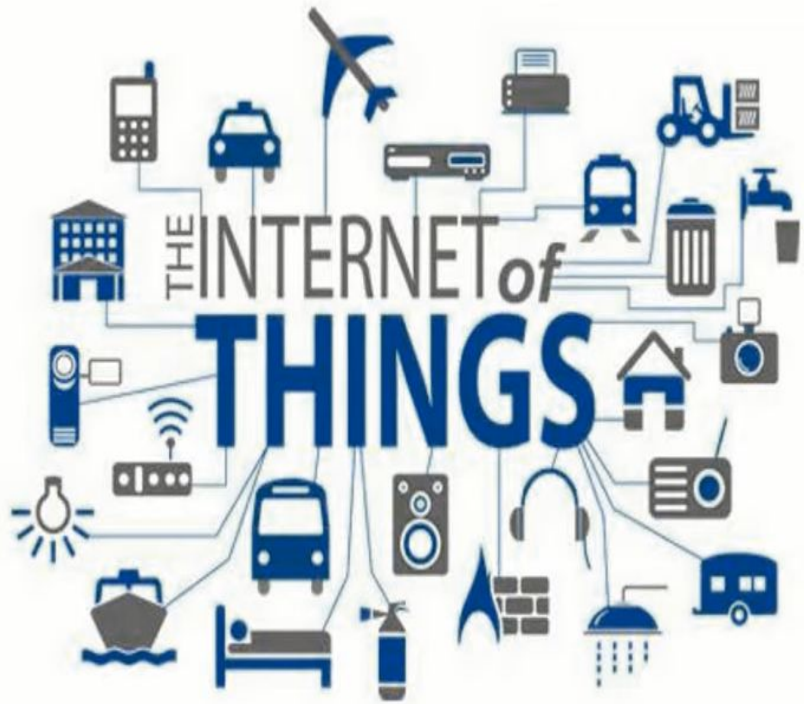
INTERNET OF THINGS

IoT

Internet of Things

The things which are working based on the internet is called as the IoT.



"IoT refers to the interconnection via the Internet of computing devices embedded in everyday objects, enabling them to send and receive data"

See the term, data!! Data is everything!!!

Definitions of IOT

- IoT is a ***world of interconnected things***.
- These things are capable of ***sensing***, ***actuating*** and ***communicating*** among themselves and with the environment.
- They have the ability to ***share information*** and ***acting autonomously to real / physical world events*** and ***activating various processes*** and ***enabling services with or without direct human intervention***.

Concept

- The Internet is a vast global network of connected servers, computers, tablets and mobiles that is governed by standard protocols for connected systems.
- It enables sending, receiving, or communication of information, connectivity with remote servers, cloud and analytics platforms.

Other Definitions

Adrian McEwen and Hakim Cassimally

Physical Object + Controller, Sensor and Actuators + Internet = IoT

General Framework for IoT

Gather + Enrich + Stream + Manage + Acquire + Organise and Analyse = IoT

Orcale

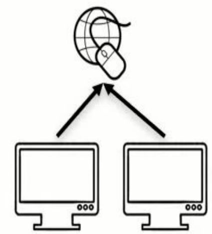
Gather + Consolidate + Connect + Collect + Assemble + Manage and Analyse = IoT

CONTD.,

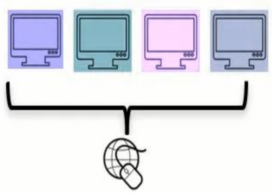
- SCOPE IS NOT JUST LIMITED TO GETTING THE DEVICES CONNECTED / NETWORKED! IT IS MORE THAN THAT!
- IT IS MORE ABOUT THE EXCHANGE OF MEANINGFUL INFORMATION FROM ONE DEVICE TO ANOTHER TO GET ANOTHER MEANINGFUL ACCOMPLISHMENT. (DATA, IS THE KING / QUEEN)
- BUT, THIS DATA HAS TO BE INTERPRETED CORRECT AND IT NEEDS MANY OTHER **THINGS** COME HAND IN HAND. MANY TECHNOLOGIES TO WORK SIDE BY SIDE.
  - IF YOU JUST GET THE DATA FROM THE SENSORS WITHOUT UNDERSTANDING HOW TO USE OR WHERE TO USE, IT IS USELESS! HENCE, INTERPRETATION MATTERS A LOT!!
- **IOT IS NOT ANY ONE TECHNOLOGY! IT IS COLLECTION OF TECHNOLOGIES AND DOMAIN KNOWLEDGE!**
- **A MECHANICAL ENGINEER HAS TO WORK WITH ELECTRICAL ENGINEER AND CS ENGINEER FOR A COMPLETE IOT PRODUCT!**

IT IS TIME TO UNDERSTAND HOW INTERNET HAS OCCUPIED OUR LIFE …

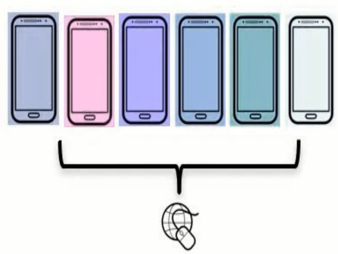
- REPORTS SUGGEST IT ALL STARTED IN 1990
  - 300,000 DESKTOP COMPUTERS WERE CONNECTED TO THE INTERNET.

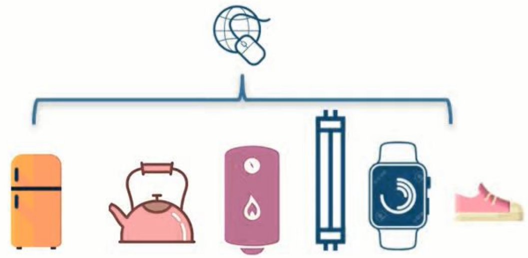


- IN 2000, IT GREW TO 300 + MILLION DESKTOP COMPUTERS (LAPTOP TOO) CONNECTED TO INTERNET.



- IN 2016 (YOU GUYS CAN TELL THIS!!) – ABOUT 2.5 BILLION MOBILE PHONES ARE CONNECTED TO INTERNET!





CONTD.,

- IN 2020, REFRIGERATORS, KETTLES, WATER HEATERS, CAMERAS, LIGHTS, HOME EQUIPMENT ETC. ALL WILL GET INTO INTERNET AND, TRUST ME, IT WOULD BE 15 BILLION DEVICES CONNECTED TO THE INTERNET.
- ABOUT 4 BILLION NAVIGATION SYSTEMS WOULD BE CONNECTED TO INTERNET. (CONNECTED CARS WOULD BECOME A REALITY)
- ABOUT 500 MILLION WEARABLES, WATCHES, SHOES ETC. WOULD BE CONNECTED TO INTERNET!
- 100S OF MILLIONS OF IMPLANTS IN HUMAN/ANIMALS WOULD TALK TO INTERNET.
- **HENCE… WHAT CAN WE EXPECT??**

CONTD.,
YES, IMPOSSIBLE IS NOTHING WITH IOT!

- WHEN YOU WAKE UP, THE WATER HEATER WOULD HAVE THE HOT WATER READY!
- DURING BATH, THE KETTLE WOULD HAVE COOKED YOUR FOOD, CONSUME AFTER BATH!
- WHEN YOU LOCK THE DOOR, THE CAR DOORS SHALL BE OPEN FOR YOU!
- A/C SHALL ADJUSTED BASED ON YOUR BODY TEMPERATURE.
- PARKING SLOTS SHALL BE INFORMED WHILE NEARING THE OFFICE.
- PILL BOX SHALL ALERT YOU THE TIME OF YOUR TABLET INTAKE!
- WHILE NEARING HOME, THE LIGHTS SHALL BE ON!
- YOUR WORKOUT ROOM LIGHTS/AMBIENCE SHALL BE SET. HEART BEAT/KM ALERT SHALL BE UPDATED!
- REFRIGERATOR SHALL ORDER MILK/EGG ON NEED.
- BED ROOM AC SHALL BE AUTOMATICALLY SWITCHED ON FOR YOU TO SLEEP.

