Unit II
Internet of Things : Concepts

Topics to cover

- **Introduction to Internet of Things (IoT)**: Definition, Characteristics of IoT, Vision, Trends in Adoption of IoT, IoT Devices, IoT Devices Vs Computers, Societal Benefits of IoT, Technical Building Blocks.
- **Physical Design of IoT**: Things in IoT, Interoperability of IoT Devices, Sensors and Actuators, Need of Analog / Digital Conversion.
- **Logical Design of IoT**: IoT functional blocks, IoT enabling technologies, IoT levels and deployment templates, Applications in IoT.
- **#Exemplar/Case Studies** Exemplary device: Raspberry Pi / Arduino: Programming: Arduino IDE/ Python, Interfacing. Other IoT Devices.
- ***Mapping of Course Outcomes for Unit II :** CO1,CO2

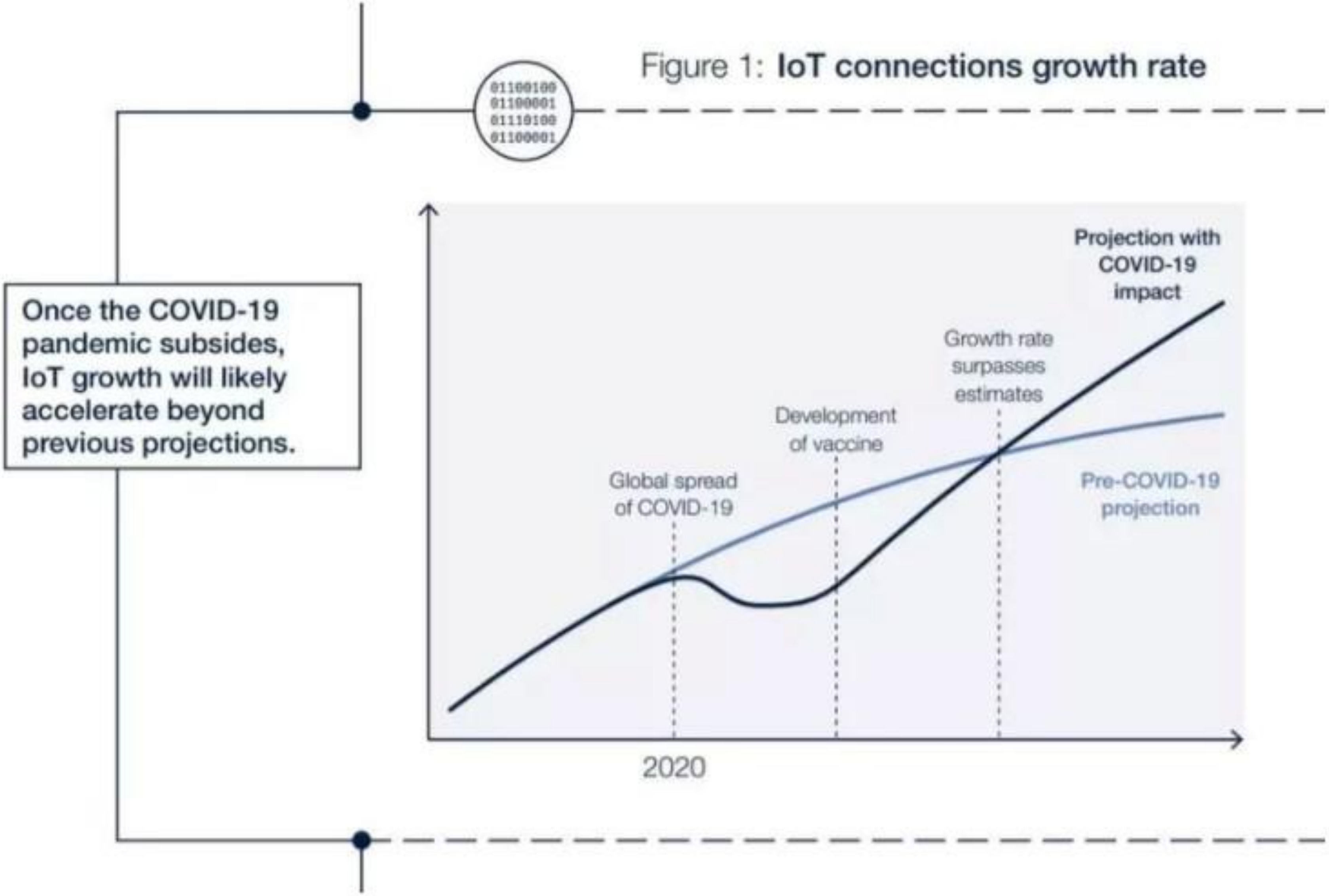
Course Objectives:

- CO1 : To understand fundamentals of Internet of Things (IoT) and Embedded Systems.
- CO2 : To learn advances in Embedded Systems and IoT.

- From soil moisture sensors being used to optimize farmer's yields, to thermostats and thermometers, the Internet of Things (IoT Technologies) is transforming the way we live and work.
- Billions of networked 'smart' physical objects around the world, on city streets, in homes and hospitals, are constantly collecting and sharing data across the internet, giving them a level of digital intelligence and autonomy.

- Around a quarter of businesses were using IoT technologies in 2019, according to McKinsey, up from 13% in 2014.
- And already, there are more connected devices than people in the world, according to the World Economic Forum's State of the Connected World report, and it is predicted that by 2025, 41.6 billion devices will be capturing data on how we live, work, move through our cities and operate and maintain the machines on which we depend.
- The digital transformation that is taking place due to emerging technologies, including robotics, the IoT and artificial intelligence, is known as the Fourth Industrial Revolution – and COVID-19 has accelerated the use of these technologies.

Figure 1: **IoT connections growth rate**



Once the COVID-19 pandemic subsides, IoT growth will likely accelerate beyond previous projections.

How COVID-19 has sped up the adoption on IoT technologies.

Image: World Economic Forum

https://www.visionofhumanity.org/what-is-the-internet-of-things/

A Brief History of IoT Technologies

- The concept of adding sensors and intelligence to physical objects was first discussed in the 1980s, when some university students decided to modify a Coca-Cola vending machine to track its contents remotely.
- But the technology was bulky and progress was limited.

History of IoT

- Kevin Ashton, co-founder of the Auto-ID Center at MIT, first mentioned the internet of things in a presentation he made to Procter & Gamble (P&G) in 1999.
- Proposed **radio frequency ID (RFID)** chips on products to track them through a supply chain.
- Ashton called his presentation "Internet of Things" to incorporate the cool new trend of 1999: the internet

- In 2000, LG announced the first smart refrigerator
- In 2007 the first iPhone was launched
- By 2008, the number of connected devices exceeded the number of people on the planet.
- In 2009, Google started testing driverless cars
- In 2011, Google's Nest smart thermostat hit the market, which allowed remote control of central heating.

Everyday Examples of Use Cases

Connected devices fall into three domains:

1. **consumer** IoT, such as wearables,
2. **enterprise** IoT, which includes smart factories and precision agriculture,
3. **public spaces** IoT, such as waste management.

- Businesses use IoT to optimize their supply chains, manage inventory and improve customer experience, while smart consumer devices such as the Amazon Echo speaker, are now ubiquitous in homes due to the prevalence of low-cost and low-power sensors.
- Cities have been deploying IoT technology for more than a decade – to streamline everything from water meter readings to traffic flow.

Internet of Things (IoT)

- The **internet of things, or IoT**, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.
- The concept of basically connecting any device with an on and off switch to the Internet (and/or to each other).
- This includes everything from cellphones, coffee makers, washing machines, headphones, lamps, wearable devices and almost anything else you can think of.

Internet of Things (IoT)

- IOT refers to the ever-growing network of physical objects that feature an **IP address** for internet connectivity, and the communication that occurs between these objects and other Internet- enabled devices and systems.

OR

- The **Internet of Things** (IoT) is the network of devices such as vehicles, and home appliances that contain electronics, software, actuators, and connectivity which allows these things to connect, interact and exchange data

Vision of IoT

- The vision of the IoT can be seen from two perspectives **"Internet-centric" and "thing-centric."**

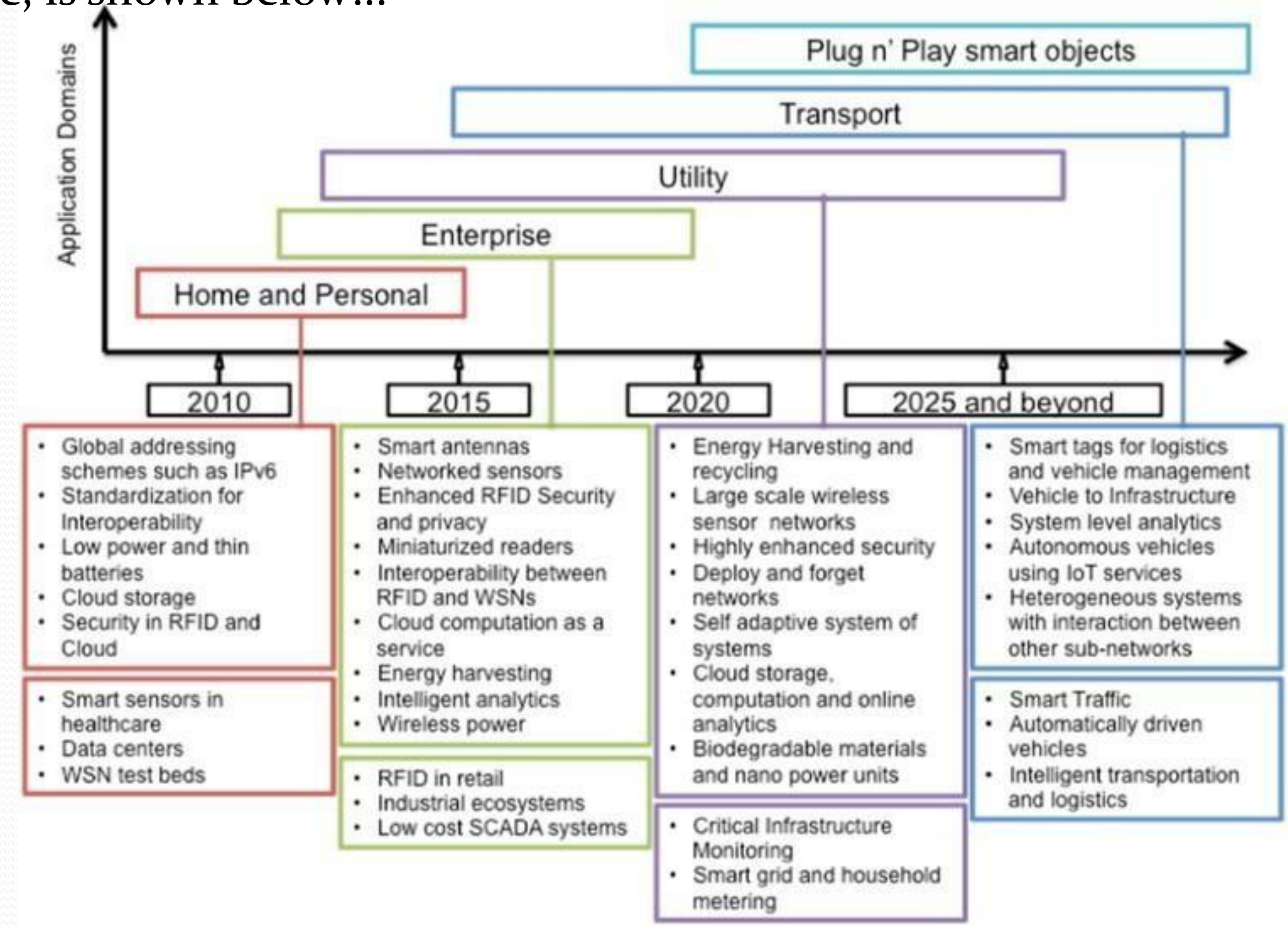
- The **Internet-centric architecture** involves Internet services as the main focus, as data is being generated by the — "things".

- In the **thing-centric architecture**, smart devices take the center stage.

- Equinix is at the heart of the interconnected world of the Internet, so we're focused on the Internet-centric view.

Vision of IoT

A roadmap of key developments in IoT research, which includes the technology drivers and key application outcomes expected in the next decade, is shown below:::



Application Domains

Plug n' Play smart objects

Transport

Utility

Enterprise

Home and Personal

2010

- Global addressing schemes such as IPv6
- Standardization for Interoperability
- Low power and thin batteries
- Cloud storage
- Security in RFID and Cloud

- Smart sensors in healthcare
- Data centers
- WSN test beds

2015

- Smart antennas
- Networked sensors
- Enhanced RFID Security and privacy
- Miniaturized readers
- Interoperability between RFID and WSNs
- Cloud computation as a service
- Energy harvesting
- Intelligent analytics
- Wireless power

- RFID in retail
- Industrial ecosystems
- Low cost SCADA systems

2020

- Energy Harvesting and recycling
- Large scale wireless sensor networks
- Highly enhanced security
- Deploy and forget networks
- Self adaptive system of systems
- Cloud storage, computation and online analytics
- Biodegradable materials and nano power units

- Critical Infrastructure Monitoring
- Smart grid and household metering

2025 and beyond

- Smart tags for logistics and vehicle management
- Vehicle to Infrastructure
- System level analytics
- Autonomous vehicles using IoT services
- Heterogeneous systems with interaction between other sub-networks

- Smart Traffic
- Automatically driven vehicles
- Intelligent transportation and logistics

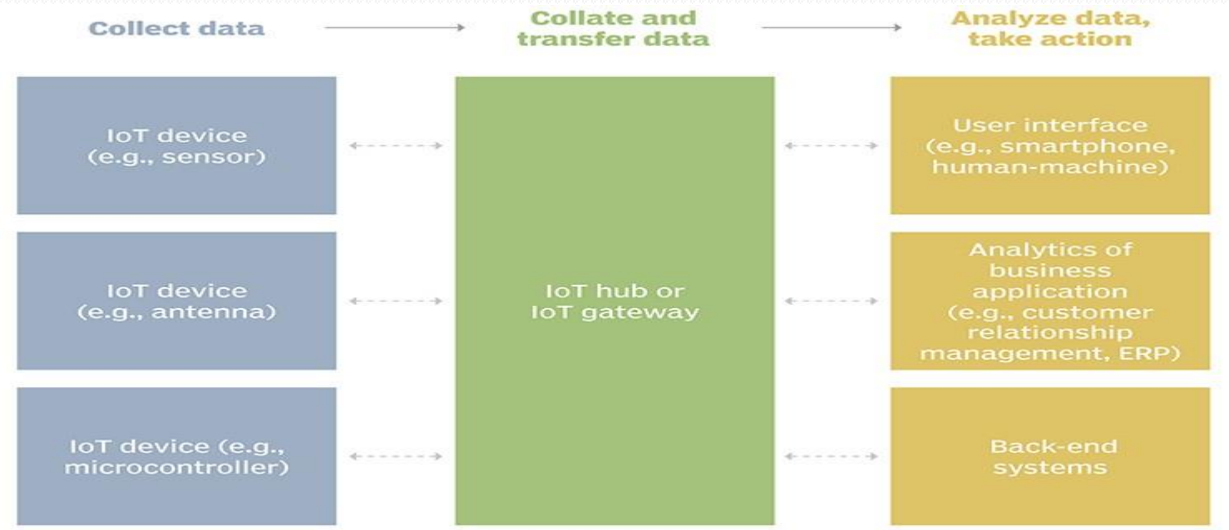
The IoT challenges:

- ❖ The increasing number of real-time IoT apps will stress the performance capabilities of today’s networks. To provide a high- quality user experience, we need to find ways to reduce the end-to-end latency among machine-to-machine interactions to single-digit milliseconds.
- ❖ In addition to pushing the limits of the network, IoT challenges include: privacy, participatory sensing, data analytics, geographic information system-based visualization and cloud computing.
- ❖ Networks also face standard wireless sensor network issues, including architecture, energy-efficiency, security, protocols, and Quality of Service.

How IOT Works

- An IoT ecosystem consists of **web-enabled smart devices** that use embedded processors, sensors and communication hardware to collect, send and act on data they acquire from their environments.
- IoT devices share the sensor data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analyzed or analyzed locally.
- Sometimes, these devices communicate with other related devices and act on the information they get from one another.
- The devices do most of the work without human intervention, although people can interact with the devices -- for instance, to set them up, give them instructions or access the data.

Example of an IOT system



Collect data

Collate and transfer data

Analyze data, take action

IoT device (e.g., sensor)

IoT device (e.g., antenna)

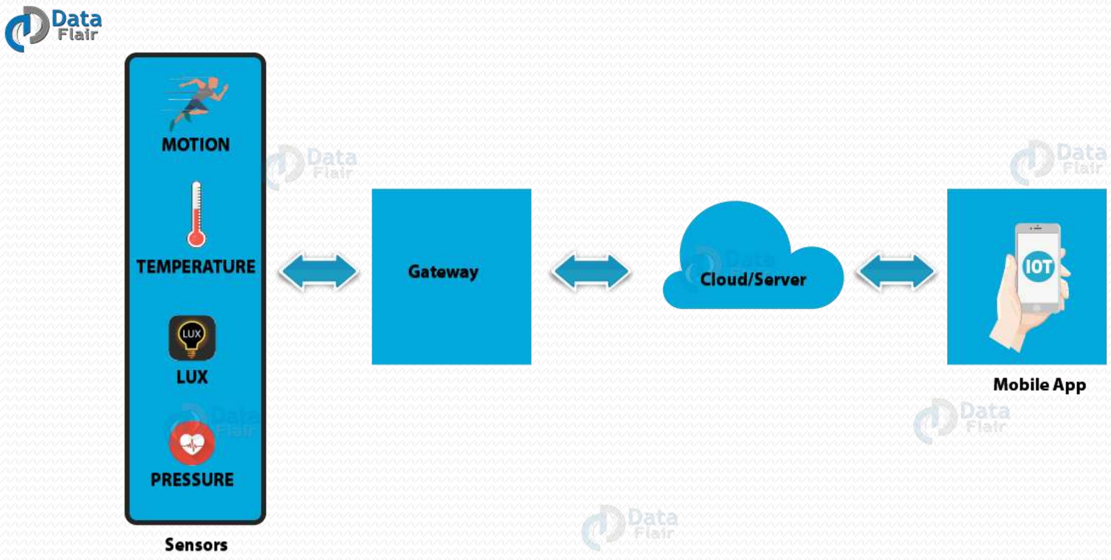
IoT device (e.g., microcontroller)

IoT hub or IoT gateway

User interface (e.g., smartphone, human-machine)

Analytics of business application (e.g., customer relationship management, ERP)

Back-end systems



Benefits of IoT

- The internet of things offers a number of benefits to organizations, enabling them to:
  - monitor their overall business processes;
  - improve the customer experience;
  - save time and money;
  - enhance employee productivity;
  - integrate and adapt business models;
  - make better business decisions; and
  - generate more revenue.

