ○ 4G ○ LoRaWAN ○ Sigfox

6.2.3 Home Automation - Home Intrusion Detection

- A smart home automation system controls lighting, temperature, security camera systems, and appliances.
- These devices and sensors are connected to the internet through the gateways.
- Fig. 6.2.3 Shows the smart home intrusion detection system, in which the security cameras are connected to the smart lightning, smart devices like - laptop, smartphones, smart cupboards, telephone and to different alarms.

(1F3)**Fig. 6.2.3 : Home Intrusion Detection System**

- When an intruder tries to enter the home, he will be detected by smart cameras and this information will be sent to all the connected devices by means of alarms, blowing the lightning system, sending messages on the smartphones and so on.
- At the same time, this information will be sent to the nearest police station through the internet and within some time police will also come to the users home to catch the intruder.
- **Sensors used in Intrusion Detection System**
  - Passive Infrared Sensor :
  - Detects the Infrared and does not emit any.
  - The sensor remembers the infrared image of its surroundings and notices the energy changes occured due to motion.
  - These changes in the surrounding energy trigger the sensor and the sensor sends the warning to the Control Panel.

Fig. 6.2.4 : Passive Infrared Sensor

- **Microwave Detector**
  - work on the principle of emitting electromagnetic radiations and analysing the waves which are reflected back to the receiver.

- According to the movement in the space, the reflected waves get altered and the receiver can identify the changes.
- This alteration in the path triggers the sensor and sends the warning signal to the Control Panel.

Photoelectric Beam Sensor

- Consists of a combination of light, an emitter and a receiver.
- Emitter emits the light which is then received by the receiver set in a straight line.
- Incase of any interruption in the path of light, the receiver detects it and converts the received amount of light to an electrical output which will be sent as the warning signal to the Control Panel.

LASER (LiDAR)

- Emits the narrow beam of light produced by the result of Optical Amplification.
- LASER emits the light in a straight line that travels into the air as a medium of transmission and hits the object in front of it.
- If someone comes into its path, the pattern of the beam gets distorted and the device takes it as an attempt of intrusion and hence sends the alert signal to the Control Panel.

Magnetic Switches

- Used for doors and windows to detect forceful entry.
- It consists of two magnetic blades, one fixed to the door or window and another one attached to the frame. When in a line with each other.
- Both magnets allow the electricity to flow and form a complete circuit.
- If the door or window is pulled, the gap between the magnets increases and the circuit gets broken.
- This sends the signal of panic to the Control Panel in reaction.

- Wireless communication protocols used

o 802.15.4 o ZigBee o WiFi o 868 o 900 o 4G o LoRaWAN o Sigfox

6.2.4 Smart Cities - Smart Parking

- Smart city applications connect people, processes, data and things.
- A smart city integrates multiple IoT solutions to manage a city's daily working of different departments like - information systems, schools, libraries, transportation systems, hospitals, power plants, water supply networks, waste management, law enforcement and other community services.
- Smart city technology is also used in the government services, transport and traffic management, energy, health care, water, innovative urban agriculture and waste management.
- Smart-city solutions can include the following services

- Smart parking spaces
- Smart street lightings
- Smart traffic solutions
- smart parking
- smart waste bins
- security and surveillance
- Smart water management, for monitoring and optimising a city's water and sewage services
- Smart health services
- Smart structures (building, bridges and historical monuments)

6.2.4.1 Smart Parking

- Vehicular traffic congestion and parking spaces is a growing problem in cities.
- Nowadays, there are a number of multilevel parking spaces provided in the cities.
- A smart parking-service of the city should enable the following:
  - Guide the drivers for the available parking slots and spaces
  - a mobile app helps a driver to obtain the appropriate parking-slot information remotely.
  - This information includes - location of the parking utility, its cost and reservation facility.
  - Information of available and unavailable slots is provided by sending messages in real time.
  - This is done by the edge sensors and devices, which accurately senses the slots available for occupancy of vehicles in real time.
  - Provides display boards at road traffic junctions for status of availability
- Different kinds of parking sensors are available in the market like
- **In-ground magnetic sensors** - creates a magnetic detection field in a parking spot
- **Video-based sensors** - detects events based on video computing (vehicle movements or presence); and **radar sensors** that sense the presence of vehicles (volumetric detection).