SO, WHAT IOT IS?

Any time – Any device

Anyone

Any service / Any business

Any Network (Non homogenous)

Anywhere / Any location

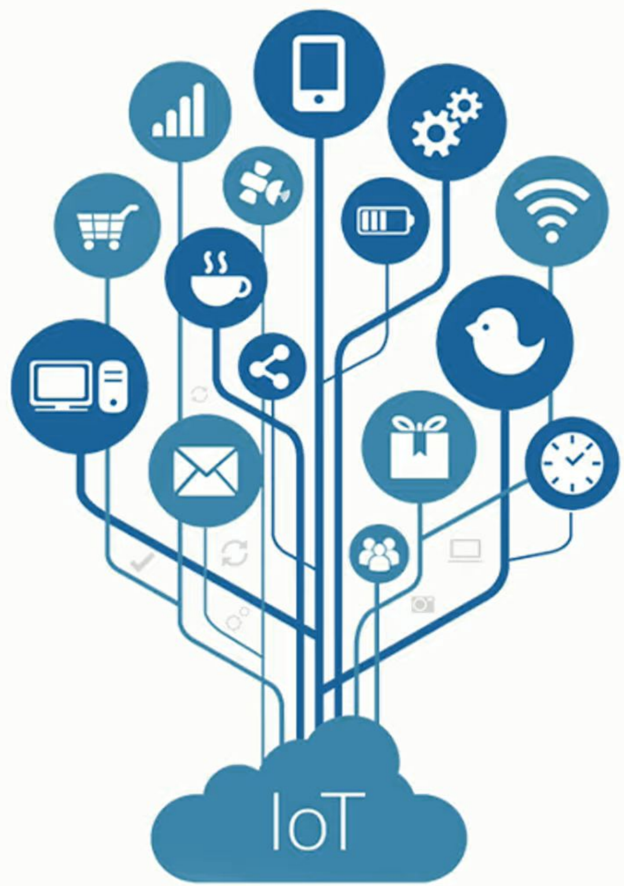
IoT enables the objects (which eventually are the things) to be sensed, while also controlling it remotely, which enables better interaction of physical world to the computers. This would improve the efficiency, accuracy with limited human intervention.

Objectives of IoT

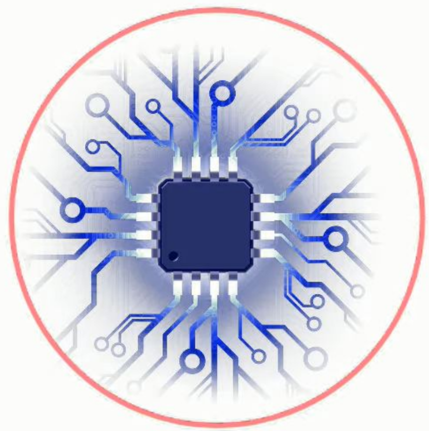
- *Connecting things* (devices, appliances, machines etc.)
- *Exchange of data and information*
- *Sensing, Processing, Control, Actuation and Monitoring*
- *Providing services*

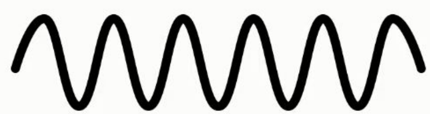
Example


HELLO !!


IoT

Example





Analog Signal

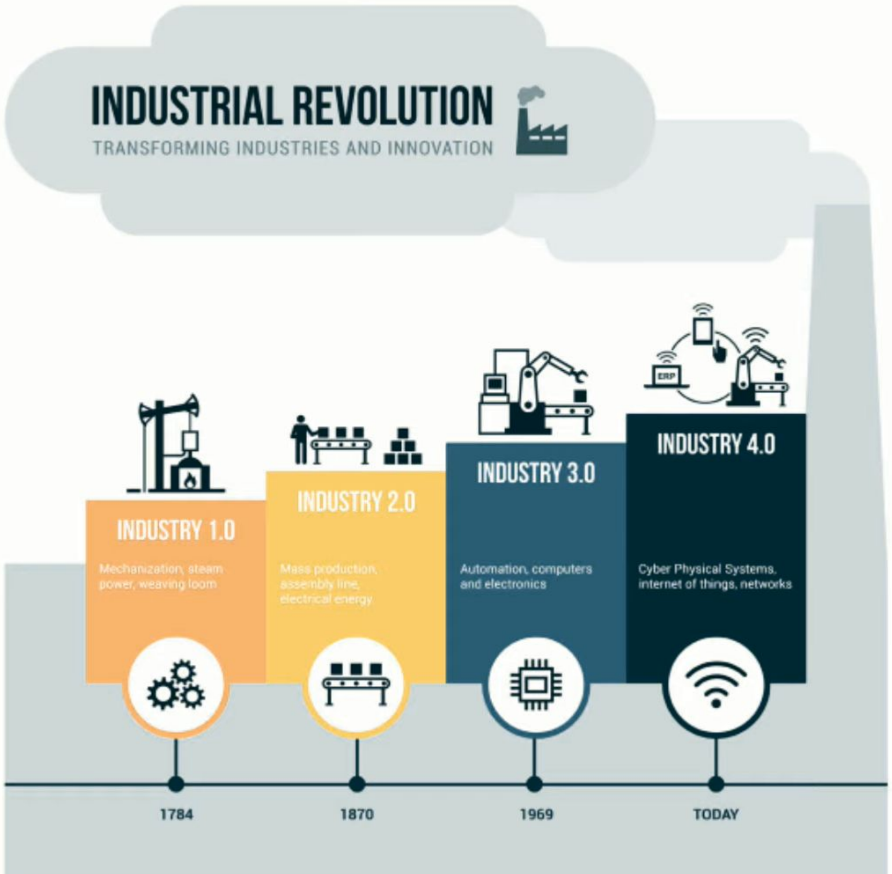
Digital Signal

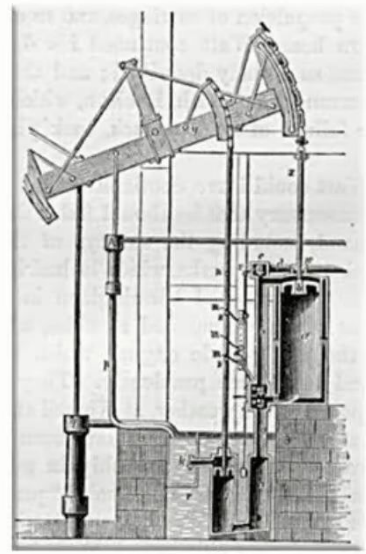
Who invented IoT?

History of IoT

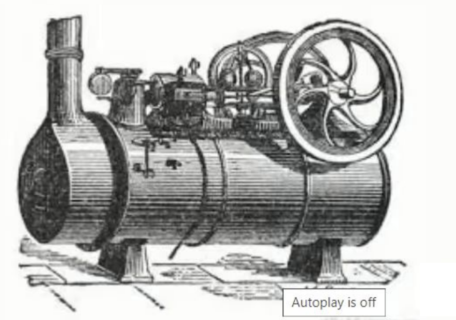
Why we need IoT?