Revision

IoT Intro….

- Internet of Things (IoT) comprises things that have unique identities and are connected to the internet.

IoT Intro [Points to remember]….

1.Existing devises , such as networked computers or 4G enabled mobile phones already have some form of unique identities and are also connected to the internet, the focus on IoT in the configuration, control and networking via the internet of devices or things , that are traditionally not associated with the Internet. These include devices such as thermostats, utility meters, a blue tooth- connected headset, irrigation pumps and sensor or control circuits for an electric car’s engine

IoT Intro [Points to remember]….

2. The scope of IoT is not limited to just connected things(Devices, appliance, machines) to the Internet.

3. Applications on IoT networks extract and create information from lower level data by filtering, processing , categorizing, condensing and contextualizing the data.

4. The information obtained is then organized and structured to infer knowledge about the system and or its user, its environment and its operations and progress towards its objectives, allowing a smarter performance.

Definition of IoT

- 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 virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network, often communicate data associated with users and their environments.

Characteristics of IoT

- Dynamic & Self-Adapting
- Self-Configuring
- Interoperable Communication Protocols
- Unique Identity
- Integrated into Information Network

1. Dynamic and self-Adapting

- IoT devices and systems may have the capability to dynamically adapt with the changing contexts and take actions based on their operating condition.
- Ex: Surveillance cameras can adapt their modes based on whether it is day or night.

2. Self – Configuring

- IoT devices may have self-Configuring capability allowing a large number of devices to work together to provide certain functionality .

3. Interoperable communication protocols

- IoT Devices may support a number of interoperable communication protocols and can communicate with other devices and also with the infrastructure.

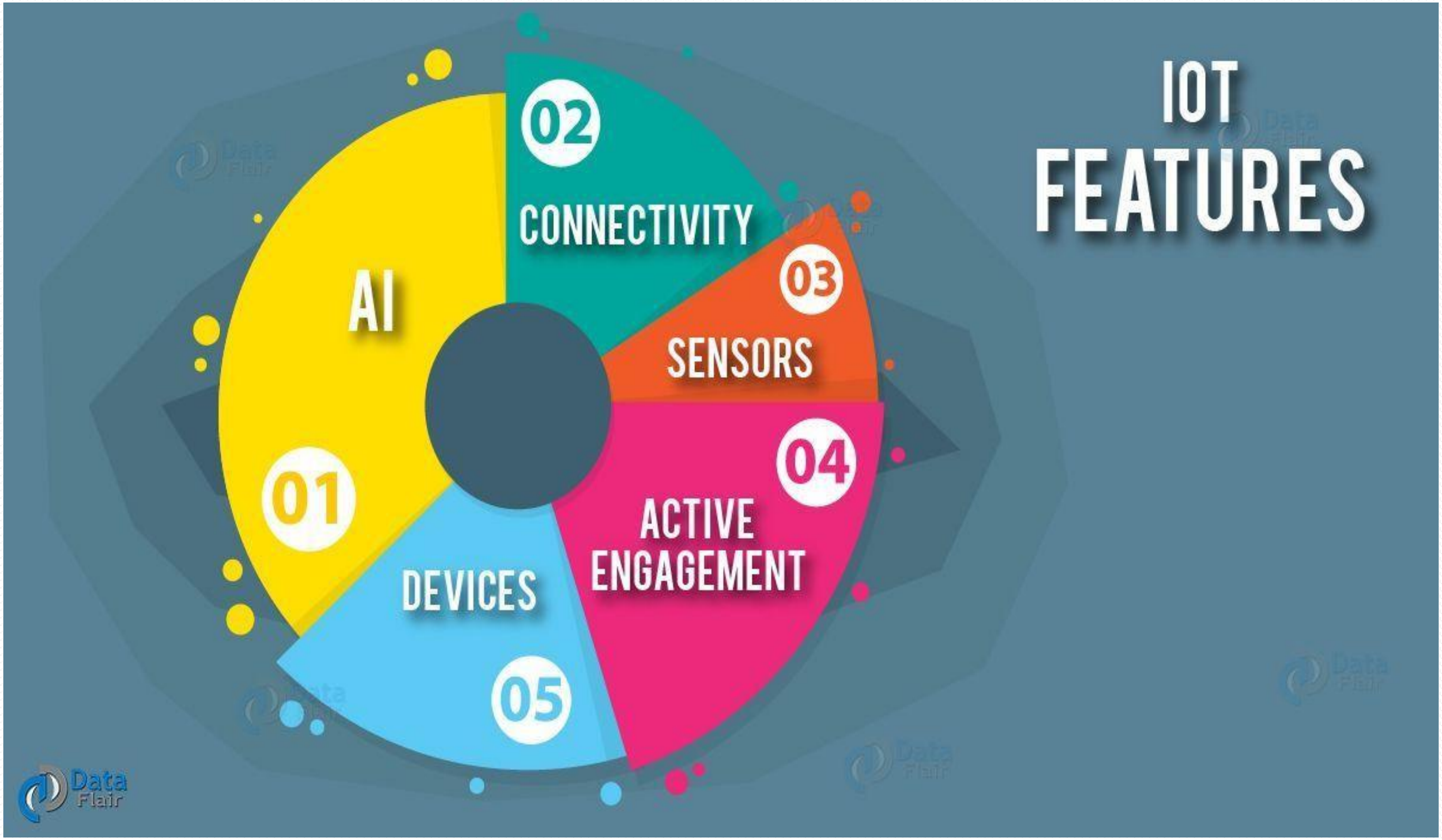
4. Unique Identity

- Each IoT devices has a unique identity and a unique identifier (IP address, URI).
- IoT systems may have intelligent interfaces which adapt based on the context, allow communication with users and the environment contexts.

5. Integrated into information network

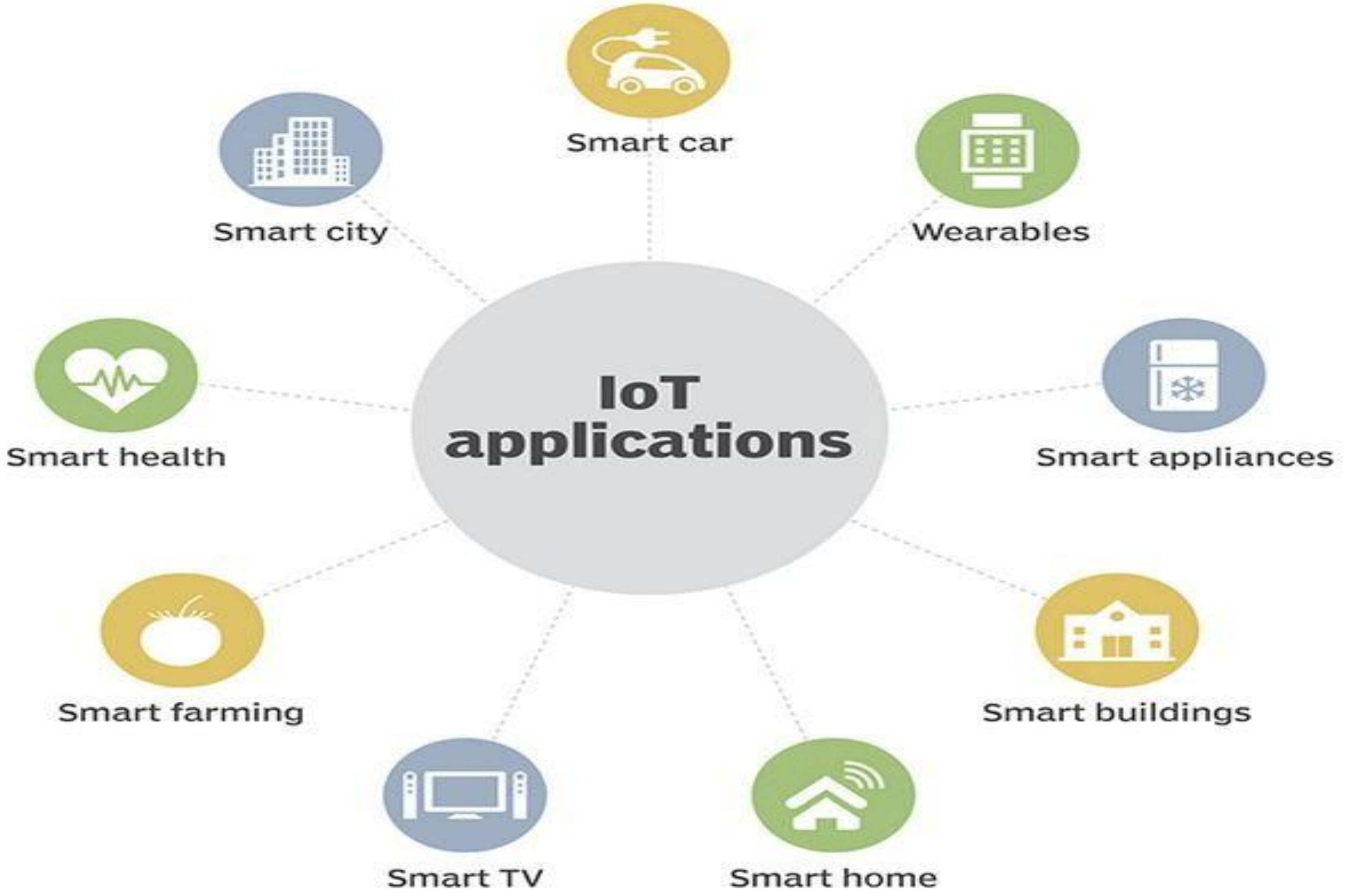
- IoT devices are usually integrated into the information network that allows them to communicate and exchange data with other devices and systems.

Features of IoT



https://data-flair.training/blogs/iot-tutorial/

Application of IOT



Internet of Things: Overview

- Internet has undergone remarkable changes since its first launch in 1960s.

- Internet is transformed to service oriented **ubiquitous** infrastructure due to anything, anytime and anywhere operations.

- In such ambient environment not only users become ubiquitous but also devices.
- The **context** of devices becomes transparent and ubiquitous.

- Networked world swamped with
  - information.
  - Sources of information,
  - Services present on devices,
  - communication infrastructures
  - the Internet.
- Need for privacy and security model.

- IoT is mandatory subset of future Internet.
- Every virtual and physical device can communicate.
- Seamless services to all stakeholders.

Internet of Things: Vision

●The end goal is to have plug-n-play smart objects that can be deployed in any environment with an interoperable interconnection backbone that allows them to blend with other smart objects around them.

●Standardization of frequency bands and protocols plays a pivotal role in accomplishing this goal.

- Interconnecting and intercommunicating devices.
- Using technologies such as RFID, NFC, ZigBee, Wifi and Bluetooth.

Trends in development and deployment of IoT applications

- Miniaturization of devices.
- Mobile phones as information gathering.
- Lower power devices
- Support for big data.
- Smart management.

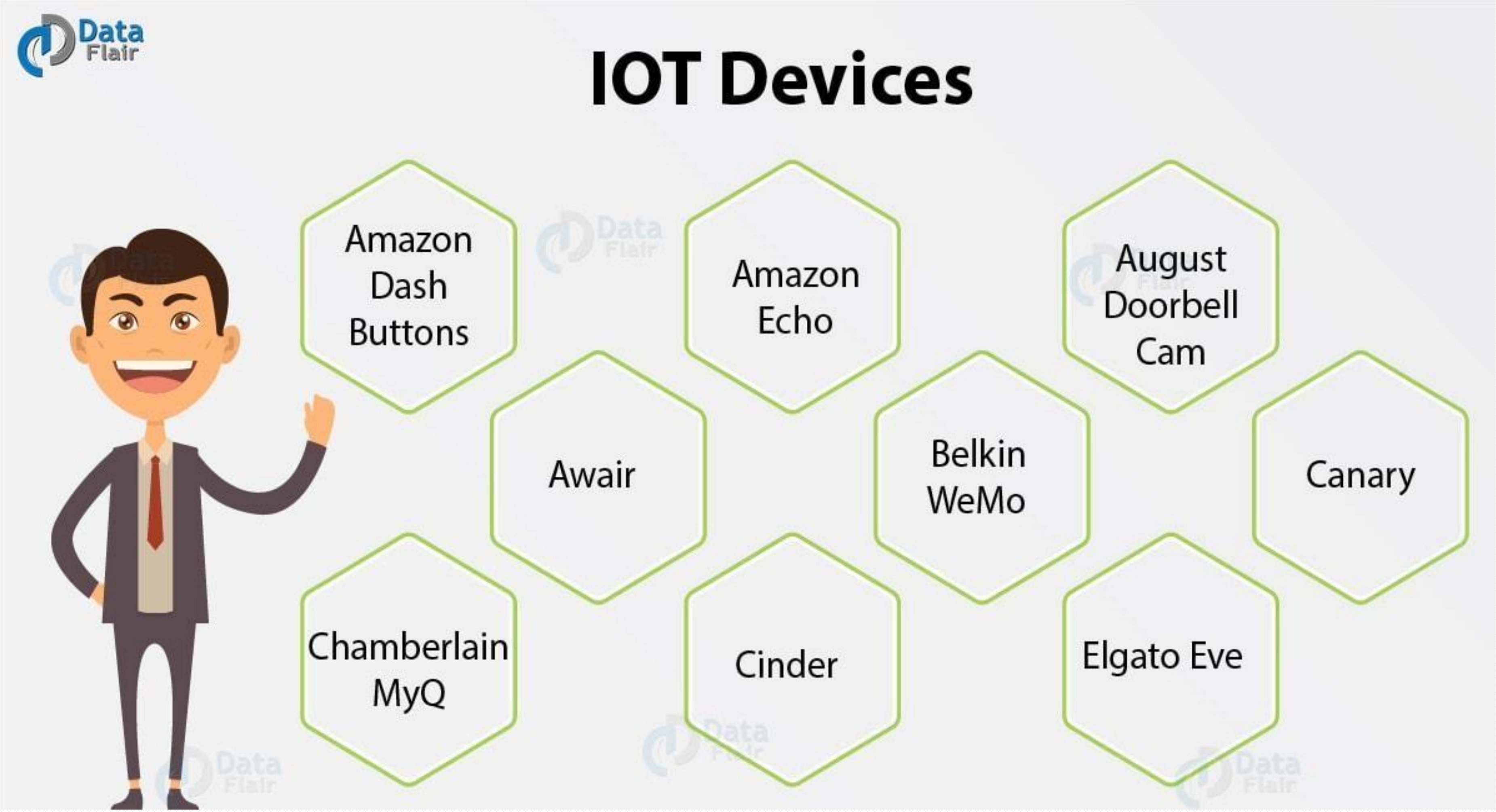
Points Covered

- Introduction to Internet of Things (IoT):
  - ✓ Definition
  - ✓ Characteristics of IoT
  - ✓ Vision
  - ✓ Trends in adaption of IoT

IoT Devices

- Internet of things (IoT) devices are nonstandard computing hardware -- such as sensors, actuators or appliances -- that connect wirelessly to a network and can transmit data.
- IoT extends internet connectivity beyond typical computing devices -- such as desktops, laptops, smartphones and tablets -- to any range of traditionally *dumb* or non-internet-enabled physical devices and everyday objects.
- Embedded with technology, these devices can communicate and interact over the internet, and can be remotely monitored and controlled.

IoT Devices



IOT Devices

- Amazon Dash Buttons
- Amazon Echo
- August Doorbell Cam
- Awair
- Belkin WeMo
- Canary
- Chamberlain MyQ
- Cinder
- Elgato Eve

IoT Devices

- IoT devices have both industrial and consumer uses and are typically integrated into other tools such as mobile devices, industrial equipment and medical devices.
- Over a broad range, they can also be used in smart cities.
- They're then used to send data or interact with other IoT devices over a network.
- IoT and IoT devices aid in making daily activities faster, easier or more convenient for consumers while also providing real-time data for industrial or enterprise use cases.

How does IoT work?

- A typical IoT system works through the real-time collection and exchange of data.
- An IoT system has three components:

1.Smart devices

2.IoT application

3.A graphical user interface

1.Smart devices

- This is a device, like a television, security camera, or exercise equipment that has been given computing capabilities.
- It collects data from its environment, user inputs, or usage patterns and communicates data over the internet to and from its IoT application.