(IF4)**Fig. 6.2.5 : Smart Parking**

- Fig. 6.2.5, shows the smart parking scenario Sensor devices, gateways, parking management system, real time parking guidance and a parking lot.
- When a car comes in a parking lot, it is wirelessly connected with the Real Time Parking Guidance System and Parking Management System, through the Gateways available in the parking lot
- If there is empty parking space available in the parking lot, the sensor located at that free space will inform to the gateway
- And this information is given to the Car Driver through the wireless network and Real Time Parking Guidance System and Parking Management System
- **Advantages of smart parking**
  - Reduces parking search time.
  - Less traffic congestion.
  - Reduces pollution
  - Reduces fuel consumption and costs.
- **Sensors used in Smart Parking System are -**

Magnetometer

- A magnetometer is a digital compass used to measure field direction and magnitude
- A ferrous (containing iron) object, like a vehicle, creates a short-range distortion of this field which can be measured.
- As shown in Fig. 6.2.6(a) , the deviations are much stronger around the engine and the wheels of a car.

Fig. 6.2.6(a) : Magnetometer

Ultrasonic ranging sensor

- This sensor uses the same time-of-flight method as IR
- The emitter sends ultrasound waves in (40-60KHz range).

- It is used in parking assistance systems in modern cars: the sensors are affixed in rear and front bumpers and help you park by signalling proximity to obstructions in the short range.

Fig. 6.2.6(b) : Ultrasonic ranging sensor

- Wireless communication protocols used in smart parking system are
  - 802.15.4
  - ZigBee
  - WiFi
  - 868
  - 900
  - 4G
  - LoRaWAN
  - Sigfox

6.2.5 Smart Environment-Monitoring

- Environment monitoring specifies the actions required to monitor the quality of the environment.
- A smart environment monitoring system include - Study of current status of the environment by monitoring different environmental parameters
- like - the air, soil and water quality, temperature, humidity
- Also monitoring harmful chemicals, biological, microbiological, radiological and other parameters
- Using different wireless sensor nodes or devices
- This information collected by the sensors, will be given to the central database and further it will be processed and the final result will be given to the intended users of the system.
- Following sensors can be used in - smart environment monitoring system:
  - Temperature, humidity and pressure
  - Carbon Monoxide (CO) for low concentrations
  - Carbon Dioxide (CO_2)
  - Oxygen (O_2)
  - Ozone (O_3)
  - Nitric Oxide (NO)
  - Nitric Dioxide high accuracy (NO_2)
  - Sulfur Dioxide high accuracy (SO_2)
  - Ammonia (NH_3)
  - Methane (CH_4) and Combustible Gas
  - Hydrogen Sulfide (H_2S)
  - Particle Matter (PM1 / PM2.5 / PM10) – Dust 1

- Following wireless communication protocols can be used to connect the devices :
  - 802.15.4
  - ZigBee
  - WiFi
  - 868
  - 900
  - 4G
  - LoRaWAN
  - Sigfox

6.2.5.1 Weather Monitoring System

- A smart weather monitoring system contains - weather measuring wireless sensor nodes with assigned an ID.
- Each node measures the T, RH and other weather parameters at assigned locations.
- All nodes will communicate with each other through the wireless network
- Each network has an access point, which receives the messages from each node.

(1F5)Fig. 6.2.7 : Weather Monitoring System

- There are interconnections between nodes, coordinators, routers and access points and APIs.
- Each access point associates a gateway and Forward and store the parameters on an Internet cloud platform
- Publishes weather messages for the display boards at specific locations in the city and communicates to weather API at mobile and web users
- Publishes the messages in real time and send alerts using a weather reporting application
- Sensors used in Weather Monitoring System are -
  - Temperature sensor.
  - Humidity / hygrometer sensor.
  - Soil moisture sensor.
  - Rain sensor
  - Barometric sensor – for measuring atmospheric pressure.
  - Anemometer – for measuring wind speed.
  - Visibility sensor – for measuring visibility during snow, rain, storm etc.