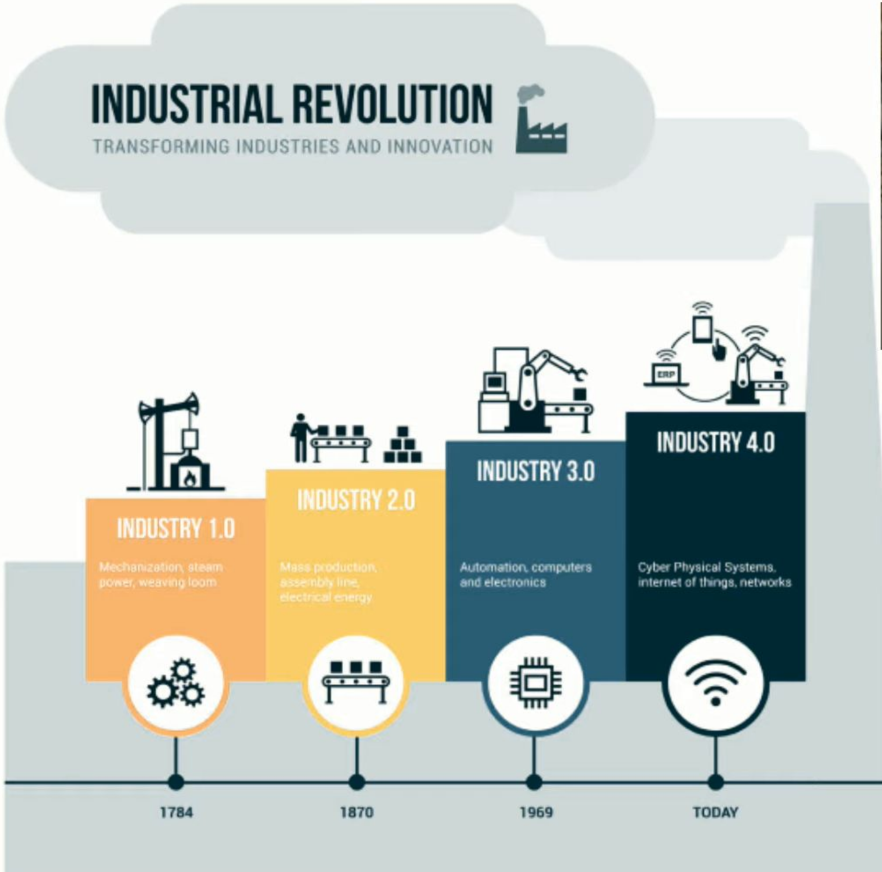
Industrial Revolution

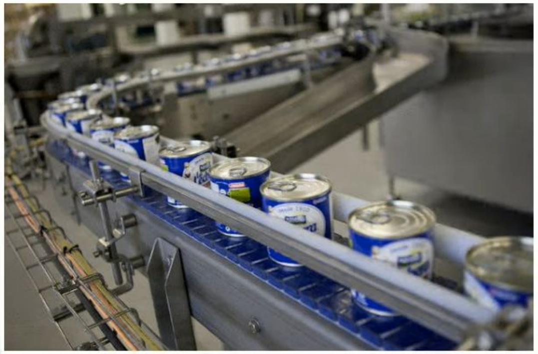


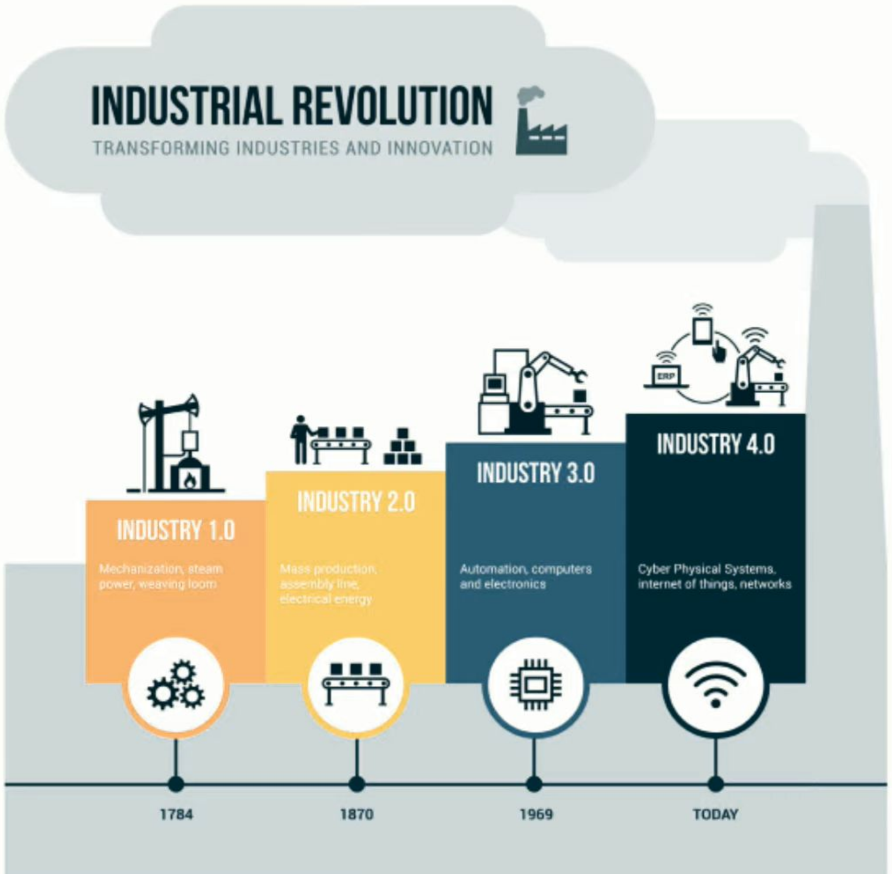


Watt's Engine, 1774

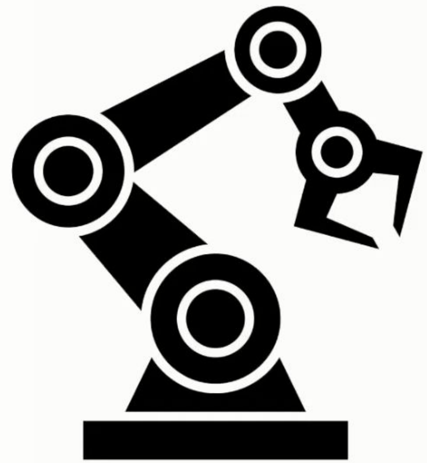


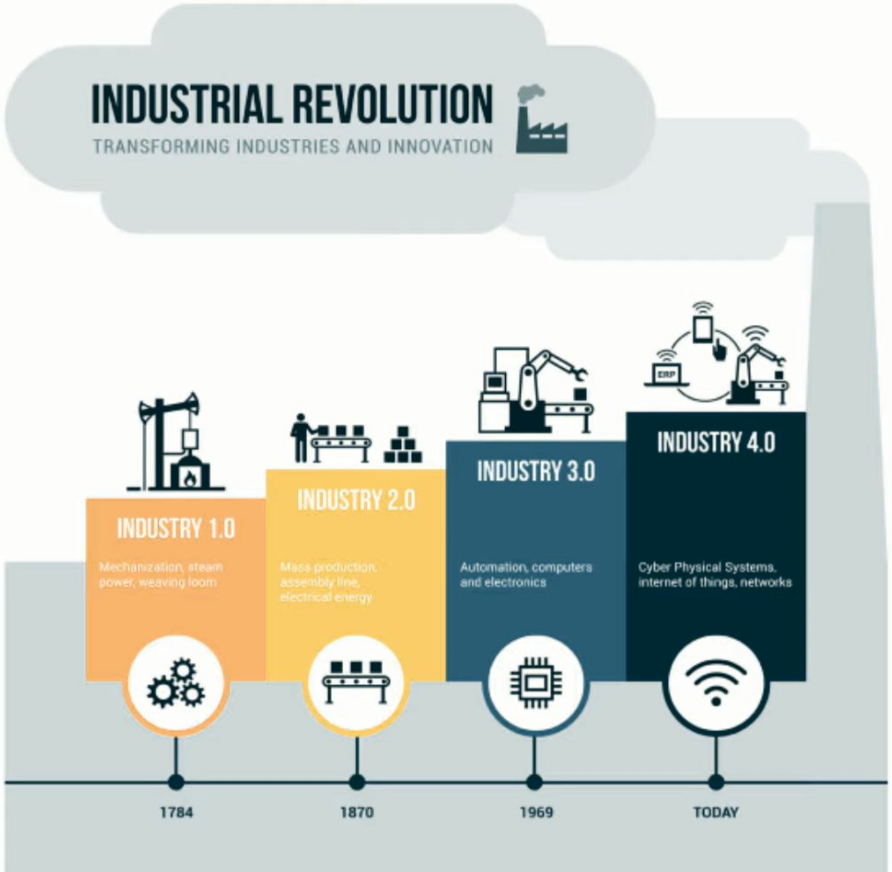

INDUSTRIAL REVOLUTION
TRANSFORMING INDUSTRIES AND INNOVATION
INDUSTRY 1.0
Mechanization, steam power, weaving loom
INDUSTRY 2.0
Mass production, assembly line, electrical energy
INDUSTRY 3.0
Automation, computers and electronics
INDUSTRY 4.0
Cyber Physical Systems, internet of things, networks
ERP
1784
1870
1969
TODAY





INDUSTRIAL REVOLUTION
TRANSFORMING INDUSTRIES AND INNOVATION
INDUSTRY 1.0
Mechanization, steam power, weaving loom
INDUSTRY 2.0
Mass production, assembly line, electrical energy
INDUSTRY 3.0
Automation, computers and electronics
INDUSTRY 4.0
ERP
Cyber Physical Systems, internet of things, networks
1784
1870
1969
TODAY