2.IoT application

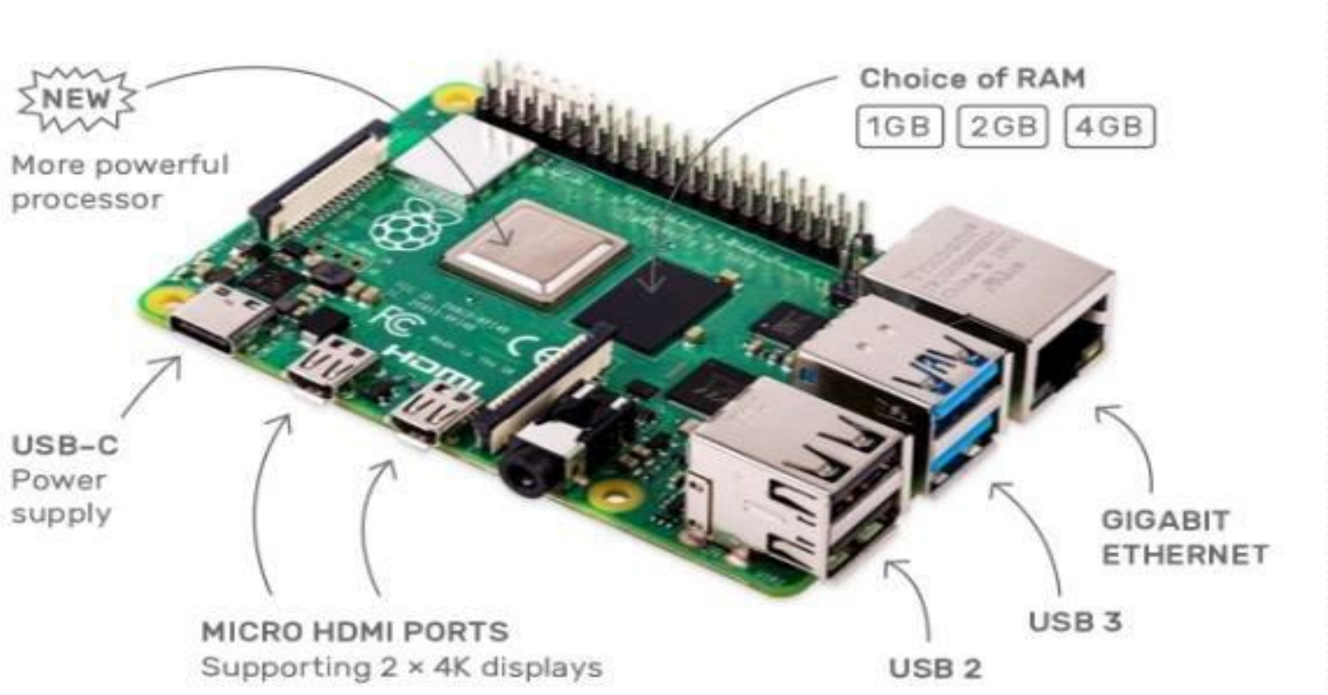
- An IoT application is a collection of services and software that integrates data received from various IoT devices.
- It uses machine learning or artificial intelligence (AI) technology to analyze this data and make informed decisions.
- These decisions are communicated back to the IoT device and the IoT device then responds intelligently to inputs.

3.A graphical user interface

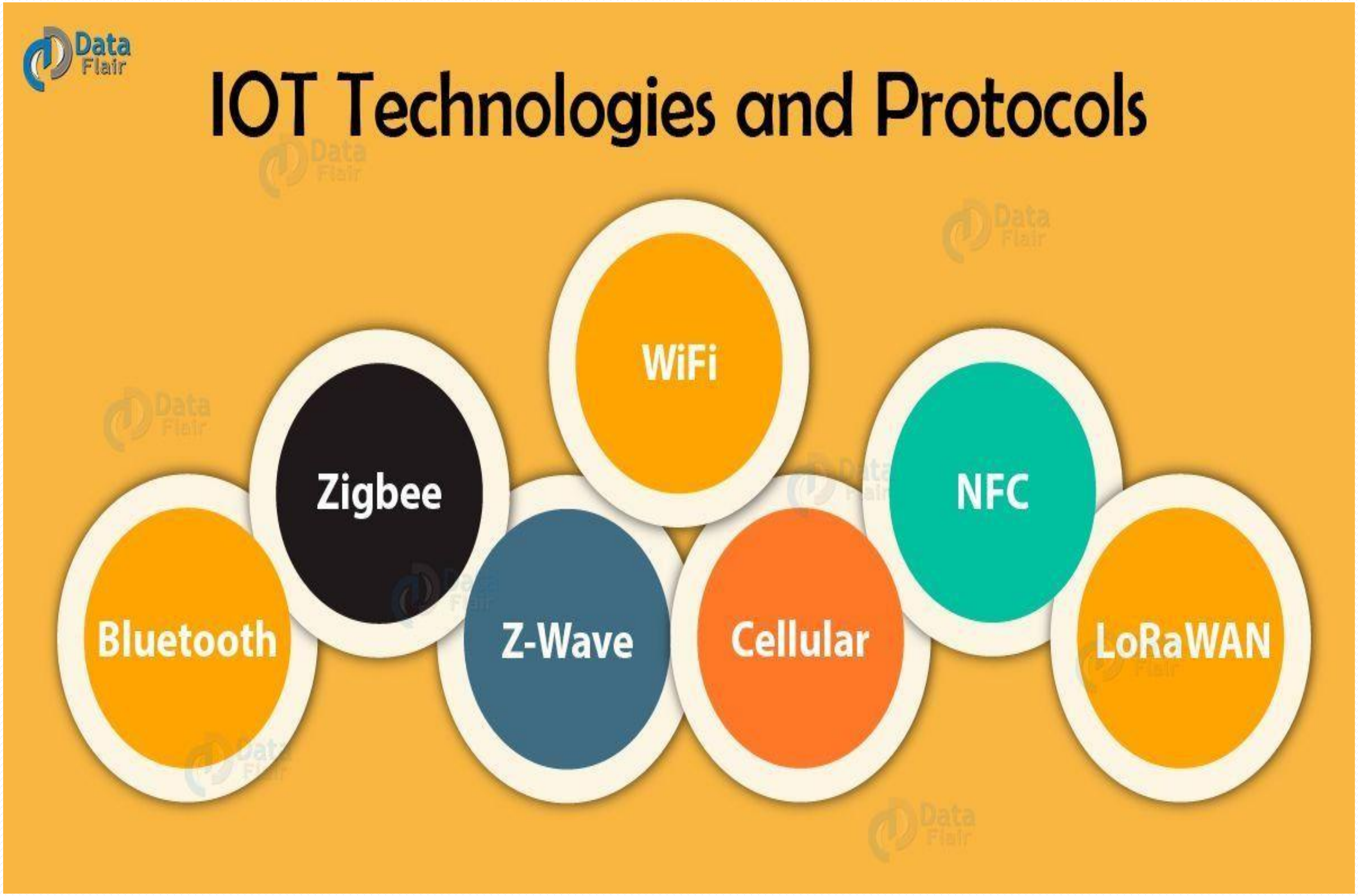
- The IoT device or fleet of devices can be managed through a graphical user interface.
- Common examples include a mobile application or website that can be used to register and control smart devices.

IoT Device Example

- Raspberry Pi 4 model B



IoT Technology



Difference between IoT devices and Computers:

IOT Devices	Computers
IoT devices are special-purpose devices.	Computers are general-purpose devices.
IoT devices can do only a particular task for which it is designed.	Computers can do so many tasks.
The hardware and software built-in in the IoT devices are streamlined for that particular task.	The hardware and software built-in in the computers are streamlined to do many tasks(such as calculation, gaming, music player, etc.)
IoT devices can be cheaper and faster at a particular task than computers, as IoT devices are made to do that particular task.	A computer can be expensive and slower at a particular task than an IoT device.
Examples: Music Player- iPod, Alexa, smart cars, etc.	Examples: Desktop computers, Laptops, etc.

What are the benefits of IoT in society?

- The Internet of Things (IoT) has had a significant impact on society in several ways:

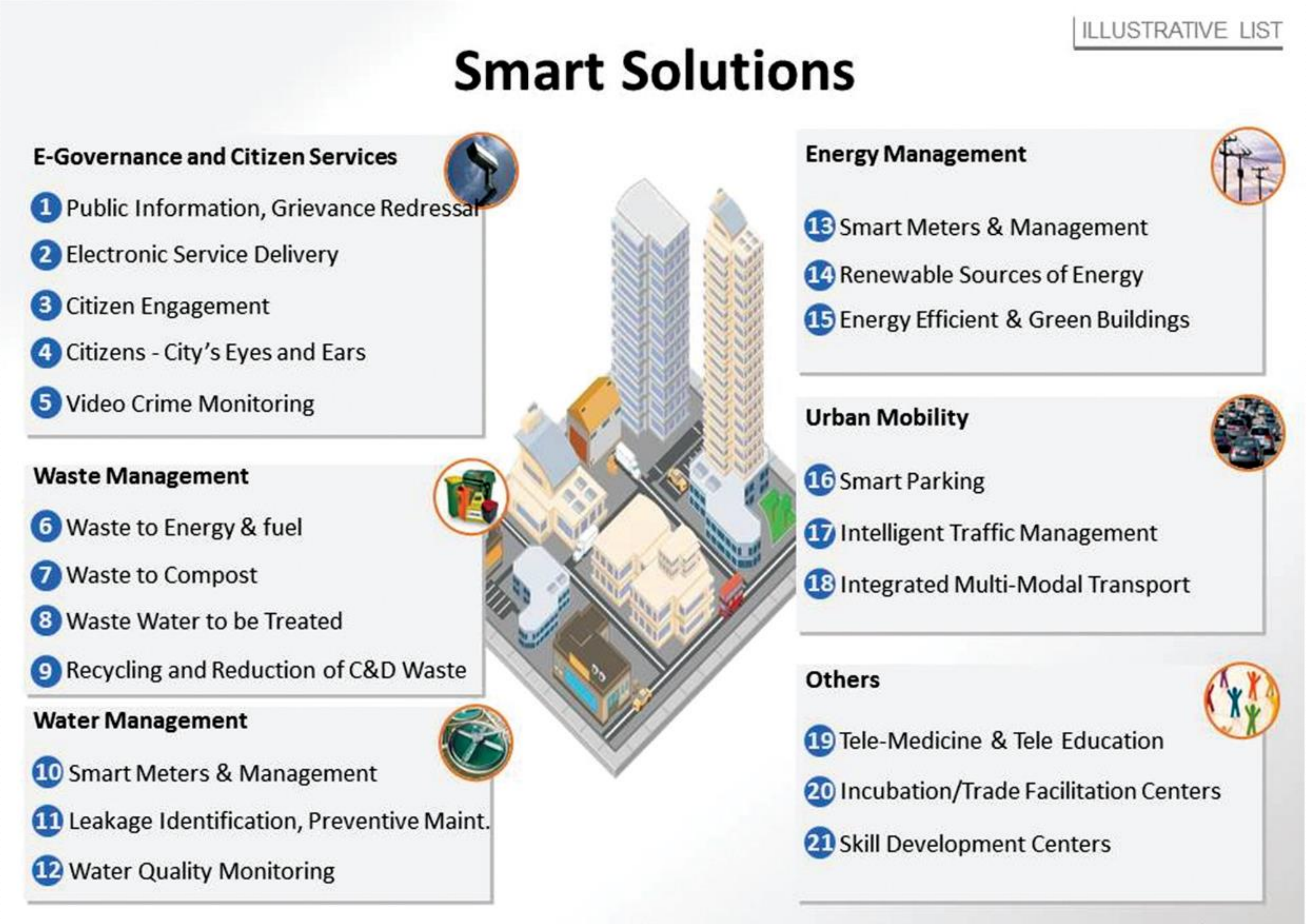
1. **Improved efficiency**: The IoT has made it possible to automate processes and connect devices, leading to increased efficiency in industries such as manufacturing, logistics, and healthcare.
2. **Enhanced convenience**: IoT has made it possible for people to remotely control devices such as thermostats, lighting, and security systems, making their lives more convenient.
3. **Better health outcomes**: IoT has enabled the creation of wearable devices that monitor vital signs and provide real-time data to healthcare professionals, helping them to provide better care.

What are the benefits of IoT in society?

4. **Enhanced safety and security**: IoT can help monitor and respond to emergency situations more quickly and effectively, improving public safety. For example, smart home security systems can provide remote monitoring and control, while smart city technology can detect and respond to natural disasters, traffic congestion, and other emergencies.
5. **Environmental benefits**: IoT can help reduce waste and conserve energy by optimizing resource utilization and reducing emissions. For example, smart home technology can reduce energy usage and smart city technology can optimize public transportation, reducing traffic and air pollution.
6. **New business opportunities**: IoT has created new business opportunities in areas such as smart homes, wearable technology, and industrial automation, leading to job creation and economic growth.

https://www.arduino.cc/education/societal-benefits-of-the-iot

Digital India



ILLUSTRATIVE LIST

Smart Solutions

E-Governance and Citizen Services

1. Public Information, Grievance Redressal
2. Electronic Service Delivery
3. Citizen Engagement
4. Citizens - City's Eyes and Ears
5. Video Crime Monitoring

Waste Management

6. Waste to Energy & fuel
7. Waste to Compost
8. Waste Water to be Treated
9. Recycling and Reduction of C&D Waste

Water Management

10. Smart Meters & Management
11. Leakage Identification, Preventive Maint.
12. Water Quality Monitoring

Energy Management

13. Smart Meters & Management
14. Renewable Sources of Energy
15. Energy Efficient & Green Buildings

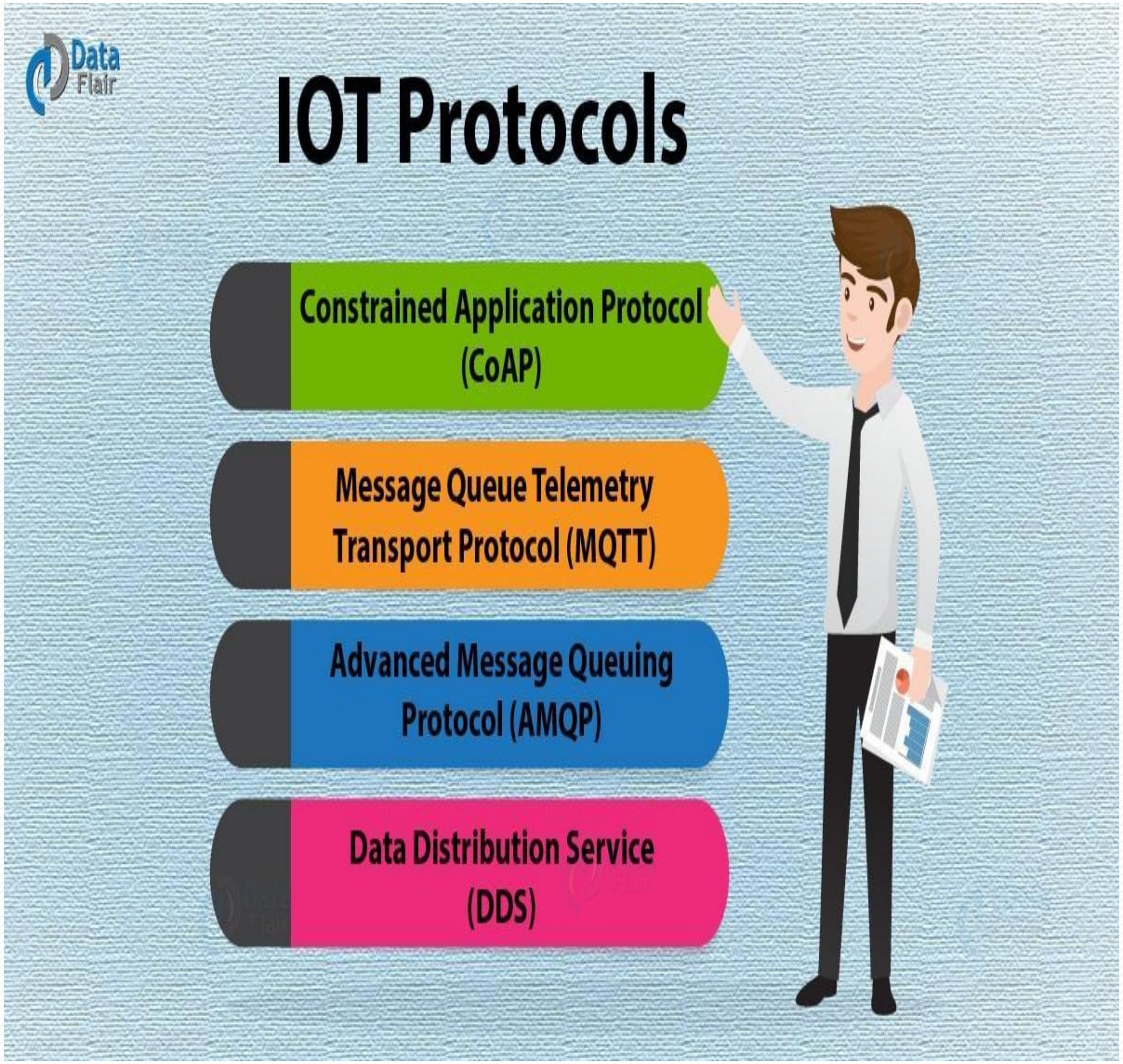
Urban Mobility

16. Smart Parking
17. Intelligent Traffic Management
18. Integrated Multi-Modal Transport

Others

19. Tele-Medicine & Tele Education
20. Incubation/Trade Facilitation Centers
21. Skill Development Centers

IoT Protocols



Data Flair
IOT Protocols
Constrained Application Protocol (CoAP)
Message Queue Telemetry Transport Protocol (MQTT)
Advanced Message Queuing Protocol (AMQP)
Data Distribution Service (DDS)

IoT security and privacy issues

- The internet of things connects billions of devices to the internet and involves the use of billions of data points, all of which need to be secured.
- Due to its expanded attack surface, IoT security and IoT privacy are

  cited as major concerns.
- One of the most notorious recent IoT attacks was **Mirai-a Botnet**

What is Mirai Botnet???

- Mirai is a self-propagating botnet virus.
- The source code for Mirai was made publicly available by the author after a successful and well publicized attack on the **Brian Krebs blog** [investigating the vDOS botnet].
- Since then the source code has been built and used by many others to launch attacks on internet infrastructure (ref Dyn).
- The Mirai botnet code **infects poorly protected internet devices** by using telnet to find those that are still using their factory **default username and password.**

Mirai Botnet

- The Mirai botnet is a malware designed to hijack Internet of Things (IoT) devices and turn them into remotely controlled “bots” capable of launching powerful volumetric distributed denial of service (DDoS) attacks.
- First seen in August 2016 and has since been used to launch large DDoS attacks on websites, networks and other digital infrastructure.
- Mirai was published as a source code by “Anna-senpai” to a public and easily accessible forum.
- The malicious code allows an attacker to gain control of vulnerable IoT devices such as webcams, DVRs, IP cameras, and routers.
- In early 2017, Krebs publicly named Josiah White and Paras Jha as the likely creators of Mirai botnet.

How does Mirai work?

- The Mirai botnet works by scanning for vulnerable IoT devices that have open ports or default usernames and passwords.
- Once it finds these vulnerable devices, it uses exploits to gain access and infects them with its malicious code.
- The infected device then joins the Mirai botnet which allows the attacker to send commands from a central server which is known as a "command & control" server (C&C).
- This C&C server can then be used to launch large-scale DDoS attacks on websites, networks and other digital infrastructure by using all of the bots in the Mirai Botnet at once.