- Following wireless communication protocols can be used to connect the devices :
  - 802.15.4
  - ZigBee
  - WiFi
  - 868
  - 900
  - 4G
  - LoRaWAN
  - Sigfox

6.2.5.2 Weather Reporting Bot

- A bot is an application that runs automated or semi-automated scripts for a specific set of tasks and communicates the results over the Internet.
- A bot generally performs tasks which are simple and structurally repetitive
- A bot can communicate with an API using Instant Messaging (IM) like - Twitter or Facebook.
- A bot can chat and give responses to the questions asked by the user.
- In the Bot application, the script fetches, analyses and files information from a web server.
- A weather bot is multitasking. It communicates a report on a mobile.
- The bot fetches, analyses and communicates information to a report seeking API.
- To detect the weather parameters like - air temperature, atmospheric (barometric) pressure, humidity, precipitation and solar radiation and wind, different sensors are used.
- These sensors send their data to the gateways. From gateways, this data will be given to the cloud.
- The bot uses the weather parameters and generates the alert messages from the database and messages for forecast by a cloud analytics service.

6.2.5.3 Air Pollution Monitoring

- Air pollution is one of the biggest problems all over the world..
- Reasons for air pollution are - from cars, toxic gases generated in factories and farms, such as carbon monoxide (CO), from chemical plants.
- Air pollution monitoring can be done by measuring - levels of CO, carbon dioxide (CO_2) and ozone (O_3) and hydrogen sulphide (H_2S).
- Also, measuring levels of hydrocarbons, such as ethanol, propane.
- As shown in Figs. 6.2.8 & Fig. 6.2.9, Sensor nodes placed at different locations will collect readings of these parameters from the air on the hourly basis and send this data to the gateways.
- These gateways will further send this data to the cloud and from the cloud the processed information will be given to the end user on their devices like smartphones.

(1F6)**Fig. 6.2.8 : Air Pollution Monitoring**

(1F7)**Fig. 6.2.9 : Air Pollution Monitoring**

- **Sensors used in Air Pollution Monitoring System are :**

PM Sensors

  - Detects particular matter having diameters ranging between 2.5 and 10 µm (called PM10) and less than 2.5µm (called PM2.5).
  - They are widely used by hobbyists.

Ozone (O3) sensor

- Monitor ozone levels.
- Breathing ozone can trigger a variety of health issues such as chest pain, throat irritation, coughing etc.

Lead (Pb) Sensor

- Pb can damage the nervous system and consecutively results in IQ loss and negative impacts on learning of the childrens.
- It causes cardiovascular and renal effects as well as anaemia in adults.
- Pb sensors help monitor Pb levels.

Sulfur Dioxide (SO2) Sensor

- The SO2 contributes to acidification of soil and surface water.
- It causes damage to vegetation.
- SO2 sensor monitors SO2 levels.

Nitrogen Dioxide (NO2) sensor

- It monitors NO2 level.

Carbon Monoxide (CO) sensor

- It monitors CO level.
- The CO reduces the amount of oxygen reaching organs and tissues of human beings. Moreover it contributes to the formation of CO2 and ozone.

Volatile Organic Compounds sensor

- Monitor toxic air pollutants which can cause cancer and/or other serious health issues.

Mercury (Hg) sensor

- Monitor Hg levels.
- Hg is deposited onto soil and into rivers, lakes and oceans.
- Other toxic air pollutants measuring sensors such as Benzene.

- **Following wireless communication protocols can be used to connect the devices**

802.15.4	ZigBee	WiFi	868
900	4G	LoRaWAN	Sigfox

6.2.5.4 Forest Fire Detection

- A big problem for countries with large forest areas is forest fires.
- A fire monitoring service using IoT, does the following tasks:
  - The fire in the forest causes an increase in temperature, increase in the level of Carbon Monoxide (CO), increase in the level of Carbon Dioxide (CO2) and increase in the infrared light (fire generated) intensity -

- To measure and monitor the changes in above entities, different sensors are used like - temperature sensor, gas sensor, smoke sensor etc.