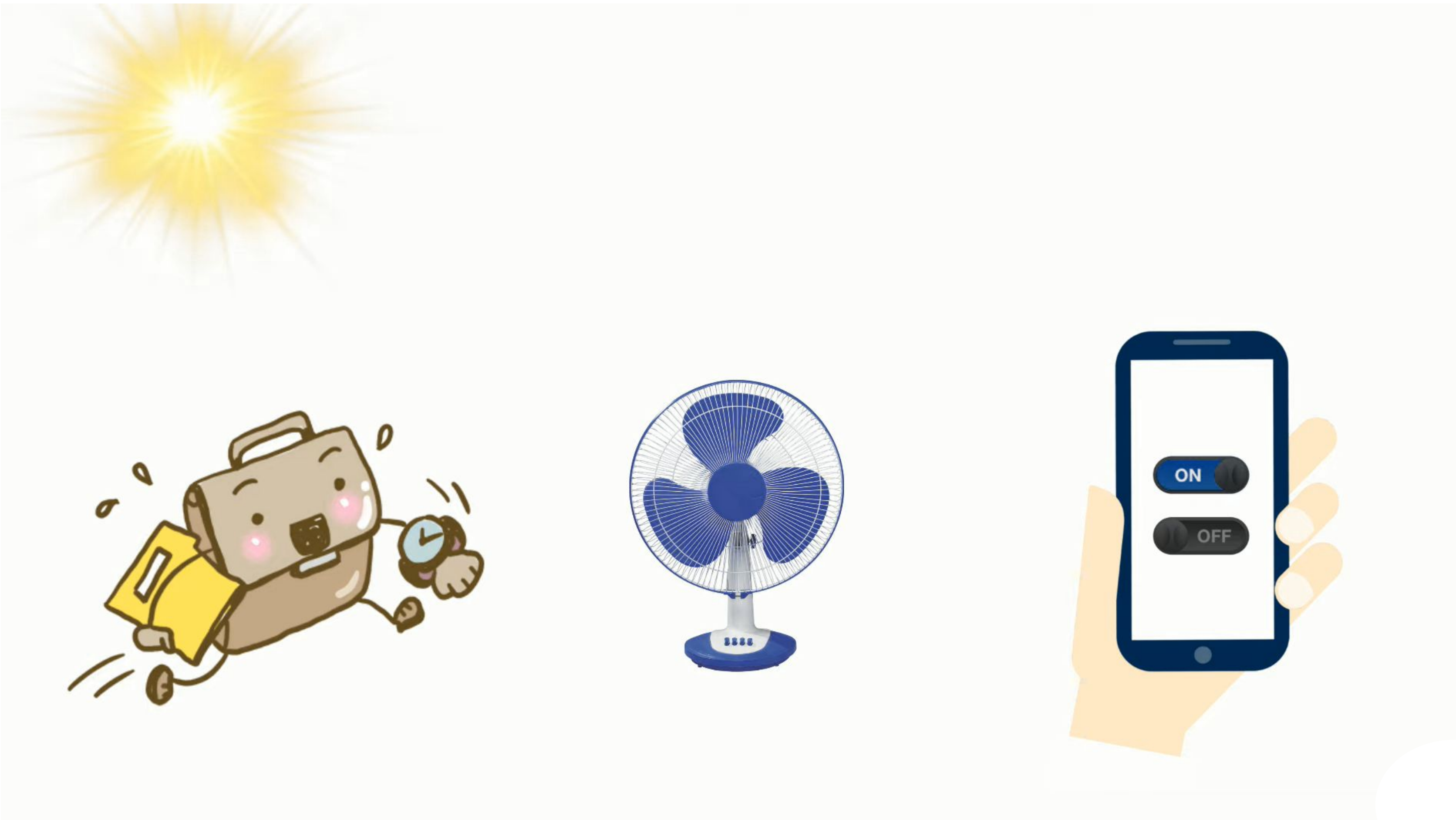


IoT

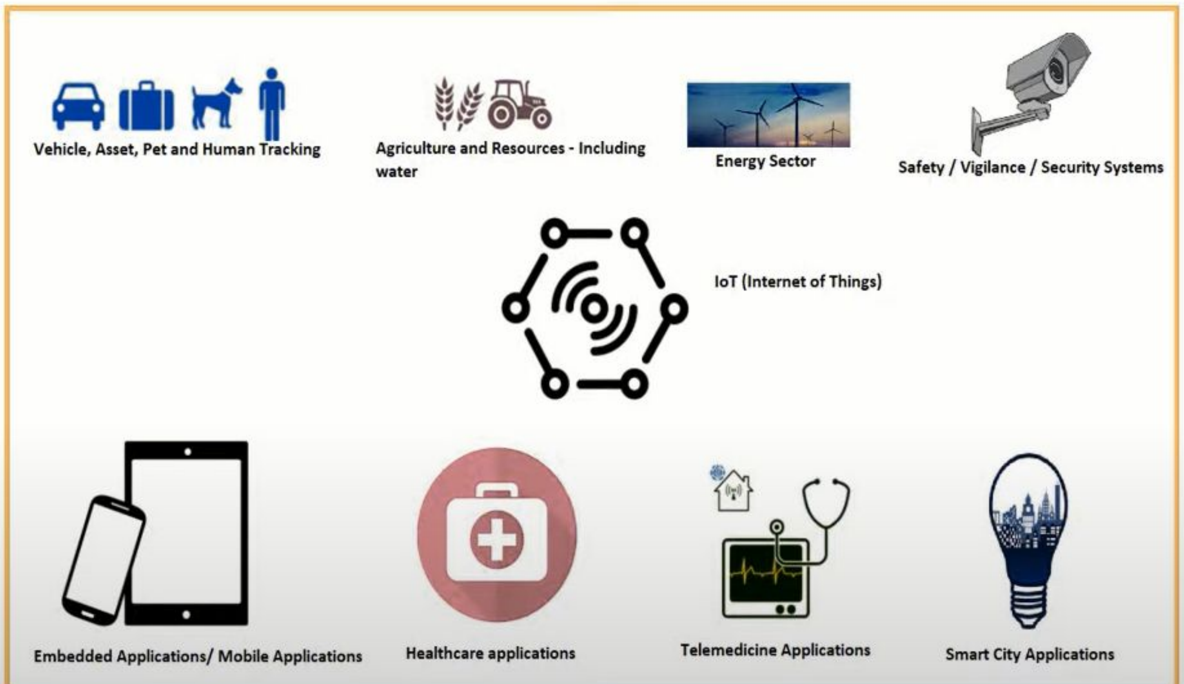
AI

Automation

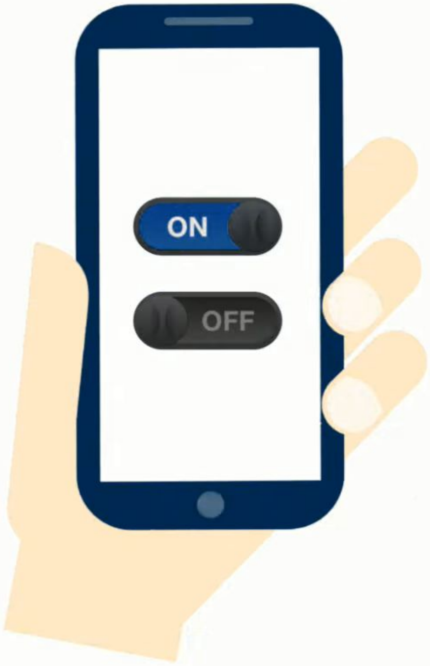
How it works?


ON
OFF

GROWTH AND APPLICATION SECTORS



Applications




Smart City
Smart Economy
Smart People


234
of 600 Cals

Smart Farm

Disadvantages of IoT

Characteristics of IOT

1. **CONNECTIVITY** -- THINGS IN IOT SHOULD BE CONNECTED TO THE INFRASTRUCTURE AND CONNECTIVITY IS AN IMPORTANT CHARACTER/REQUIREMENT FOR AN IOT INFRA. ANYONE, ANYWHERE, ANYTIME – CONNECTIVITY SHOULD BE GUARANTEED IN THE IOT INFRA. WITHOUT CONNECTION, NOTHING MAKES SENSE! (I SAY THAT, THINGS ARE CONNECTED AND THEY NEED CONNECTIVITY)