How Mirai Works

- ❖ Two main components to Mirai:
- ✓ the virus itself and
- ✓ Command and Control Center (CnC).

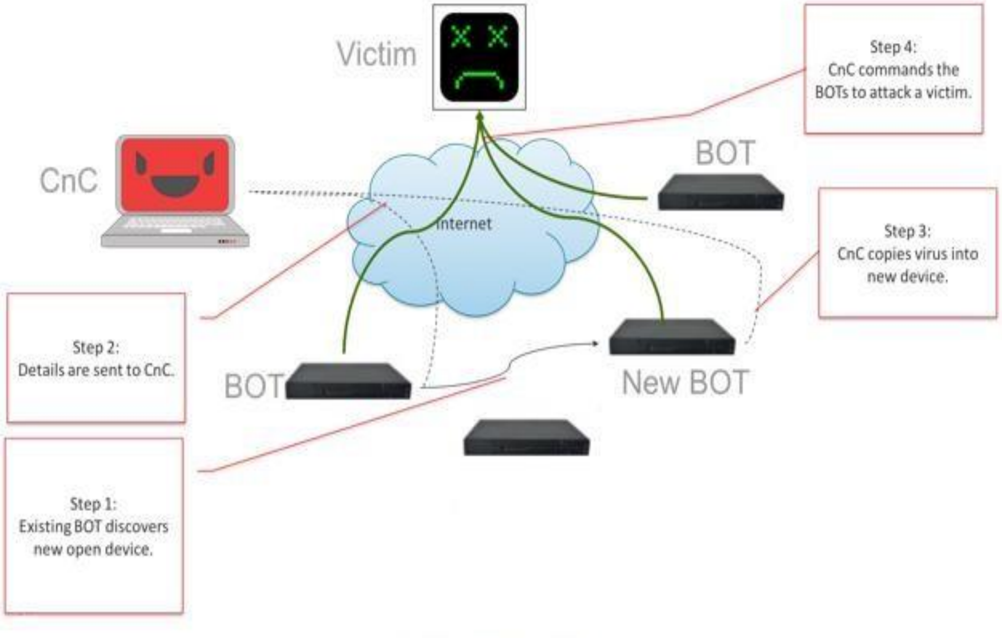

Victim
Step 4: CnC commands the BOTs to attack a victim.
CnC
BOT
Internet
Step 3: CnC copies virus into new device.
Step 2: Details are sent to CnC.
BOT
New BOT
Step 1: Existing BOT discovers new open device.

Figure 1 Mirai System

- ➢ The virus contains the attack vectors, Mirai has ten vectors that it can launch, and a scanner process that actively seeks other devices to compromise.
- ➢ The CnC is a separate image that controls the compromised devices (BOT) sending them instructions to launch one of the attacks against one or more victims.

How Mirai Works

- The scanner process runs continuously on each BOT using the telnet protocol (on TCP port 23 or 2323) to try and login to IP addresses at random.
- The login tries up to 60 different factory default username and password pairs when login succeeds the identity of the new BOT and its credentials are sent back to the CnC.
- The CnC supports a simple command line interface that allows the attacker to specify an attack vector, a victim(s) IP address and an attack duration.
- The CnC also waits for its existing BOTs to return newly discovered device addresses and credentials which it uses to copy over the virus code and in turn create new BOTs.

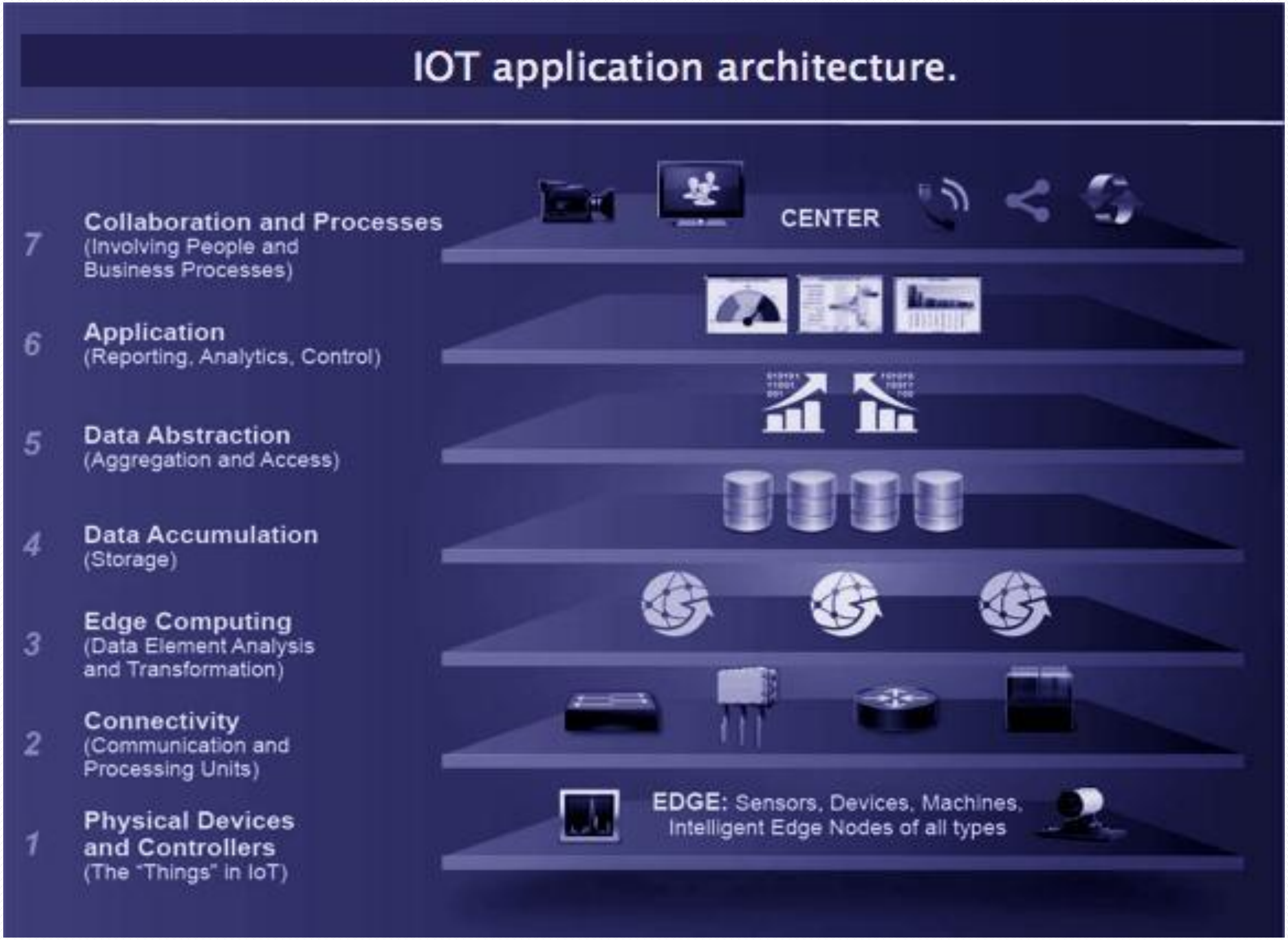
The Mirai Code

- The virus is built for multiple different CPU architectures (x86, ARM, Sparc, PowerPC, Motorola)
- Once the virus is loaded into memory on the BOT it deletes itself from the BOT's disk.
- The virus will remain active until the BOT is rebooted.
- Immediately after a reboot the device is free of the virus however it only takes a few minutes before its once again discovered and re-infected.
- The attack vectors are highly configurable from the CnC but by default Mirai tends to randomize the various fields (port numbers, sequence numbers, ident etc) in the attack packets so they change with every packet sent.

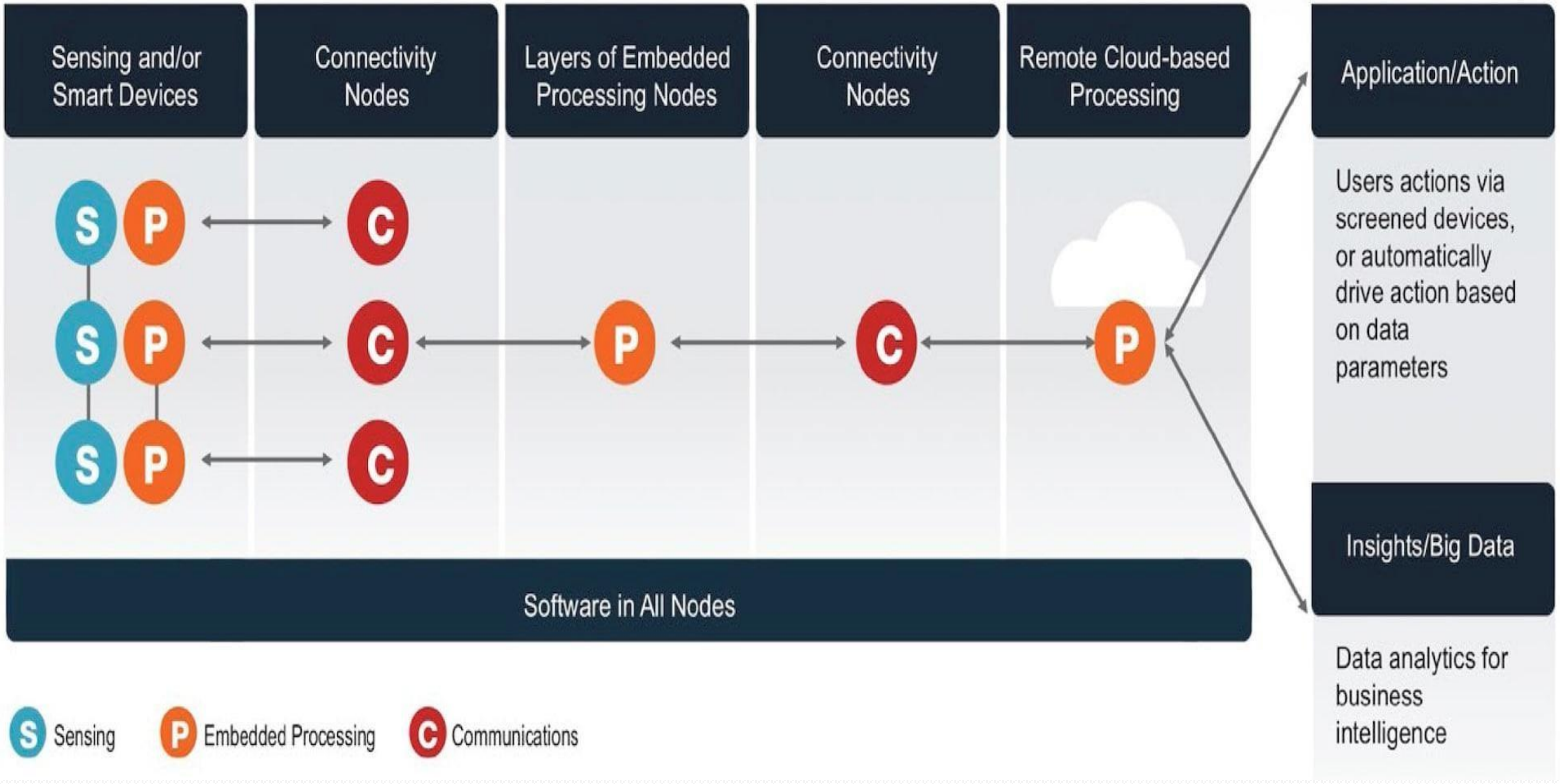
Mirai botnet analysis and detection

- The best way to protect against Mirai Botnet attacks is by ensuring that your IoT devices are secure at all times.
- This means regularly updating firmware on any connected device, changing default passwords, disabling remote access if not needed, keeping your network firewall up-to-date, regularly monitoring for suspicious activity and avoiding public Wi-Fi networks whenever possible.
- It's also important to note that many IoT manufacturers now offer security solutions specifically designed for their products, so it's worth researching what type of protection your connected devices offer before purchasing them.
- Finally, if you suspect your device has already been compromised by a Mirai attack, you should immediately disconnect it from your home network until you can confirm its safety.

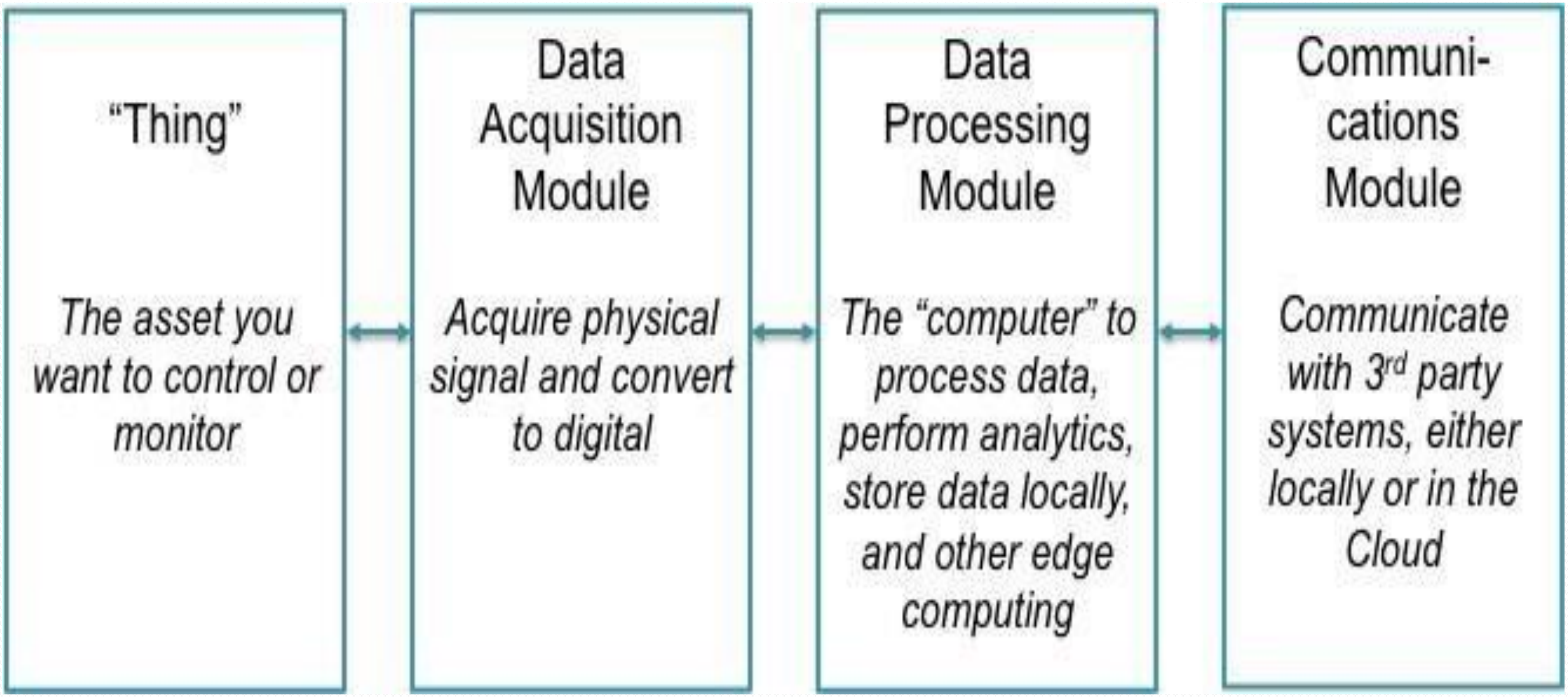
IOT Application Architecture


IOT application architecture.
7
Collaboration and Processes
(Involving People and Business Processes)
CENTER
6
Application
(Reporting, Analytics, Control)
5
Data Abstraction
(Aggregation and Access)
4
Data Accumulation
(Storage)
3
Edge Computing
(Data Element Analysis and Transformation)
2
Connectivity
(Communication and Processing Units)
1
Physical Devices and Controllers
(The "Things" in IoT)
EDGE: Sensors, Devices, Machines, Intelligent Edge Nodes of all types

Technical Building blocks of IoT


Sensing and/or Smart Devices
Connectivity Nodes
Layers of Embedded Processing Nodes
Connectivity Nodes
Remote Cloud-based Processing
Application/Action
Users actions via screened devices, or automatically drive action based on data parameters
Insights/Big Data
Data analytics for business intelligence
S
P
C
Software in All Nodes
S Sensing
P Embedded Processing
C Communications

Technical Building blocks of IOT



"Thing"

The asset you want to control or monitor

Data Acquisition Module

Acquire physical signal and convert to digital

Data Processing Module

The "computer" to process data, perform analytics, store data locally, and other edge computing

Communi-cations Module

Communicate with 3rd party systems, either locally or in the Cloud

Technical Building Blocks of IoT

❖Thing:

- “**Thing**”in IoT is the asset that you want to control or monitor or measure, that is, observe closely.
- In many IoT products, the “**thing**” gets fully incorporated into a smart device.
- For example, products like a smart refrigerator or an

  Automatic vehicle. These products control and monitor themselves.
- There are sometimes many other applications where the thing stands as an alone device, and a separate product is connected to ensure it possesses smart capabilities.

Technical Building blocks of IoT

❖ Data Acquisition Module

- Focuses on acquiring physical signals from the thing which is being observed or monitored and converting them into digital signals that can be manipulated or interpreted by a computer.
- Hardware component of an IoT system that contains all the sensors that help in acquiring real-world signals such as temperature, pressure, density, motion, light, vibration, etc. The type and number of sensors you need depend on your application.
- Also includes the necessary hardware to convert the incoming sensor signal into digital information for the computer to use it. This includes conditioning of incoming signal, removing noise, analog-to-digital conversion, interpretation, and scaling.

Technical Building blocks of IoT

❖**Data Processing Module**

➢The third building block of the IoT device is the data processing module. This is the actual ―computer‖ and the main unit that processes the data performs operations such as local analytics, stores data locally, and performs some other computing operations.