(1F8)**Fig. 6.2.10 : Forest Fire Detection System**

- As shown in Fig. 6.2.10, when the fire is started in the forest, each sensor node will sense an increase in the temperature and the intensity of the gases generated due to the fire.
- This captured data is continuously sent to the gateways, through the communication network.
- From gateways this data will be given to the cloud using the internet.
- The cloud system will inform the intended users of the system, on their own devices by giving an alarm or by other means.

- Sensors used in Forest Fire Detection System are :
  - CO2 detector
  - Temperature Sensor
  - Temperature Sensor
  - X
  - Gas Sensor
  - Light Sensor

- Wireless communication protocols used
  - 802.15.4
  - ZigBee
  - WiFi
  - 868
  - 900
  - 4G
  - LoRaWAN
  - Sigfox

6.2.6 Smart Agriculture

- In IoT-based smart farming, the crop field is monitored with the help of sensors, robotics, automation vehicles, control systems, automated hardware, variable rate technology, and so on.
- These sensors track every essential for crop production - like soil moisture, humidity, light, temperature, etc., and automates the irrigation system.

- With this system, farmers can monitor the field conditions from anywhere.
- Smart Agriculture System contains many subsystems like - smart irrigation, smart crop monitoring, smart disease monitoring system etc.
- Benefits of Smart Agriculture :
  - Higher crop yield
  - Better quality
  - Understand which factors govern crop growth and yields

(1F9)**Fig. 6.2.11 : Smart Agriculture**

  - Guaranteeing food security
  - Less transport costs: human interventions only when needed.
  - Less time spent
  - Reduce crop losses through disease or adverse weather
  - Cost savings reducing use of fertilisers, pesticides and consumables
  - Fight against droughts, scarcity and famine
- Sensor used in Smart Agriculture
  - Soil moisture (3 depths)
  - Leaf wetness
  - Atmospheric pressure
  - Anemometer
  - Air temperature
  - Luminosity (Luxes Accuracy) for Smart Lighting
  - Soil temperature
  - Solar radiation (PAR and UV)
  - Stem, truck and fruit diameter
  - Wind vane
  - Air humidity
  - Ultrasound (distance measurement)

- pressure sensors
- Soil morphology
- Solar radiation
- Weather station
- Leaf humidity
- Pluviometer

- Wireless communication protocols used
  - 802.15.4
  - ZigBee
  - WiFi
  - 868
  - 900
  - 4G
  - LoRaWAN
  - Sigfox

6.2.6.1 Smart Irrigation

- Uses sensors for measuring soil moisture and actuators for watering channels.
- Each soil moisture sensor is installed at a certain depth in the fields.
- An array of actuators (solenoid valves) are placed along the water channels to control deficiencies in moisture levels above thresholds during a given crop period.
- Each sensor board is in a waterproof cover and communicates to an access point using ZigBee protocol.
- If the soil moisture detects less moisture than the minimum threshold level, then the signal will be given to actuators of the watering channel, for providing water to the crop.
- After enough moisture level required for that crop is achieved, the soil moisture sensor will again give a signal to actuators of watering channels, to stop the water.
- In this way, each plant will automatically get the required amount of water only and at the required time.

Fig. 6.2.12 : Smart Irrigation System

- As shown in Fig. 6.2.12, from sensor nodes, farmers will receive the data about the current condition of the crop and water requirements.
- Access point also receives this data and transfers it to an associated gateway.
- Gateways communicates this data to a cloud platform
- Analytics at the cloud platform analyses the moisture data and communicates to the actuators of water irrigation channels
- An algorithm uploads and updates the programs for the gateways and nodes.
- **Sensors Used in Smart Irrigation are :**
- Soil Moisture Sensor • Wind Sensor • Rain Sensor • Water Pressure Sensor
- **Wireless communication protocols used in Smart Irrigation are :**

6.2.7 Smart Library

- 802.15.4
- ZigBee
- WiFi
- 868
- 900
- 4G
- LoRaWAN
- Sigfox

- As shown in Fig. 6.2.13, the smart library system contains a tagging area, where the new books will be registered and entered in the library and attached with the RFID labels.
- The user will have the unique RFID tag with him, when he comes to the library for issuing or returning the books
- A unique special tag is attached to the book-shelves.
- When the user enters the library through the electronic access gate control, the RFID reader receives the signal of a tag passed on the book and the ID-card.