2. **INTELLIGENCE AND IDENTITY --** THE EXTRACTION OF KNOWLEDGE (I.E. WHAT IS TO BE INFERENCED) FROM THE GENERATED DATA IS VERY IMPORTANT. SENSORS GENERATE DATA, THE DATA IS TO BE INTERPRETED PROPERLY! EACH IOT DEVICE HAS AN UNIQUE IDENTITY (REMEMBER IP ADDRESS). THIS IDENTITY IS HELPFUL IN TRACKING THE EQUIPMENT AND AT TIMES TO QUERY THE STATUS.

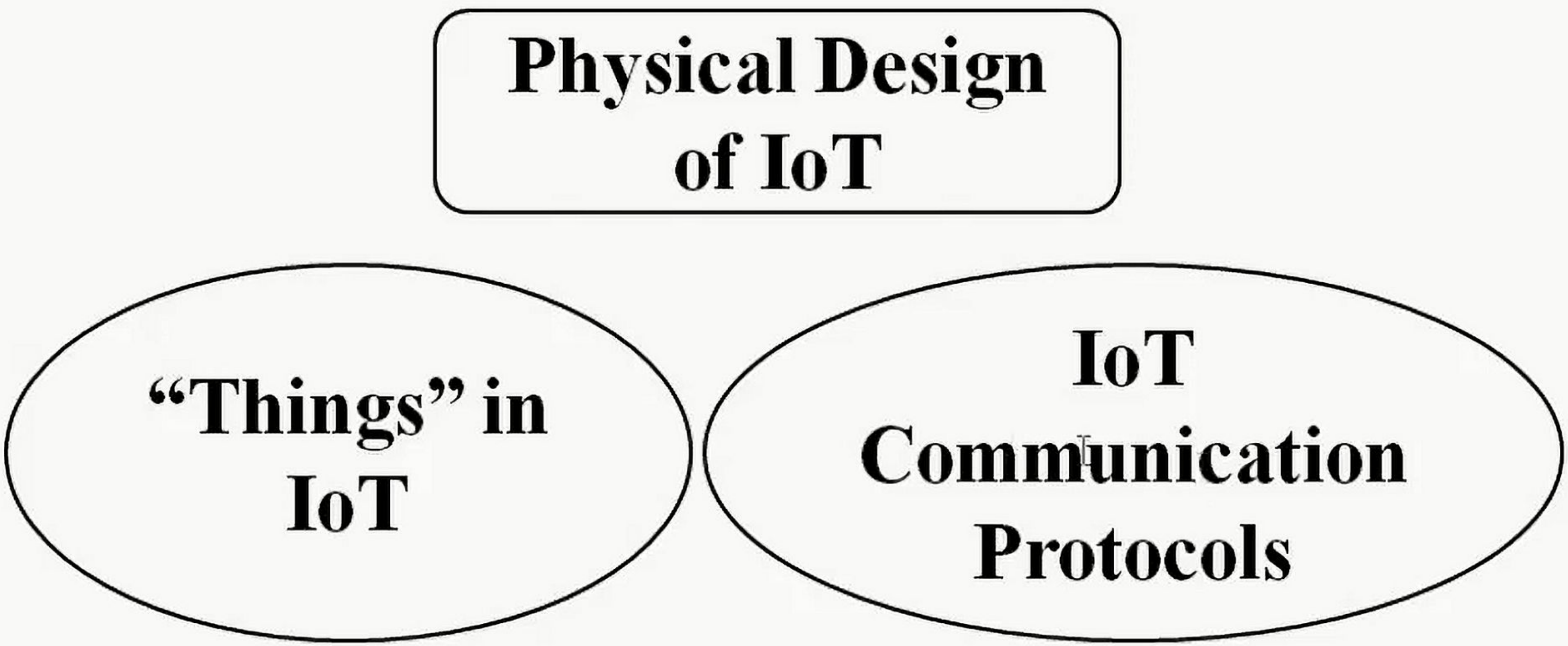
3. **SCALABILITY –** THE NUMBER OF THINGS (DEVICES) GETTING CONNECTED TO IOT INFRA IS GETTING INCREASED DAY BY DAY. HENCE, AN IOT SETUP SHALL BE CAPABLE TO HANDLE THE MASSIVE EXPANSION. ALSO, THE DATA GENERATED SHALL BE MASSIVE AND IT SHOULD BE HANDLED APPROPRIATELY.