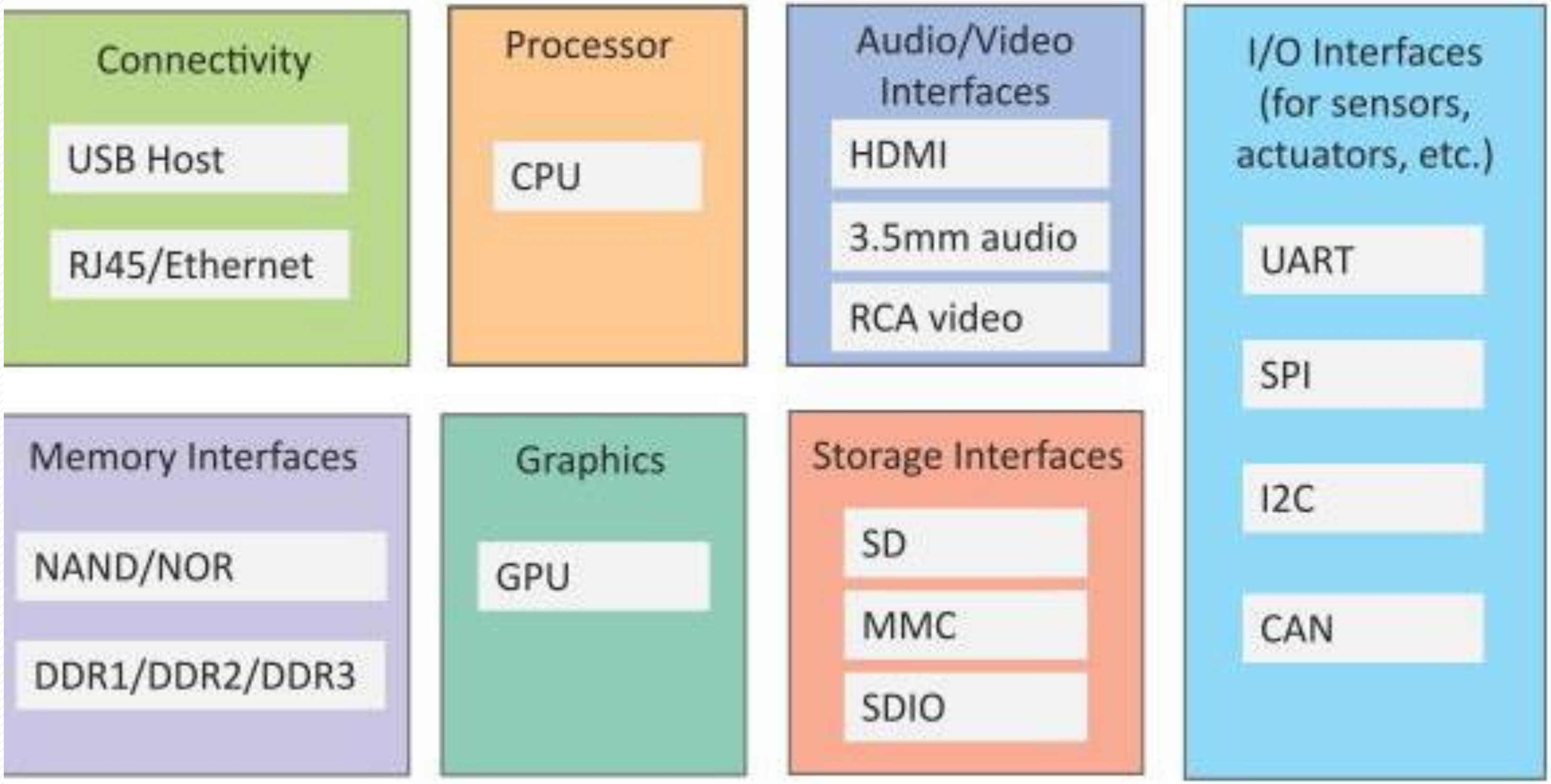
❖**Communication Module**

➢The last building block of IoT hardware is the communications module. This is the part that enables communications with your Cloud Platform, and with 3rd party systems either locally or in the Cloud.

Physical Design of IoT

- The "Things" in IoT usually refers to IoT devices which have unique identities and can perform remote sensing, actuating and monitoring capabilities.
- IoT devices can:
  - Exchange data with other connected devices and applications (directly or indirectly), or
  - Collect data from other devices and process the data locally or
  - Send the data to centralized servers or cloud-based application back-ends for processing the data, or
  - Perform some tasks locally and other tasks within the IoT infrastructure, based on temporal and space constraint

Generic block diagram of An IoT Device



Generic block diagram of An IoT Device

An IoT device may consist of

- several interfaces for connections to other devices, both wired and wireless.
  - I/O interfaces for sensors
  - Interfaces for Internet connectivity
  - Memory and storage interfaces
  - Audio/video interfaces.

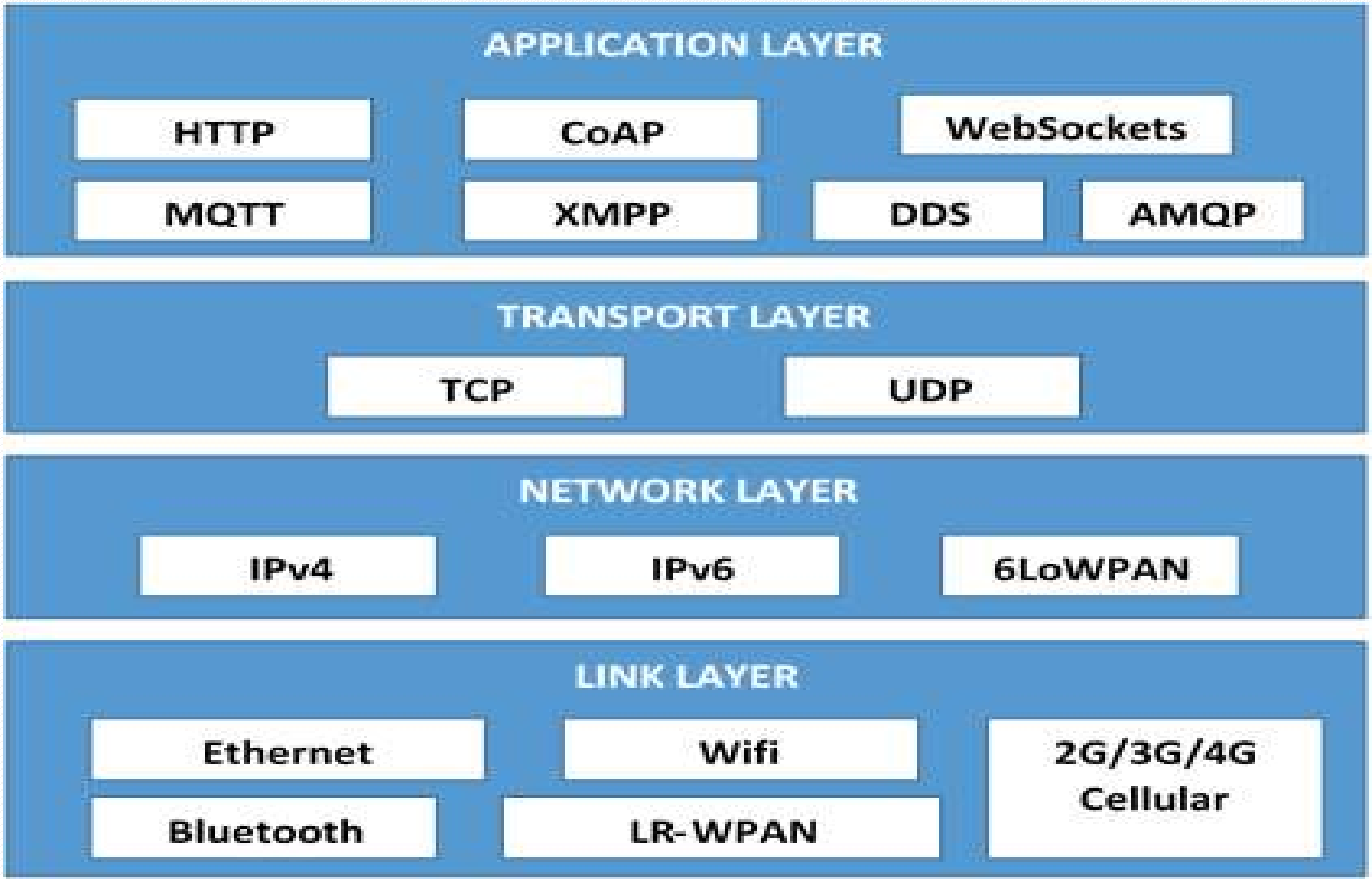
Things in IoT

- An IoT Device can collect various types of data from the onboard or attached sensors, such as temperature, humidity, light intensity.
- IoT devices can also be varied types, for instance, wearable sensors, smart watches, LED light automobiles and industrial machines.
- Generate data in Some form or the other which when processed by Data Analytics systems leads to useful information to guide further actions locally or remotely.

Points Covered

- Introduction to Internet of Things (IoT):
  - ✓ Definition
  - ✓ Characteristics of IoT
  - ✓ Vision
  - ✓ Trends in adaption of IoT
  - ✓ IoT Devices
  - ✓ IoT Devices Vs Computers
  - ✓ Societal Benefits of IoT
  - ✓ Technical Building Blocks.
- Physical Design of IoT:
  - ✓ Things in IoT

IoT Protocols



APPLICATION LAYER

HTTP

CoAP

WebSockets

MQTT

XMPP

DDS

AMQP

TRANSPORT LAYER

TCP

UDP

NETWORK LAYER

IPv4

IPv6

6LoWPAN

LINK LAYER

Ethernet

Wifi

2G/3G/4G Cellular

Bluetooth

LR-WPAN

Link Layer

- Link Layer protocols determine how the data is physically sent over the network's physical layer or medium(example copper wire, electrical cable, or radio wave).
- The scope of the link layer is the local network connections to which host is attached.
- Host on the same link exchange data packets over the link layer using the link layer protocols.
- Link layer determines how the packets are coded and signaled by the hardware device over the medium to which the host is attached (such as a coaxial cable).

802.3 – Ethernet

- 802.3 is a collections of wired Ethernet standards for the link layer.
- For example,
  - 802.3 10BASE5 Ethernet that uses coaxial cable as a shared medium,
  - 802.3.i is standard for 10 BASET Ethernet over copper twisted pair connection,
- Standards provide data rates from 10 Mb/s to 40 gigabits per second and the higher.
- The shared medium in Ethernet can be a coaxial cable , twisted pair wire or and Optical fiber.
- Shared medium carries the communication for all the devices on the network.

Ethernet 802.3

- Ethernet is a set of technologies and protocols that are used primarily in LANs.
- It was first standardized in 1980s by IEEE 802.3 standard.
- IEEE 802.3 defines the physical layer and the medium access control (MAC) sub-layer of the data link layer for wired Ethernet networks.
- Ethernet is classified into two categories: classic Ethernet and switched Ethernet.

Classic Ethernet

- Classic Ethernet is the original form of Ethernet that provides data rates between 3 to 10 Mbps.
- The varieties are commonly referred as 10BASE-X.
- Here, 10 is the maximum throughput, i.e. 10 Mbps, BASE denoted use of baseband transmission, and X is the type of medium used.
- Most varieties of classic Ethernet have become obsolete in present communication scenario.

Ethernet 802.3 10BASE5

- The 10 refers to its transmission speed of 10 Mbit/s.
- The BASE is short for baseband signaling (as opposed to broadband[a]), and
- the 5 stands for the maximum segment length of 500 meters

Switched Ethernet

- A switched Ethernet uses switches to connect to the stations in the LAN.
- It replaces the repeaters used in classic Ethernet and allows full bandwidth utilization.

- There are a number of versions of IEEE 802.3 protocol. The most popular ones are -

1. IEEE 802.3: This was the original standard given for 10BASE-5. It used a thick single coaxial cable into which a connection can be tapped by drilling into the cable to the core. Here, 10 is the maximum throughput, i.e. 10 Mbps, BASE denoted use of baseband transmission, and 5 refers to the maximum segment length of 500m.
2. IEEE 802.3a: This gave the standard for thin coax (10BASE-2), which is a thinner variety where the segments of coaxial cables are connected by BNC connectors. The 2 refers to the maximum segment length of about 200m (185m to be precise).

3. IEEE 802.3i: This gave the standard for twisted pair (10BASE-T) that uses unshielded twisted pair (UTP) copper wires as physical layer medium. The further variations were given by IEEE 802.3u for 100BASE-TX, 100BASE-T4 and 100BASE-FX.

4. IEEE 802.3i: This gave the standard for Ethernet over Fiber (10BASE-F) that uses fiber optic cables as medium of transmission.

802.1- WI-FI

- IEEE 802.3 is a collections of wireless Local area network (WLAN) communication standards, including extensive descriptions of the link layer.
- For example,
  - 802.11a operate in the 5 GHz band,
  - 802.11b and 802.11g operate in the 2.4 GHz band.
  - 802.11ac operates in the 5 GHz band.
  - 802.11ad operates in the 60G hertz band
- These standards provide data rates from 1 Mbps to upto 6.75Gbps.

802.16 wiMAX

- IEEE 802.16 is a collection of wirless broadband and Standards, including extensive descriptions for the link layer also called WiMAX.
- Wimax standard provides a data rates from from 1.5 Mbps to 1Gbps.
- The recent update provides data rates of hundred Mbps for mobile station.

802.15.4 LR-WPAN

- IEEE 802.15.4 is a collections of standard for low rate wireless personal area network(LR-WPAN).
- These standard form the basis of specifications for high level communication Zigbee.
- LR-WPAN standards provide data rates from 40 kbps 200kbps.
- These standards provide low cost and low speed Communications for power constrained devices.

2G/3G/4G mobile communications

- These are the different generations of mobile communication standards including second generation (2G including GSM and CDMA).
- 3rd Generation (3G including UMTS and CDMA2000) and 4th generation 4G including LTE.
- IoT devices based on these standards communicate over cellular networks.
- Data rates for these standards range from 9 Kbps [2G] to 100 Mbps [4G].

Network / internet layer

- The network layer are responsible for sending of IP datagrams from the source network to the destination network.
- This layer performs the host addressing and packet routing.
- The datagrams contains a source and destination address which are used to route them from the source to the destination across multiple networks.
- Host Identification is done using the hierarchy IP addressing schemes such as IPv4 or IPv6.

Network/Internet Layer

- IPv4 : 32-bit address scheme
- IPv6 : 128-bit address scheme
- 6LoWPA :
  - (IPv6 over Low-Power Wireless Personal Area Networks), is a low power wireless mesh network where every node has its own IPv6 address.
  - This allows the node to connect directly with the Internet using open standards.
  - It was created with the intention of applying the Internet Protocol even to the smallest devices, enabling low-power devices with limited processing capabilities to participate in the Internet of Things
  - 2.4 GHz frequency range and provides data transfer rates of 250Kbps
  - Works with 802.15.4link layer protocol

Transport layer

- The Transport layer protocols provides end-to-end message transfer capability independent of the underlying network.
- The message transfer capability can be set up on connections, either using handshake or without handshake acknowledgements.
- Provides functions such as error control , segmentation, flow control and congestion control.

Transport Layer

- TCP
- UDP

TCP

- Transmission Control Protocol (TCP)
  - a communications standard that enables application programs and computing devices to exchange messages over a network
  - It is designed to send packets across the internet and ensure the successful delivery of data and messages over networks.
  - Most widely used to transport layer protocol that is used by the web browsers along with HTTP , HTTPS application layer protocols email program (SMTP application layer protocol) and file transfer protocol
  - A connection Oriented and stateful protocol while IP protocol deals with sending packets, TCP ensures reliable transmissions of packets in order.
  - TCP also provide error detection capability so that duplicate packets can be discarded and low packets are retransmitted.
  - The flow control capability ensures that the rate at which the sender sends the data is not to too to high for the receiver to process.
  - The congestion control capability of TCP helps in avoiding network congestion and congestion collapse which can lead to degradation of network performance.

UDP

- User Datagram Protocol (UDP)
  - a communications protocol for time-sensitive applications like gaming, playing videos, or Domain Name System (DNS) lookups
  - results in speedier communication because it does not spend time forming a firm connection with the destination before transferring the data
  - unlike TCP, which requires carrying out an initial setup procedure, UDP is a connection less protocol
  - useful for time sensitive application that have very small data units to exchange and do not want the overhead of connection setup.
  - UDP is a transactions oriented and stateless protocol.
  - UDP does not provide guaranteed delivery, ordering of messages and duplicate eliminations.

Application layer

- Application layer protocol define how the application interfaces with the lower layer protocols to send the data over the network.
- Data are typically in files, is encoded by the application layer protocol and encapsulated in the transport layer protocol.
- Application layer protocol enable process-to-process connection using ports.

Application Layer

- HTTP
- CoAP
- WebSocket
- MQTT
- XMPP
- DDS
- AMQP

HTTP

- Hypertext transfer protocol is the application layer protocol that forms the foundations of world wide web http includes, ,commands such as GET, PUT, POST, DELETE, HEAD, TRACE, OPTIONS etc.
- The protocol follows a request-response model where are client sends request to server using the http, commands.
- HTTP is a stateless protocol and each http request is independent of other request and http client can be a browser or an application running on the client example and application running on an IoT device, mobile, mobile applications or other software.

HTTP (HyperText Transfer Protocol)

- Hypertext Transfer Protocol is the most widely used protocol for navigating the Internet and to make data available over REST-APIs.
- The main advantage of using HTTP for IoT is that web application developers can use the same mechanism to send data to a Webserver - via an HTTP POST request.
- The drawbacks are that HTTP uses a connectionless request-respond communication, meaning every message needs to include authentication information—which requires data and energy consumption.
- Nevertheless, HTTP might be ideal for use cases which have fewer data and battery constraints and where devices already need to call existing REST-APIs.

CoAP (Constrained Application Protocol)

- Constrained Application Protocol (CoAP) is designed for low-power, lossy networks, also known as “constrained” networks.
- CoAP is usually paired with User Datagram Protocol (UDP) which makes it highly efficient, making it appealing for IoT applications where battery conservation is important.
- For example, it’s often used in smart meter communications.
- CoAP can also use TCP or SMS as transport mechanism.