(1F10)**Fig. 6.2.13 : Smart Library**

- In this scenario, if the user (student) entersv in the central area, the IoT system will detect the registered user and the book, and send this information to the central cloud system to analyze and display the user in the monitoring system.
- It also makes a previous registration of the borrowed book into its database.
- When the user is going out from the protected area of the library, a detection system checks whether the object (book) was registered with the checkout system of the library.
- The process of check-in and check-out can be implemented automatically. The user will scan his ID card. The system will automatically verify its database from the servers.
- If the customer is a valid user, the book can be scanned by the reader and registered in the system. If the book is getting back to the library, the message will be sent to the user, who has given the demand of that book earlier.
- **Sensors Used in Smart Irrigation are :**
  - RFID Reader
  - RFID tags
  - Motion Detector Sensor
  - Camera
- **Wireless communication protocols used in Smart Irrigation are -**
  - 802.15.4
  - ZigBee
  - WiFi
  - 868
  - 900
  - 4G
  - LoRaWAN
  - Sigfox

6.3 INTRODUCTION TO IIOT

- Industrial Internet of Things (IIoT) is a branch of Internet of Things (IoT).
- IIoT is used in manufacturing and other industrial processes to improve the working, increase life of the machine.
- First Industrial Revolution

  After the invention of the steam engine in 1760, steam was used to power everything from agriculture to textile manufacturing.
- Second Industrial Revolution

  At the end of the 19th century electricity was came into picture, which started the mass production
- Third Industrial Revolution

  In the second half of the 20th century, Semiconductors and electronic controllers were developed and the automation era started.

Fourth Industrial Revolution

- In 2011, Henning Kagermann, Wolf-Dieter Lukas, and Wolfgang Wahlster introduced the term "Industry 4.0" in the manufacturing system using the capabilities of the latest digital technologies

- **The Industrial Internet of Things (IIoT)** is made up of the Internet, Artificial Intelligence based predictive analytics, automated industrial robotics, cloud computing, data collection and analytics, and high performance computing.

(1F11)**Fig. 6.3.1 : First to Fourth Industrial Revolution**

- In IIoT, the data is collected from all the industrial devices in the manufacturing unit or from an organisation for various business purposes like production forecasting, production monitoring, product shipping, and overall data analysis and decision making for marketing, sales, financial analysis, human resource management etc.
- IIoT includes – machine learning, big data technology, machine - to - machine interaction (M-2-M) automation.

Some of the real-life applications of IIoT are

- Increase the production by gathering real-time data from production facilities.
- The AI-based product design cycle uses real-time data to launch a new product in the market.
- Sensor devices and AI-based modern applications detect the defects and damages in the manufacturing units in the early stages and inform the maintenance team before any accidents occur.

The key application areas of IIoT are

- **Manufacturing industry**
  - Smart manufacturing comprises of - interconnection and integration of devices, workforce & work platform
  - This reduces the operational cost and wastage.
- **Healthcare Service industry**
  - Connectivity of healthcare devices to the internet helps to know the status of the patients monitor by them

- Availability of patients' history helps in easy diagnosis.

- **Transportation & logistics**
  - Easy monitoring of equipment, vehicles, engines and tracks using the connected devices, sensors, GPS etc.
  - Optimum scheduling of the routes will provide good customer services by reducing cancellation and delays
  - reduce fuel consumption
  - reduce maintenance expenses of the machines and vehicles
- **Mining**
  - Different gas sensors are used for detecting oxygen, methane and poisonous gasses etc.
  - strata monitoring device, rock mass deformation device is used to detect the internal structural condition of the mine
  - RFID tags are used to track the miners
  - These will result in - early disaster warning, locating and monitoring miners safety and increasing efficiency
- **Firefighting**
  - Sensor networks with RFID tags are used to help in real-time monitoring
  - early warning of disaster

Examples of IIoT

Examples of IIoT are :

- Unmanned aerial vehicles (UAVs) to inspect oil pipelines.
- Monitoring food safety using sensors.
- Minimising workers' exposure to noise,
- Chemicals and other hazardous gases.
- Unmanned marine vehicle which can collect data up to a year without fuel or crew.