4. **DYNAMIC AND SELF ADAPTING (COMPLEXITY) –** THE IOT DEVICES SHOULD DYNAMICALLY ADAPT ITSELF TO THE CHANGING CONTEXTS. ASSUME A CAMERA MEANT FOR SURVEILLANCE. IT MAY HAVE TO WORK IN DIFFERENT CONDITIONS AS DIFFERENT LIGHT SITUATIONS (MORNING, AFTERNOON, NIGHT)

5. **ARCHITECTURE –** ARCHITECTURE CANNOT BE HOMOGENEOUS IN NATURE. IT SHOULD BE HYBRID, SUPPORTING DIFFERENT MANUFACTURER'S PRODUCT TO BE IN THE IOT NETWORK.

6. **SAFETY** - HAVING GOT ALL THE THINGS CONNECTED TO INTERNET, THE PERSONAL DATA (IF SENSITIVE) IS UNDER THREAT. HENCE, SECURING THE DATA IS A MAJOR CHALLENGE . NOT ONLY DATA SECURITY, THE EQUIPMENT GETTING INVOLVED IN IOT NETWORK IS HUGE. HENCE, PERSONA SAFETY IS ALSO TO BE CONSIDERED. PRIVACY WITH PROTECTION! ☺

Physical Design of IoT

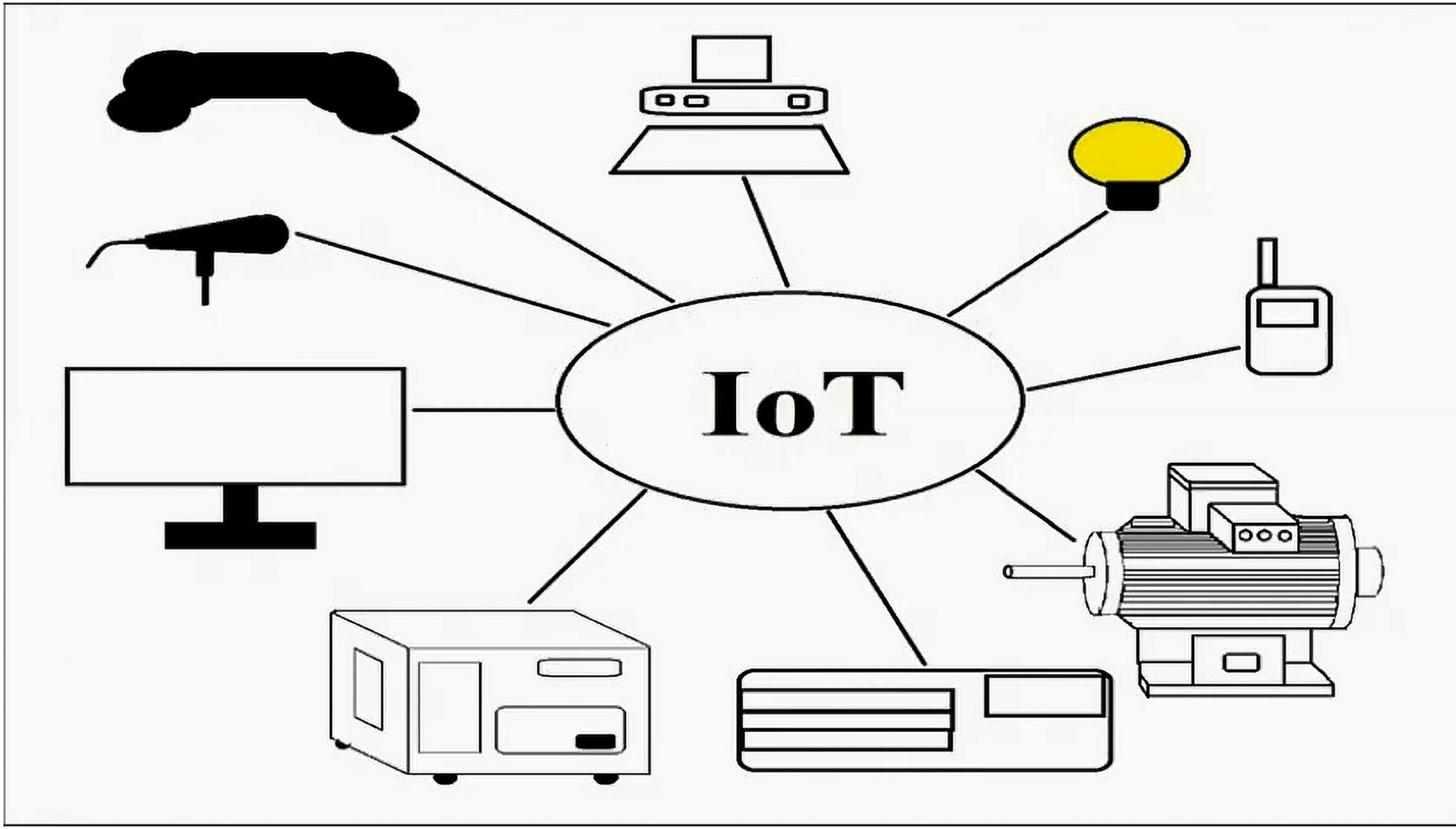


What are 'Things' in IoT ?

- *'Things' in IoT* can be *anything that operates or functions by electrical or electronic means.*

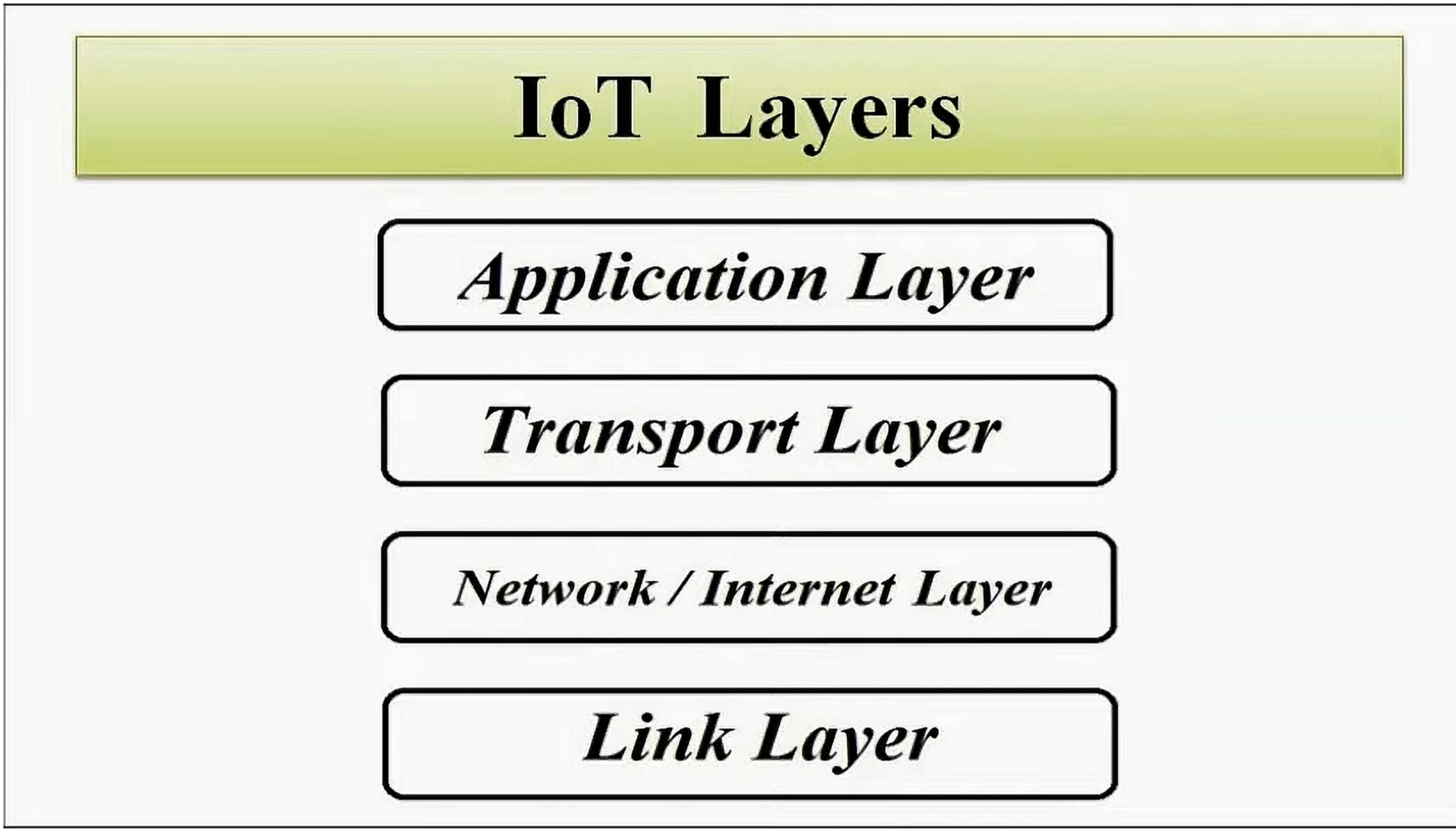
- It includes *electrical machinery, electronic gadgets, sensors and transducers, actuators and controllers, processors* etc.