WebSocket

- WebSocket is a bi-directional communication protocol designed to quickly send large quantities of data in web applications.
- A WebSocket establishes a connection between client and server and therefore after the initial connection establishment—every single message only has small overhead.
- Devices and servers can simultaneously transmit and receive data in real-time, making this protocol best-suited for IoT applications where low latency is critical, communication happens frequently, and data consumption is less important.

MQTT (Message Queueing Telemetry Transport)

- MQTT is a lightweight communication protocol specifically designed for IoT and M2M applications.
- It’s ideal for remote environments or applications with limited bandwidth.
- MQTT uses a connection oriented publish/subscribe architecture, where MQTT applications can either publish (transmit) or subscribe to (receive) topics, and an MQTT broker passes information from the publishing client to the subscribed client.
- So for instance, an oil rig may have a predictive maintenance sensor that detects changes in vibrations and uses the MQTT protocol.
- The sensor “publishes” the vibration level to the broker, which the MQTT broker then passes to a software application that has subscribed to the topic “vibrations,” which can then trigger an alert when the level is above a threshold.

DDS [Data distribution service]

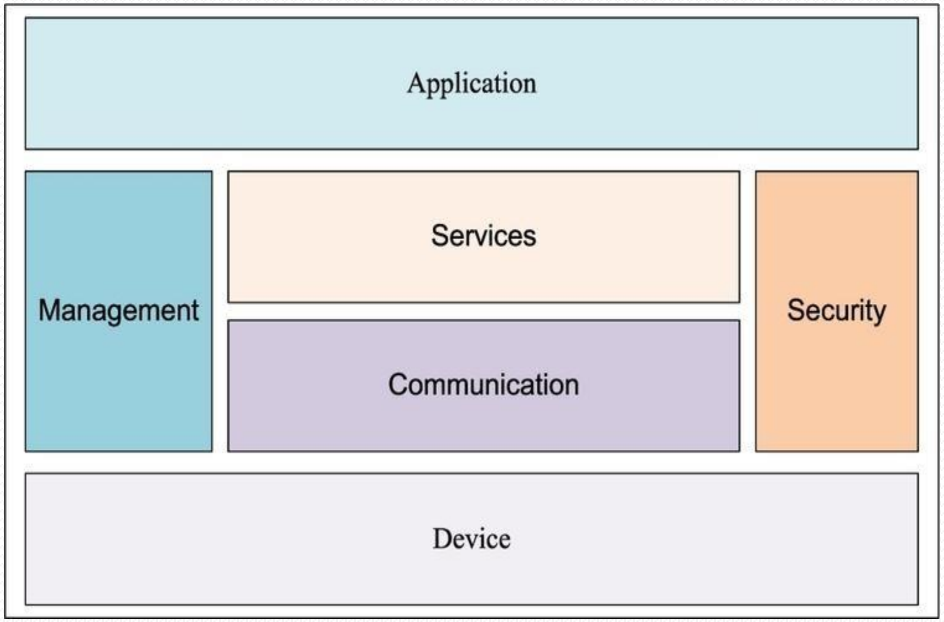
- DDS is the date centric middleware standard for device-to-device machine to machine communication .
- DDS uses a publish subscribe model where publisher example device that generate data create topics to which subscribers per can subscribe publisher is an object responsible for data distributions and the subscriber responsible for receiving published data.
- DDS provide quality of service (QoS) control and configurable reliability

AMQP: Advanced Message Queuing protocols

- it is an open application layer protocol for business messaging.
- AMQP support point to point and publish - subscribe model routing and queuing.
- AMQP broker receive message from publishers example devices or applications that generate data and about them over connections to consumers publishers publish the message to exchange which then distribute message copies to queues.

Logical Design of IoT

- Logical design of an IoT system refers to an abstract representation of the entities and processes without going into the low-level specifics of the implementation.
- An IoT system comprises a number of functional blocks that provide the system the capabilities for identification, sensing, actuation, communication and management.



Logical Design of IoT

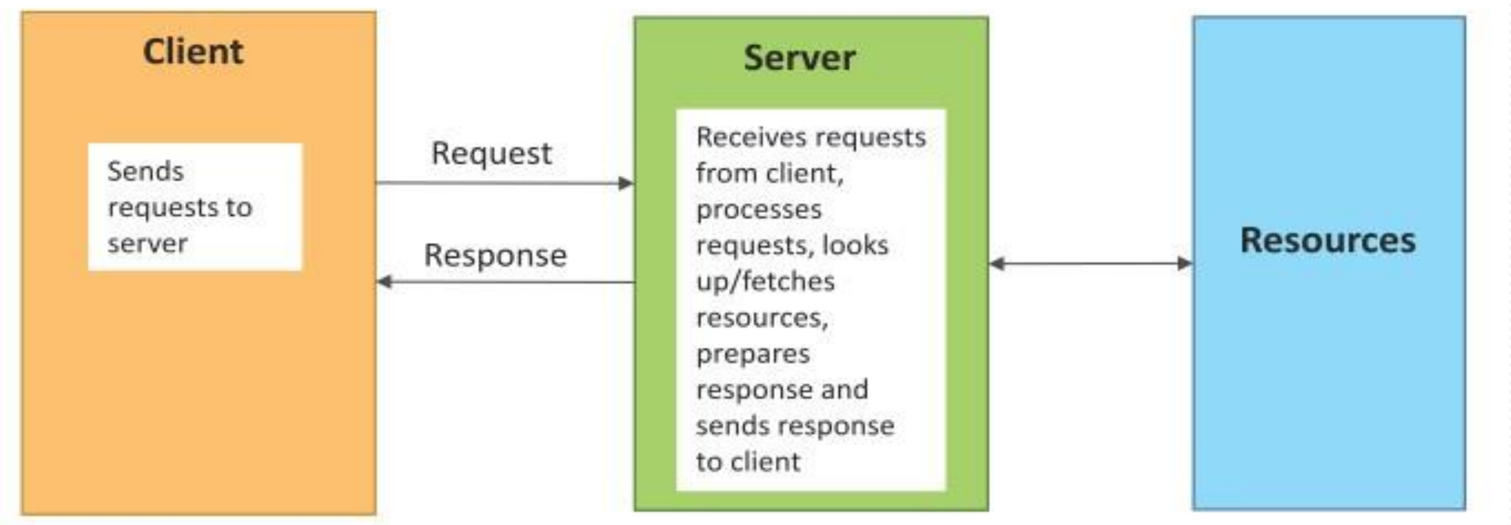
- **Device :** An IoT system comprises of the devices such as sensing, actuation, monitoring and control functions.
- **Communication :** communication block handle the communication systems
- **Services :** An IoT system uses various types of IoT Protocols Services like device monitoring, device control services, data publishing services and device discovery
- **Management :** Functional blocks provide various functions to govern the IoT system
- **Security :** Security functional block secures IoT system by providing functions such as application authorization message and content integrity and data security.
- **Applications:** IoT application provides and interface that the user can used to control and monitor various aspects of the IoT system. Application also allow users to view the system status and view or analyze the processed to data.

IoT Communication Models

1. Request–Response Communication Model
2. Publish–Subscribe Communication Model
3. Push–Pull Communication Model
4. Exclusive Pair Communication Model

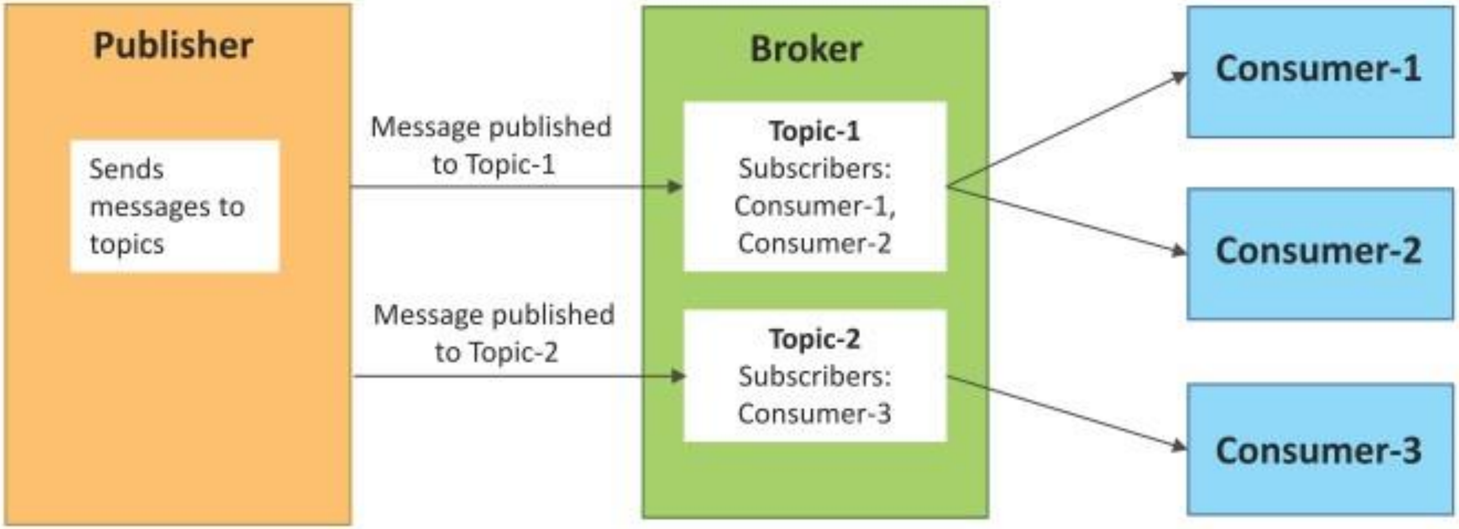
Request–Response Communication Model

- Request–Response is a communication model in which the client sends requests to the server and the server responds to the requests.
- When the server receives a request, it decides how to respond, fetches the data, retrieves resource representations, prepares the response and then sends the response to the client.
- Stateless communication model
- Each request –response pair is independent of others


Client
Sends requests to server
Request
Response
Server
Receives requests from client, processes requests, looks up/fetches resources, prepares response and sends response to client
Resources

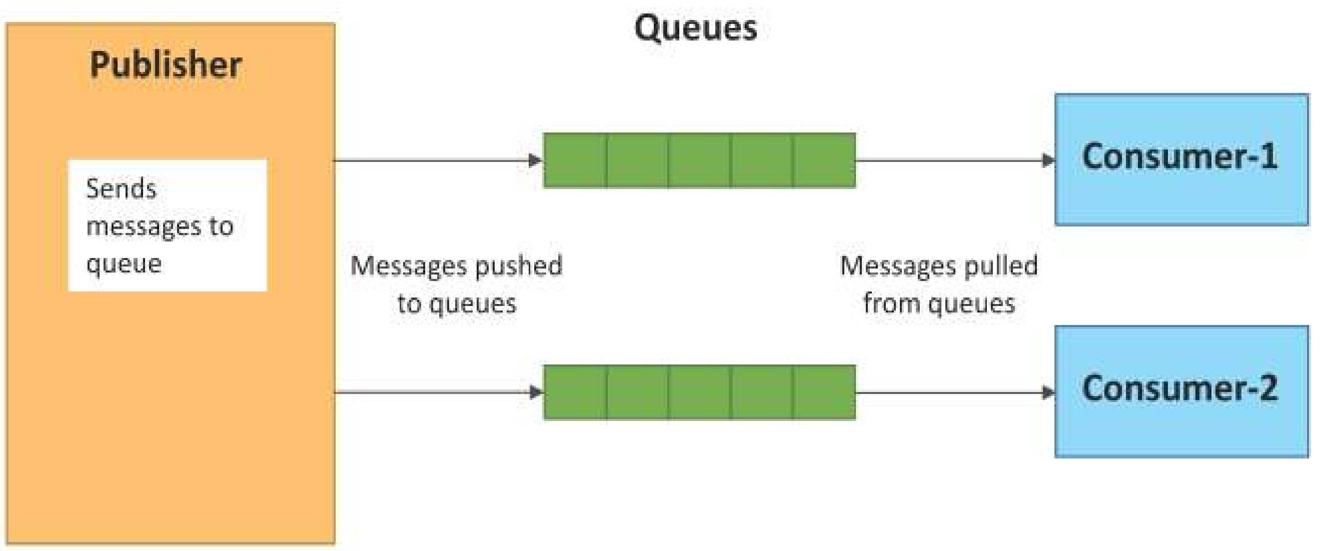
Publish–Subscribe Communication Model

- 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.



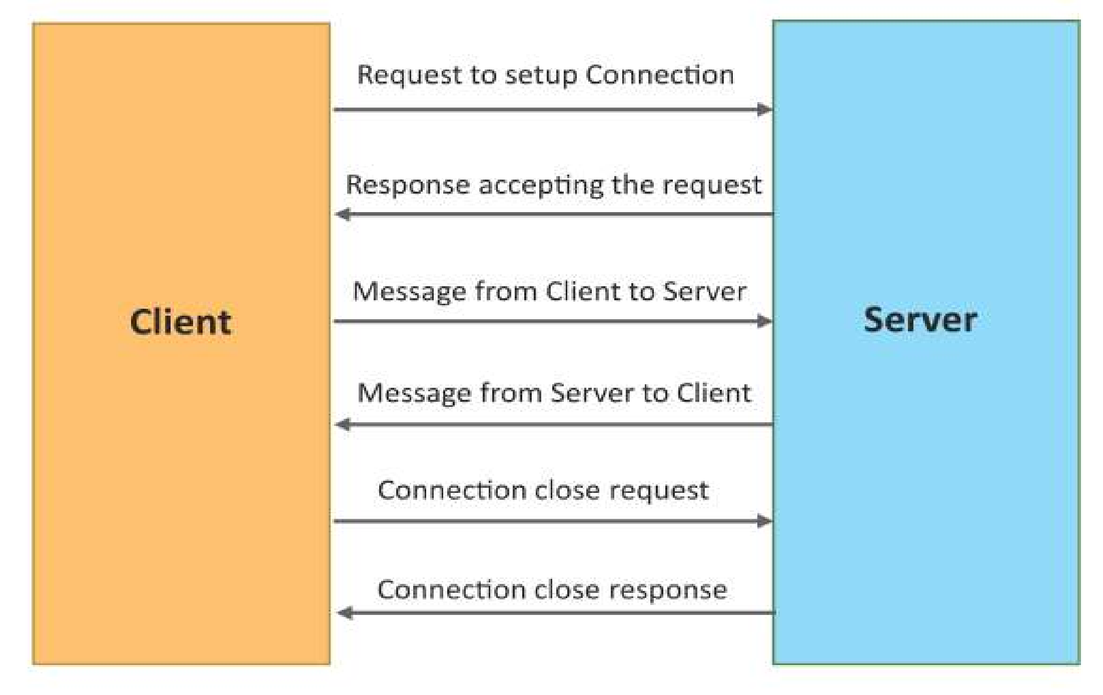
Push–Pull Communication Model

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



Exclusive Pair Communication Model

- Exclusive Pair is a bidirectional, fully duplex communication model that uses a persistent connection between the client and the server.
- Once the connection is set up it, remains open until the client sends a request to close the connection.
- Client and server can send messages to each other after connection setup.

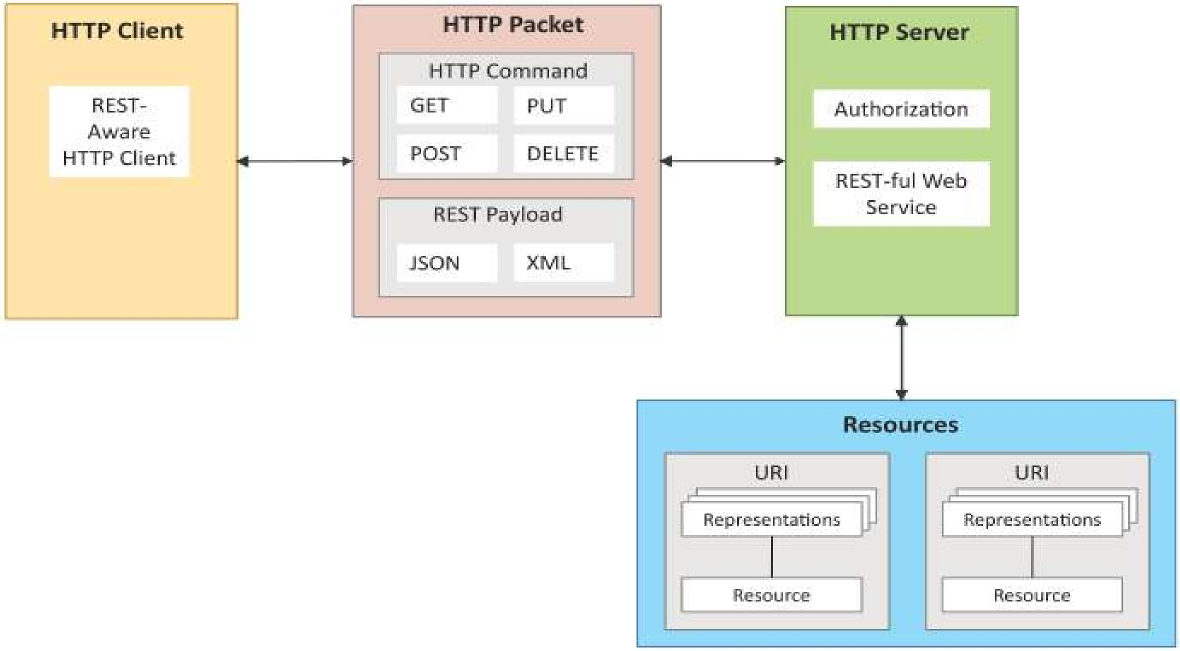

Client
Request to setup Connection
Response accepting the request
Message from Client to Server
Message from Server to Client
Connection close request
Connection close response
Server

IoT communication APIs

- REST- based communication API:

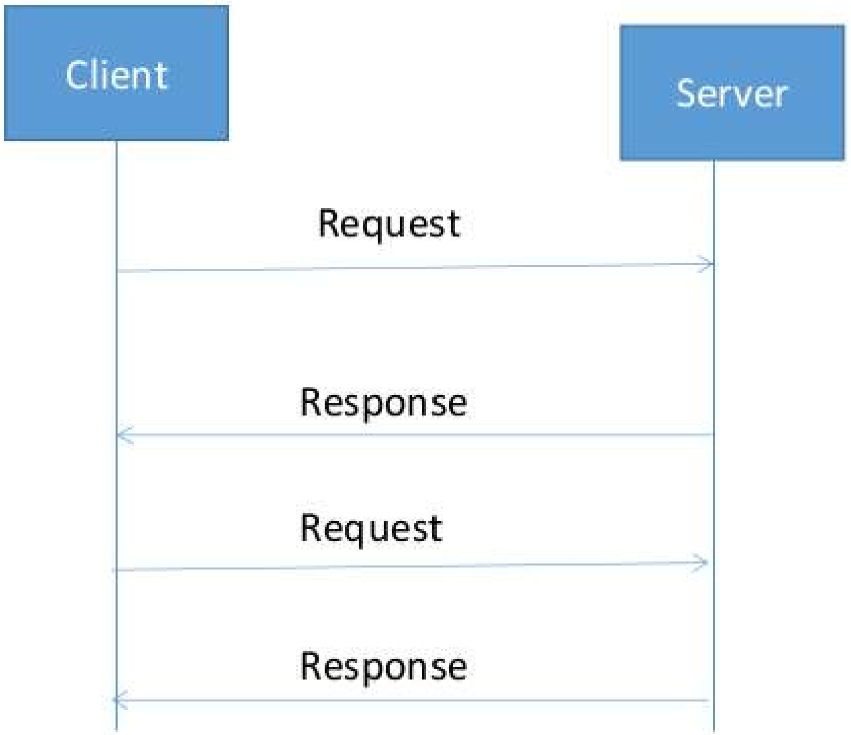
REST-based Communication APIs

- Representational State Transfer (REST) is a set of architectural principles by which you can design web services and web APIs that focus on a system's resources and how resource states are addressed and transferred.
- REST APIs follow the request–response communication model.
- REST architectural constraints apply to the components, connectors and data elements within a distributed hypermedia system.