Challenges in IIoT

- **Challenges in Data Integration**
  - Big data volume
  - Complex and different types of data is generated at different sensors and actuators
  - Frequency of data generated by multiple devices
  - Understanding the generated data for analysis
- **Cybersecurity**
  - In IIoT all the devices are interconnected with each other and with the real world through the internet
  - These connected devices need to be secured from unauthorized access.

- Security concerns of IIoT are - information security & data privacy.

- **Lack of standardization**
  - There are many different kinds of protocols used in IIoT systems at different network layers, by different kind of devices.
  - There must be a common contract between such different kinds of devices and different platforms of IoT systems, to have compatibility among the IIoT applications.
  - Large automation supplier firms don't accept open standardization, as it will reduce the customer's reliance on them
  - Lack of standardization leads to different issues related to : Device interoperability, Semantic interoperability (data semantics), Security and privacy etc.

- **Lack of skills**
  - IIoT applications contain new digital technologies like - Artificial Intelligence, Data Science, Big Data, Machine Learning, Robotics etc.
  - There is a huge pool of skilled workers required.
  - Because the technologies associated with IIoT are new, the workers must have vast and diverse knowledge
  - It is very difficult to obtain the workers with IIoT related skills

6.4 USE CASES OF THE I-IOT

Manufacturing operations

This includes all operations performed by the manufacturing execution system (MES) that can manage - planning, production optimization, and supply chain management.

- **Asset management :** This includes monitoring of production-assets like - quality, performance, damage or breakdowns, bottlenecks, etc.
- **Field service organizations :** using a well-connected, well-aware, digitized and IoT-enabled manufacturing ecosystem is beneficial for the company.
- **Remote monitoring and operation :** this contains - flow optimization flow, waste elimination, energy saving and cost reduction.
- **Condition-based maintenance :** making machines available, interruption minimization, and increasing the throughput.
- **Big data :** monitors the quality and improves the outcome of this aggregated data.

Fig. 6.4.1 : Use cases of the I-IoT

6.5 IOT AND IIOT - SIMILARITIES AND DIFFERENCES

6.5.1 Similarities between IIoT and IoT

Sr. No	Parameter	IoT	IIoT
1.	Cloud Computing	Required	Required
2.	High-speed internet connection	Required	Required
3.	IoT Platforms and Tools	Required	Required
4.	Common I/O Devices used	HD cameras, beamforming microphones, GPS, geofencing technology, temperature sensors, and water droplet sensors.	HD cameras, beamforming microphones, GPS, geofencing technology, temperature sensors, and water droplet sensors
5.	Artificial Intelligence & Machine Learning	Required	Required

6.5.2 Differences Between IIoT and IoT

Sr. No.	Parameter	IoT	IIoT
1	Focused on	Applications are developed for consumer homes or offices.	Applications are developed for the manufacturing industries, power plants, and large production houses.
2	Applications	IoT applications are developed for small-scale applications, for small group of people	IIoT is used for large-scale applications, and its output could cater to millions of people.
3	Sensors Used	IoT uses basic sensors to detect temperature, motion, and water.	IIoT uses thousands of instruments like pressure sensors, MEMS (Micro Electro - Mechanical System) sensors, speed sensors, RFID sensors, torque sensors, etc.
4	Programming and Networking	most IoT applications require easy programming using the mobile app and do not require complex coding and networking infrastructure	IIoT needs large-scale networking infrastructure and apps with real-time and remote programming capabilities.
5	Security	IoT requires less network security to protect individuals data	IIoT infrastructure requires robust cyber security systems to protect the organization's data
6	Usage	IoT devices come with a shorter life cycle than industrial ones.	IIoT devices are made for robust environments. So, these devices can be roughly used.
7	Reliable	Less	Highly
8	Examples	air conditioners, sensors, smart watches, mobile phones etc	amazon warehouse, smart robotics, Air bus etc.