- These devices have unique identities.
- They should be capable of remote sensing, processing controlling, actuation and monitoring.
- IoT devices exchange data with each other.


IoT

IoT Communication Protocols

- The communication protocols handle the ***exchange or transfer of data between devices in the IoT network***.
- It operates at several levels or layers such as ***link layer***, ***network layer***, ***transport layer*** and ***application layer***.


IoT Layers
Application Layer
Transport Layer
Network / Internet Layer
Link Layer

Functions of IoT Layers

- **Link Layer:** It determines how the data is sent physically over the connecting medium i.e., twisted pair cable, coaxial cable, optical fibre or wireless mode.

- **Ex- 802.3 – Ethernet, 802.11 – WiFi,**
  802.16 – WiMax, 2G/3G/4G communications etc.

- **Network / Internet Layer :** This layer is responsible for sending IP (internet protocol) datagrams from source network to destination network. The datagrams contain source and destination identitiy or address.

- It performs ***addressing of the communicating devices*** and ***routing of data packets***.

- Ex- **IPv4**, **IPv6**, **6LoWPAN** etc.

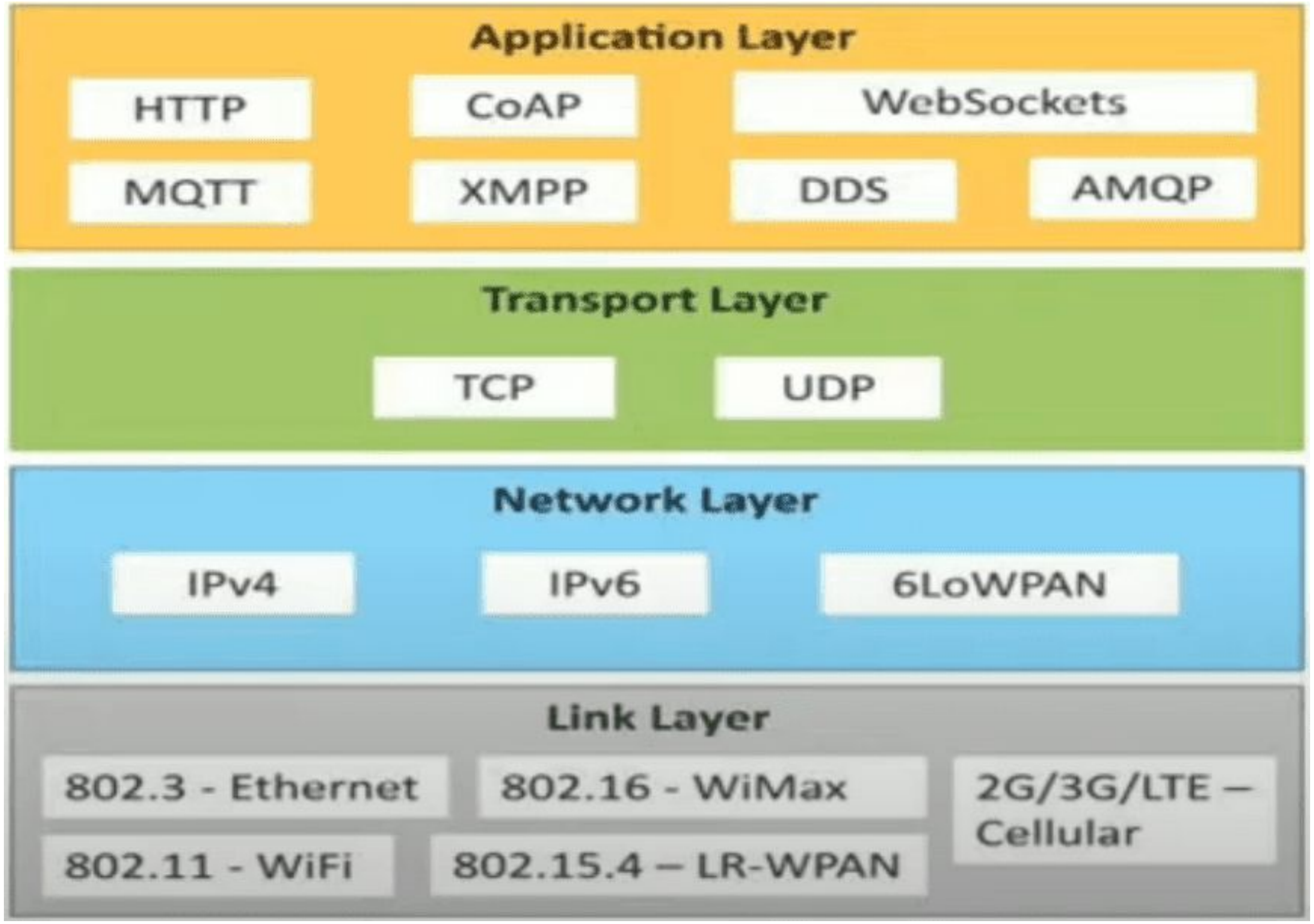
- **Transport Layer:** This layer provides end – to – end message transfer capability.