REST-based Communication APIs Constraints

- Client – Server
- Stateless
- Cache-able
- Layered System
- Uniform Interface
- Code on demand



1. **Client server**: The principle behind the client-server conference separations of concerns. For example, client should not be concerned with the storage of data which is their concern of the server. Similarly, the server should not be concerned about the user interface which is a concern of the client. Separation allows client and server to be independently deployed and updated.
2. **Stateless**: Each request from client to server must contain all the information necessary to understand the request , and cannot take advantage of any stored context on the server .

3. **Cache-able**: Cache constraint requires that the data within a response to a request be implicitly or explicitly labeled as cache-able or non-cache-able. If a response is cache-able, then a client cache is given the right to reuse that response data for later, equivalent requests. Caching can partially or completely eliminate some interactions and improve efficiency and scalability.
4. **Layered system**: Layered system constraint, constraints the behavior of components such that each component cannot see beyond the immediate layer with which they are interacting. Example client cannot tell whether it is connected directly to the end server or to an intermediary along the way. System scalability can be improved allowing intermediaries to respond to requests instead of the end server, without the client having to do anything different.

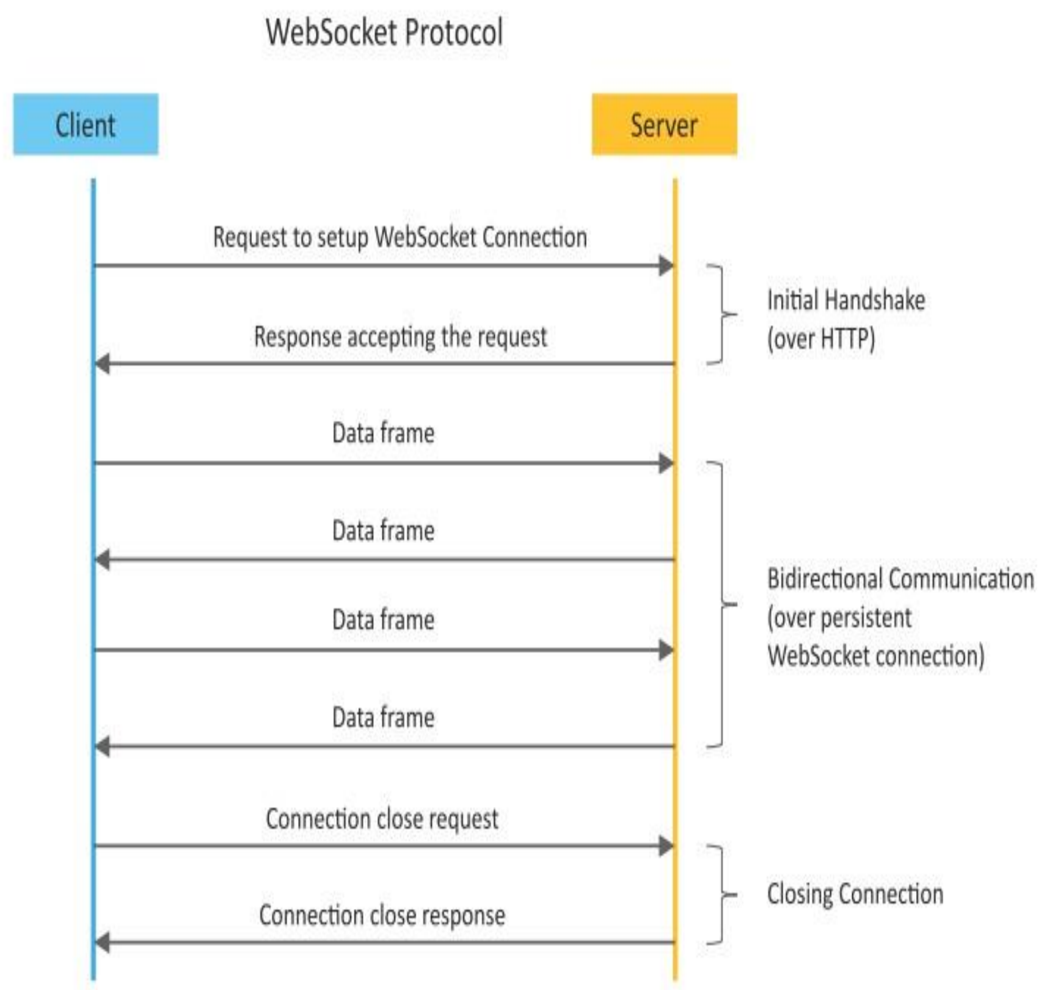
5. **Uniform interface**: Uniform interface constraints requires that the method of communication between client and server must be uniform. Resources are identified in the request (by URIs in web based systems) and are themselves separate from the representations of the resource that are returned to the client. When climbing holds a representation of a resource it has all the information required to update or delete the resource (provided the client has required permissions). Each message includes enough information to describe how to process the message.
6. **Code on demand** : Server can provide executable code script for clients to execute in their context. This is the only optional constraint.

Revise : WebSocket Protocol

- WebSocket is a bi-directional communication protocol designed to quickly send large quantities of data in web applications.
- A WebSocket establishes a connection between client and server and therefore after the initial connection establishment—every single message only has small overhead.
- Devices and servers can simultaneously transmit and receive data in real-time, making this protocol best-suited for IoT applications where low latency is critical, communication happens frequently, and data consumption is less important.

WebSocket-based Communication APIs

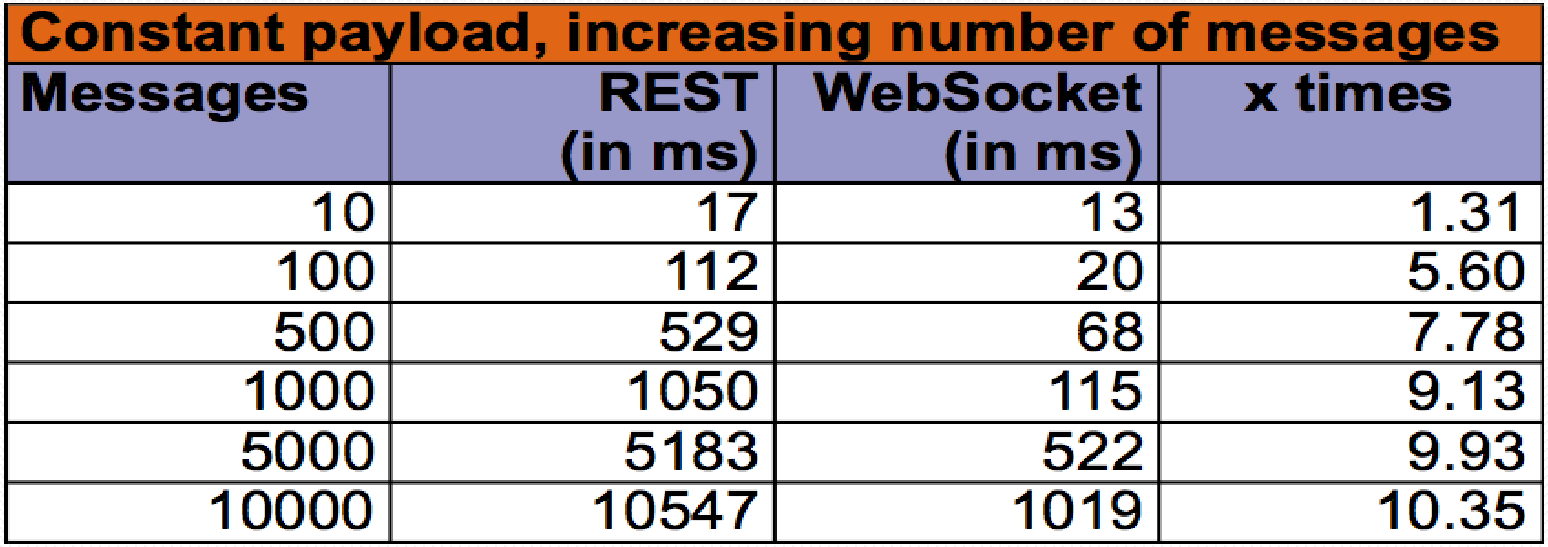
- WebSocket APIs allow bi-directional, full duplex communication between clients and servers.
- WebSocket APIs follow the exclusive pair communication model.
- Web Socket is a low-level protocol.


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

Difference between REST and WebSocket-based service

Parameter	REST	WebSocket
State	Stateless	Stateful
Directional	Unidirectional	Bidirectional
Req-Res/Full Duplex	Follow Request Response Model	Exclusive Pair Model
TCP Connections	Each HTTP request involves setting up a new TCP Connection	Involves a single TCP Connection for all request
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

Difference between REST and WebSocket-based



Constant payload, increasing number of messages			
Messages	REST (in ms)	WebSocket (in ms)	x times
10	17	13	1.31
100	112	20	5.60
500	529	68	7.78
1000	1050	115	9.13
5000	5183	522	9.93
10000	10547	1019	10.35

- This table shows that the REST overhead increases with the number of messages.
- This is true because that many TCP connections need to be initiated and terminated and that many HTTP headers need to be sent and received.
- The last column particularly shows the multiplication factor for the amount of time to fulfill a REST request.

IoT Enabling Technologies

- **Wireless Sensor Network**
  - ✓ Weather monitoring system
  - ✓ Indoor Air quality monitoring system
  - ✓ Soil moisture monitoring system
  - ✓ Survelliance systems
  - ✓ Health monitoring systems
  - ✓ **Protocols-Zigbee**
- **Cloud Computing**
- **Big Data Analytics**
- **Embedded Systems**

1. Wireless Sensor Network(WSN)

- A WSN comprises distributed devices with sensors which are used to monitor the environmental and physical conditions.
- A wireless sensor network consists of end nodes, routers and coordinators.
- End nodes have several sensors attached to them where the data is passed to a coordinator with the help of routers.
- The coordinator collects the data from all the nodes.
- The coordinator also acts as the gateway that connects WSN to the internet.
- Example :
  - Weather monitoring system
  - Indoor air quality monitoring system
  - Soil moisture monitoring system
  - Surveillance system
  - Health monitoring system
- WSNs are enabled by wireless communication protocols such as IEEE 802.15.4.
- ZigBee is one of the most popular wireless technologies used by WSNs.

2. Cloud Computing

- It provides us the means by which we can access applications as utilities over the internet.
- Cloud means something which is present in remote locations.
- With Cloud computing, users can access any resources from anywhere like databases, webservers, storage, any device, and any software over the internet.
- Characteristics:
    1. Broad network access
    2. On demand self-services
    3. Rapid scalability
    4. Measured service
    5. Pay-per-use
- Provides different services, such as – IaaS, PaaS, SaaS

Cloud Services

- **IaaS** (Infrastructure as a service)
  Infrastructure as a service provides online services such as physical machines, virtual machines, servers, networking, storage and data center space on a pay per use basis. Major IaaS providers are Google Compute Engine, Amazon Web Services and Microsoft Azure etc. Ex : Web Hosting, Virtual Machine etc.
- **PaaS** (Platform as a service)
  Provides a cloud-based environment with a very thing required to support the complete life cycle of building and delivering West web based (cloud) applications – without the cost and complexity of buying and managing underlying hardware, software provisioning and hosting. Computing platforms such as hardware, operating systems and libraries etc. Basically, it provides a platform to develop applications. Ex : App Cloud, Google app engine
- **SaaS** (Software as a service)
  It is a way of delivering applications over the internet as a service. Instead of installing and maintaining software, you simply access it via the internet, freeing yourself from complex software and hardware management.
  SaaS Applications are sometimes called web-based software on demand software or hosted software. SaaS applications run on a SaaS provider's service and they manage security availability and performance. Ex : Google Docs, Gmail, office etc.

3. Big Data Analytics

- It refers to the method of studying massive volumes of data or big data.
- Collection of data whose volume, velocity or variety is simply too massive and tough to store, control, process and examine the data using traditional databases.
- Big data is gathered from a variety of sources including social network videos, digital images, sensors and sales transaction records.
- Several steps involved in analyzing big data –
    1. Data cleaning
    2. Munging / wrangling
    3. Processing
    4. Visualization
- Examples :
    - Bank transactions
    - Data generated by IoT systems for location and tracking of vehicles
    - E-commerce [Big-Basket]
    - Health and fitness data generated by IoT system such as a fitness bands

4. Communications Protocols

- They are the backbone of IoT systems and enable network connectivity and linking to applications.
- Communication protocols allow devices to exchange data over the network.
- Multiple protocols often describe different aspects of a single communication.
- A group of protocols designed to work together is known as a protocol suite; when implemented in software they are a protocol stack.
- They are used in
  - Data encoding
  - Addressing schemes

5. Embedded Systems

- It is a combination of hardware and software used to perform special tasks.
- It includes microcontroller and microprocessor memory, networking units (Ethernet Wi-Fi adapters), input output units (display keyword etc.) and storage devices (flash memory).
- It collects the data and sends it to the internet.
- Embedded systems used in
  - DVD player, music player
  - Industrial robots
  - Wireless Routers etc.
  - Digital camera

IoT Levels and Deployment Templates

An IoT system comprises of the following components:

- Device
- Resource
- Controller Service
- Database
- Web Service
- Analysis Component
- Application

IoT Levels and Deployment Templates

An IoT system comprises of the following components:

- **Device:** An IoT device allows identification, remote sensing, actuating and remote monitoring capabilities.

- **Resource:** Resources are software components on the IoT device for accessing, processing and storing sensor information, or for controlling actuators connected to the device. Resources also include the software components that enable network access for the device.

- **Controller Service**: Controller service is a native service that runs on the device and interacts with the web services. Controller service sends data from the device to the web service and receives commands from the application (via web services) for controlling the device.

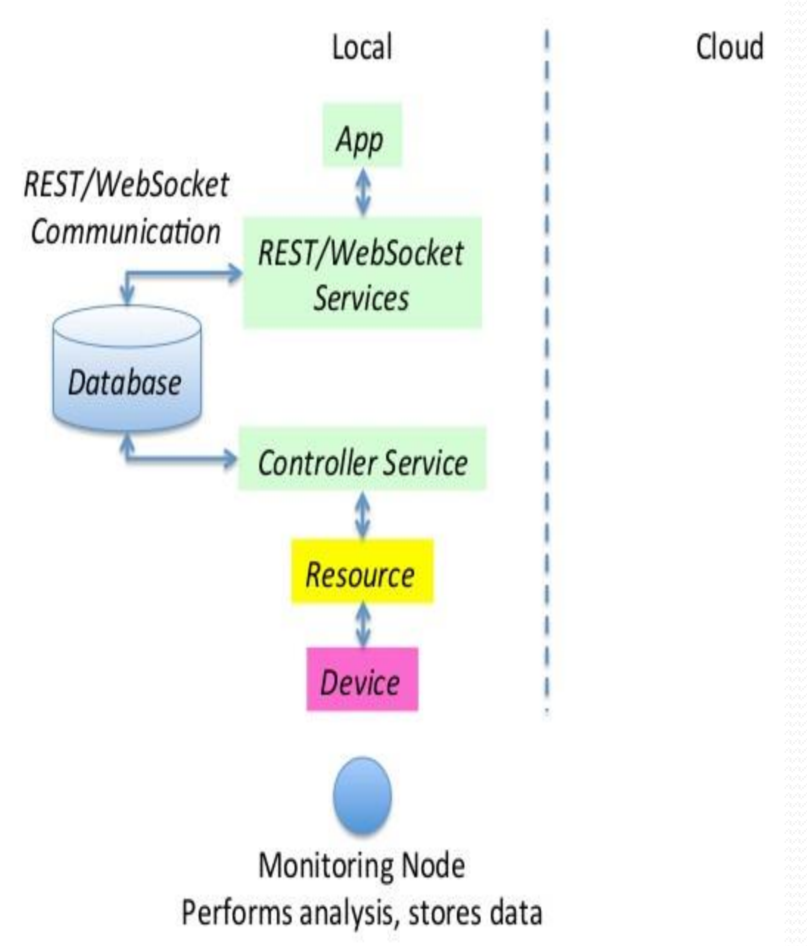
IoT Levels and Deployment Templates

An IoT system also comprises of the following components:

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

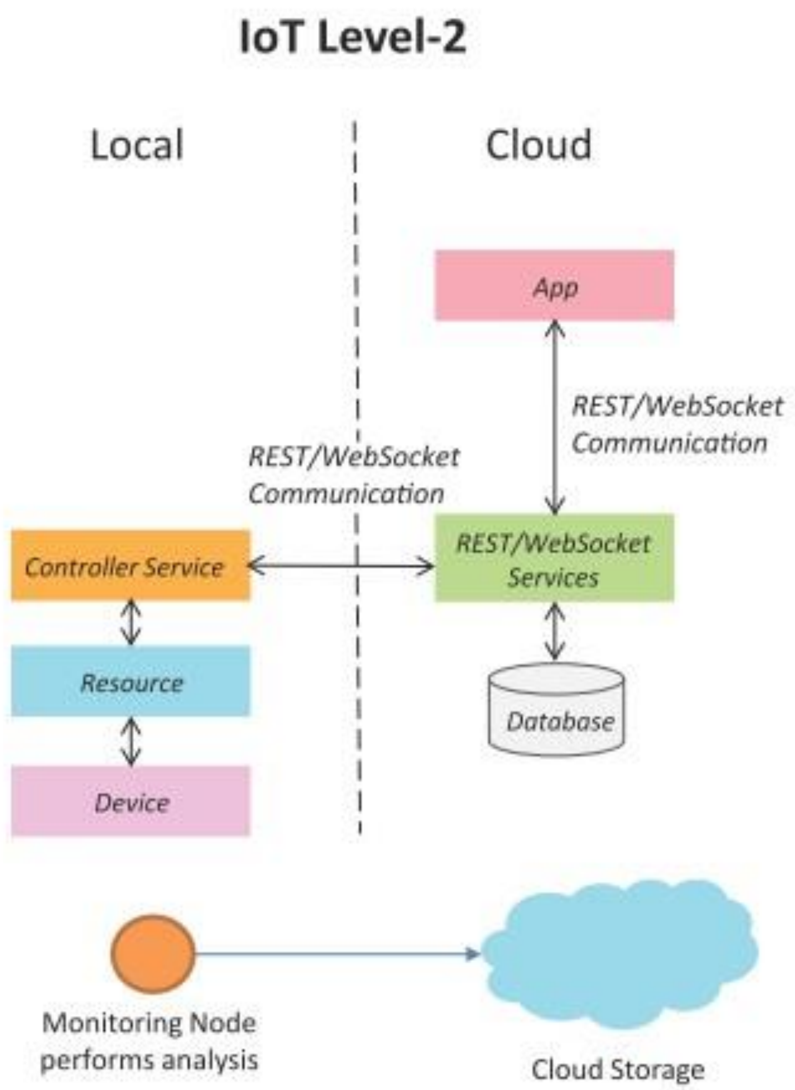
IoT Level-1 (Home Automation)

- **A level-1 IoT system has a** **single node / device** that performs sensing and / or actuation, stores data, and performs analysis and hosts the application.
- Level-1 IoT systems are suitable for **modelling low- cost and low-complexity solutions** where the **data involved is not big** and the analysis requirements are not computationally intensive.