6.6 INTERNET OF BEHAVIOUR (IOB)

Internet of Behaviours (IoB)

- Using IoB, is used by the organisations, to understand the behaviour of clients / customers to design, develop, launch and promote new products in the market.
- Customer's information is collected through the connected devices (using IoT platform), and this information is monitored and analysed for further use.
- This collected information from the individuals may contain - their interests, their preferences, desires, tastes, priorities, habits etc.

Fig. 6.6.1 : IoT and IoB

- Such type of information can be collected from user's - smartphones, vehicles, exercise reloads, social media accounts, smartphone geolocation data, other online activities, purchasing habits from e-commerce portals, credit cards and from everything else which is connected to the Internet.
- IoB is using facial recognition, place tracking, big data and science, data processing, and the psychology of behavior, to monitor, analyze, perceive and respond to human behaviors.
- The IoT translates data to information, and the IoB may translate our knowledge into genuine wisdom, as seen in the pyramid, Fig. 6.6.2.

(1F12)**Fig. 6.6.2 : Internet of Things (IoT) and Internet of Behaviour (IoB) pyramid**

6.7 SELF-LEARNING TOPICS

6.7.1 Internet of Behaviours (IoB) and its Role in Customer Services

- Customer data may come from various places, including our social media activity, smartphone geolocation data, credit card purchases, and even food preferences.
- Using this data, interested parties may obtain more detailed insight into people's behavior
- Several companies have created health apps for cell phones that track food, sleep habits, heart rate, and blood sugar levels in the medical field.
- Netflix, for example, utilizes user data to forecast what they like or dislike.

- They provide suggestions based on their personal preferences and ratings for a specific film or series.
- Another example, Uber or any other cab aggregator has been utilizing IoT to track drivers and passengers.
- They offer a survey after each journey to review the user's experience. to understand the passenger's reaction, and build on that input via IoB

Fig. 6.7.1 : Different strategies of IoB for customer services

- As shown in Fig. 6.7.1, various tools, traits, and strategies associated with the Internet of Behavior (IoB), improves the overall customer satisfaction and make their purchase valuable and worthy.
- The IoB affects customer choice, and it also restructures the supply chain.
- It offers companies to improve their profile, like insurance companies and banks.
- Companies supply data-driven utilities by extracting them from the IoT.
- IoB can forecast responses to shopping ads or social media messages
- Customers can be involved in the product and service production, to develop or create personalized and customized products as per customer requirements.
- In this digital era, the environment defines human behavior.
- The internet is connected with computers and other IoT devices and data points.
- Based on the customers' behavior of purchasing and using the things, companies can identify the needs of their customer and then develop the products required by the customer.
- IoB businesses are using statistics, knowledge and behavior trends to meet their customers' requirements.
- Using IoB, businesses monitor the actions of their workers and build strong relationships between employees.
- IoB connects and interprets the response to technology and the actions of an individual.

6.8 CASE STUDIES BASED IN IOB

- A mobile phone (smartphone) can be easily connected to a laptop, a voice assistant, a smart home system or to a smart vehicle.
- Marketing research algorithms from Google, Facebook or Amazon are configured to anticipate customer desires and behaviors.

Case study 1 : Health App

- A software company has developed a health app for smartphones that tracks diet, sleep patterns, heart rate or blood sugar levels of the user.
- The app can alert adverse situations in the user's health and suggest behavioral modifications for the more positive outcome.
- Health Passport (with apps such as Aarogya Setu in India, and The Health Code in China) and Social Distancing Technologies are partners in this emerging health technology.

Case Study 2 : Transportation

- In relation to transportation, for example, Uber, uses IoT data on drivers, passenger locations, peak timings and preferences.
- Also, large companies, such as Ford, have joined other start-ups, such as Argo AI, to design autonomous cars that vary their behavior in each city based on vehicle traffic, pedestrians, bicycles and scooters.

Case Study 3 : Playing Golf

- A software company has carried out a project to help golfers improve their playing skills with the help of a mobile application and tracking of wearable devices, namely correcting existing ball striking techniques and learning new techniques.
- Making use of a handheld device connected to the mobile phone, each hit on the golf ball is recorded in the app and analyzed (stroke force, trajectory, angle, etc.).
- As a result, the player can see their mistakes and get visual recommendations on how to improve their swing and stroke.

Chapter Ends.

❑❑❑