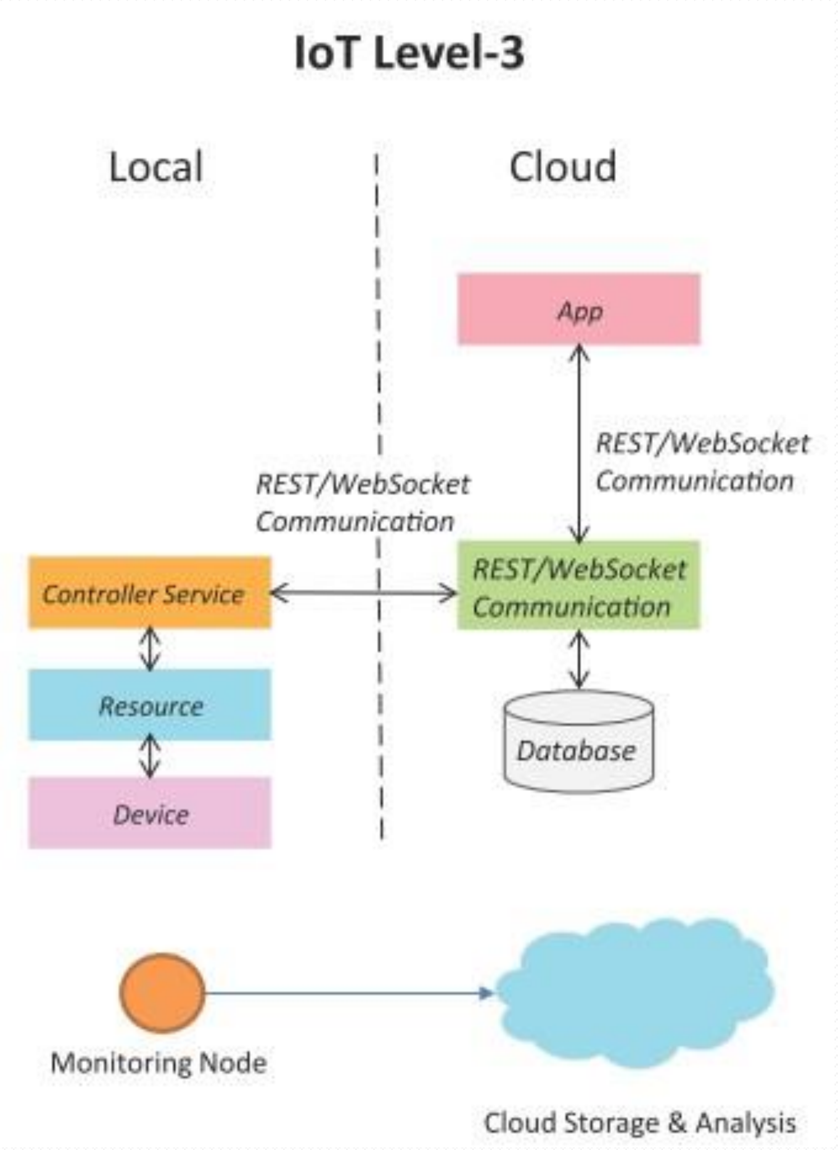
IoT Level-2 (Smart Irrigation)

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



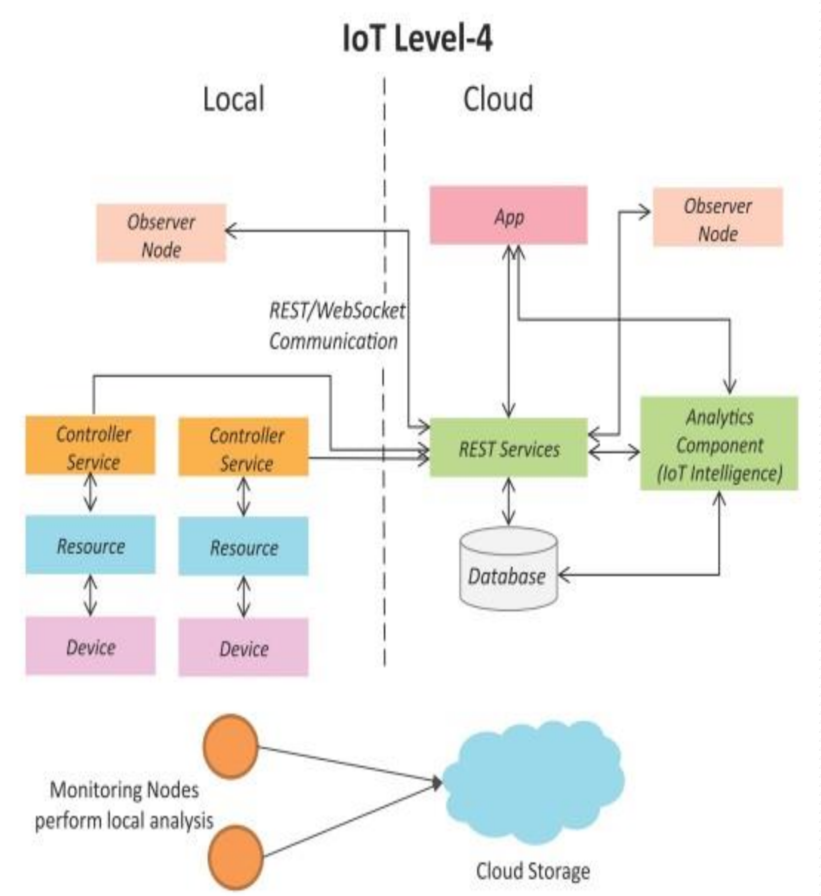
IoT Level-3 (Vibration Monitoring)

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



IoT Level-4 (Noise Monitoring)

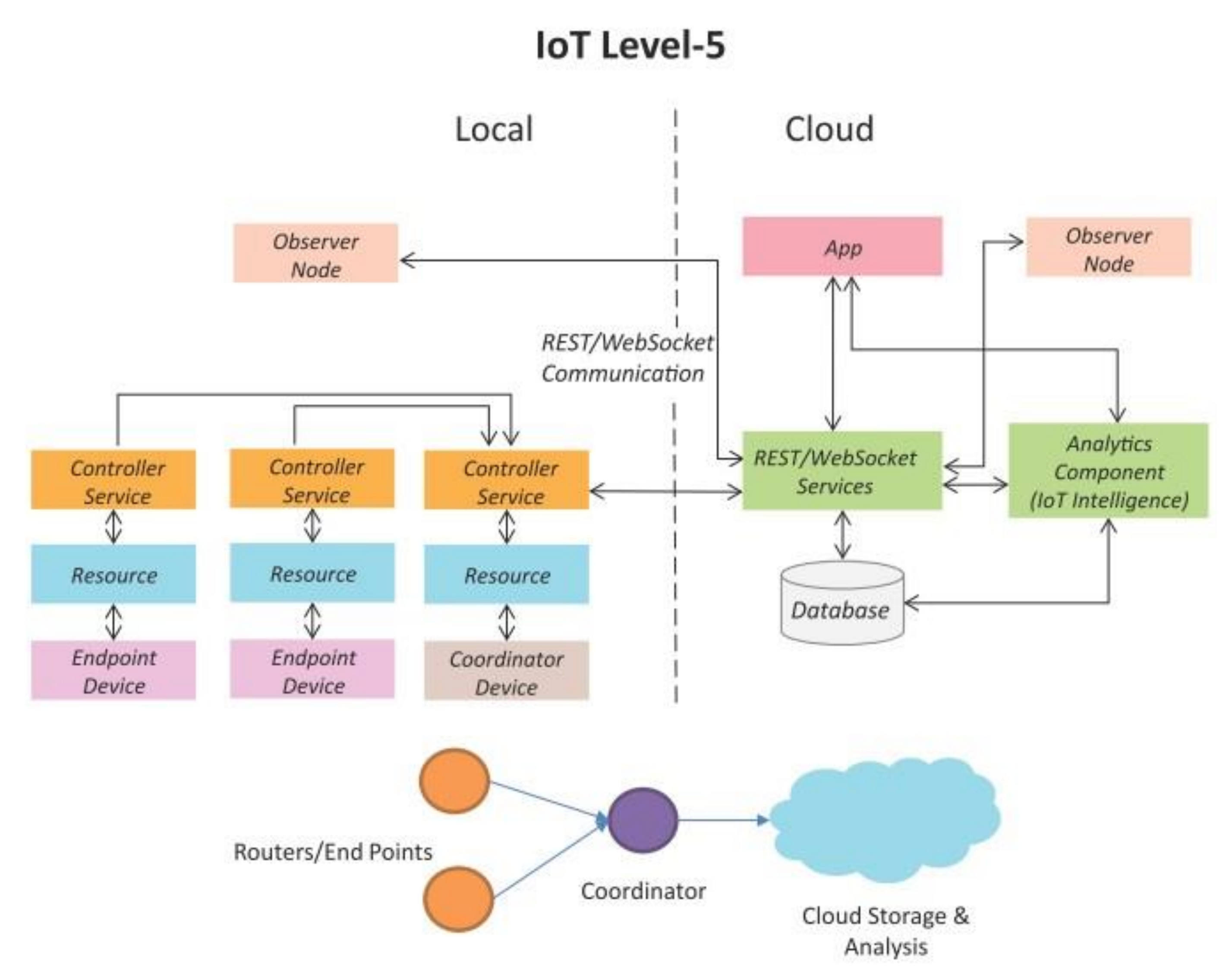
- A level-4 IoT system has **multiple nodes** that perform **local analysis**.
- Data is stored in the cloud and the application is **cloud-based.**
- Level-4 contains local and cloud- based observer nodes which can subscribe to and receive information collected in the cloud from IoT devices.
- Level-4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



IoT Level-5 (Forest Fire Detection)

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

Detection)


IoT Level-5
Local
Cloud
Observer Node
App
Observer Node
REST/WebSocket Communication
Controller Service
Controller Service
Controller Service
REST/WebSocket Services
Analytics Component (IoT Intelligence)
Resource
Resource
Resource
Database
Endpoint Device
Endpoint Device
Coordinator Device
Routers/End Points
Coordinator
Cloud Storage & Analysis

g)

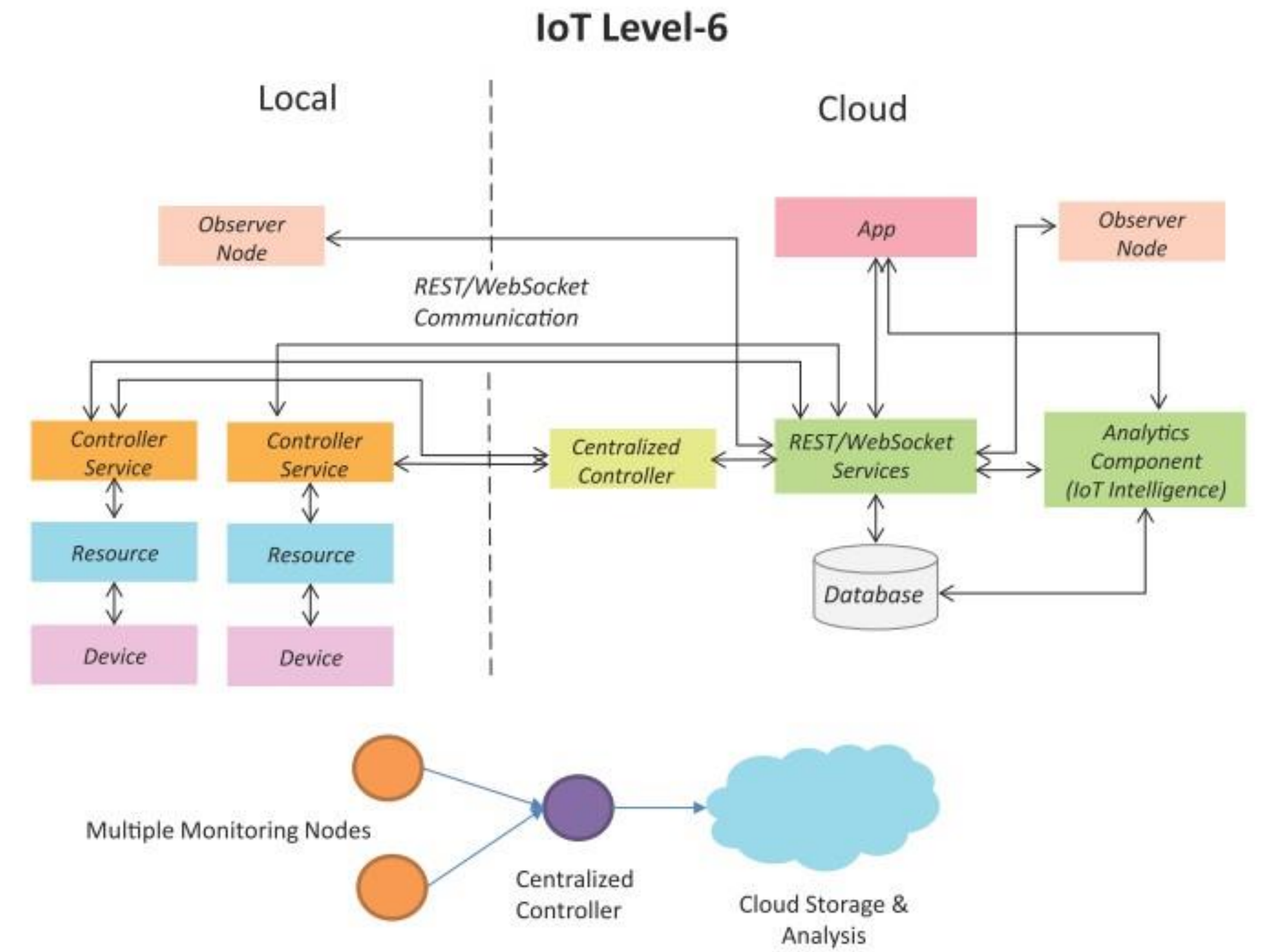
- A level-6 IoT system has **multiple independent end nodes** that perform

  sensing and/or actuation and send data to the cloud.
- Data is stored in the cloud and the application is **cloud-based.**
- The analytics component analyzes the data and stores the results in the

  cloud database.
- The results are visualized with the cloud-based application.
- The centralized controller is aware of the status of all the end nodes

  and sends control commands to the nodes.

g)


IoT Level-6
Local
Cloud
Observer Node
App
Observer Node
REST/WebSocket Communication
Controller Service
Controller Service
Centralized Controller
REST/WebSocket Services
Analytics Component (IoT Intelligence)
Resource
Resource
Database
Device
Device
Multiple Monitoring Nodes
Centralized Controller
Cloud Storage & Analysis

# of IOT

Level	Node	Analysis	Storage	Example
1	Single	Local	Local	Home Automation
2	Single	Local	Cloud	Smart Irrigation
3	Single	Cloud	Cloud	Vibration Monitoring
4	Multiple	Local	Cloud	Noise Monitoring
5	Multiple + Coordinator	Cloud	Cloud	Forest Fire detection
6	Multiple + Centralized Controller	Cloud	Cloud	Weather Monitoring

Challenges

Security

- Cyber Attacks, Data Theft

Privacy

- Controlling access and ownership of data.

InterOperability

- Integration Inflexibility

Legality and Rights

•Data Protection laws be followed, Data Retention an destruction policies

Economy and Development

- Investment Incentives, Technical Skill are Equirement

Course Outcomes: On completion of the course, learners should be able to -

CO1: Understand the fundamentals and need of Embedded Systems for the Internet of Things.

CO2: Apply IoT enabling technologies for developing IoT systems.