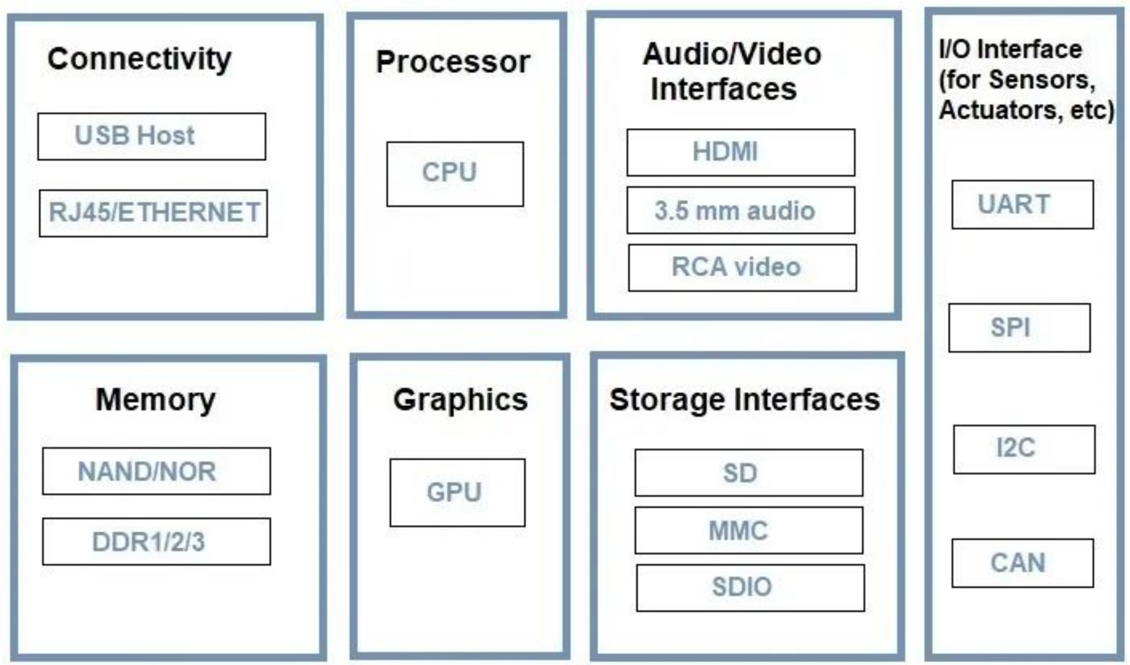
- It also provides various services such as ***error control***, ***segmentation***, ***flow*** and ***congestion control***.

- Ex- **Transmission Control Protocol (TCP)** ,

  USER **Datagram Protocol (UDP)** etc.

- **Application Layer :** This layer defines protocols about how a particular layer interfaces with the layers below it to send data over the network.
- It performs ***data encoding*** and ***process-to-process communication***, while the transport layer performs ***data encapsulation***.
- Ex- **Hypertext Transfer Protocol (HTTP), Websocket, Data Distribution Service (DDS)** etc.


Application Layer
HTTP
CoAP
WebSockets
MQTT
XMPP
DDS
AMQP
Transport Layer
TCP
UDP
Network Layer
IPv4
IPv6
6LoWPAN
Link Layer
802.3 - Ethernet
802.16 - WiMax
2G/3G/LTE – Cellular
802.11 - WiFi
802.15.4 – LR-WPAN



Generic Block Diagram of IoT Devices

UART – Universal Asynchronous Receiver / Transmitter

SPI : Serial Peripheral Interface

I2C : Inter-Integrated Circuit

CAN : Control Area Network Interface

HDMI : **High-Definition Multimedia Interface**

RCA : Radio Corporation of India

SD : Secure Digital

MMC : Multimedia Card

SDIO :

DDR : Double Data Rate

Internet of Things (IoT)

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

Dynamic & Self - Adapting

Capable to adapt dynamically with the changing contexts and take actions based on their operating conditions, user's context or sensed environment

Eg: Surveillance Cameras

Self- Configuring

- ✔ Allows large number of devices to work together to provide certain functionality
- ✔ Ability to configure themselves with IoT infrastructure
- ✔ Setup the networking, and fetch latest software upgrades with minimal manual or user intervention

Interoperable Communication Protocols

- ✔ Support a number of interoperable communication protocol
- ✔ Communicate with other devices and other IoT infrastructure

Unique Identity

- ✔ IoT devices – unique identity – IP address / URI
- ✔ IoT systems – intelligent interfaces – adapt to context , user and environment
- ✔ IoT device interface – user query the device, monitor the status, control remotely

Integrated into Information Network

IoT devices dynamically discovered in the network

Eg: Weather Monitoring

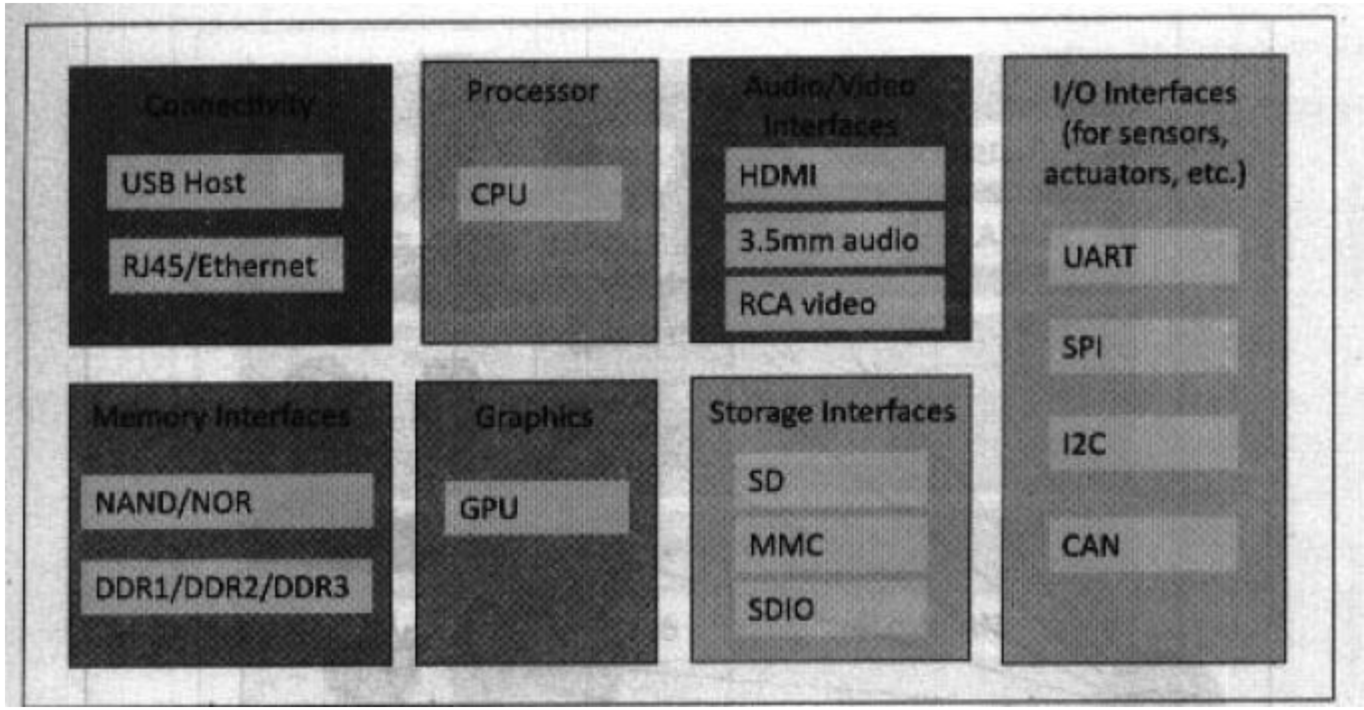
Physical Design of IoT

Things in IoT

- Things in IoT usually refers to IoT devices which have unique identities and can perform
  - ✔ Remote Sensing
  - ✔ Actuating
  - ✔ Monitoring capabilities

- Devices can exchange data with other connected devices and applications or collect data from other devices and process the data either locally or send the data to centralised servers or cloud based application for performing the data or perform local task within the IoT infrastructure

Block Diagram of an IoT Device



- Several interfaces for connections to other devices

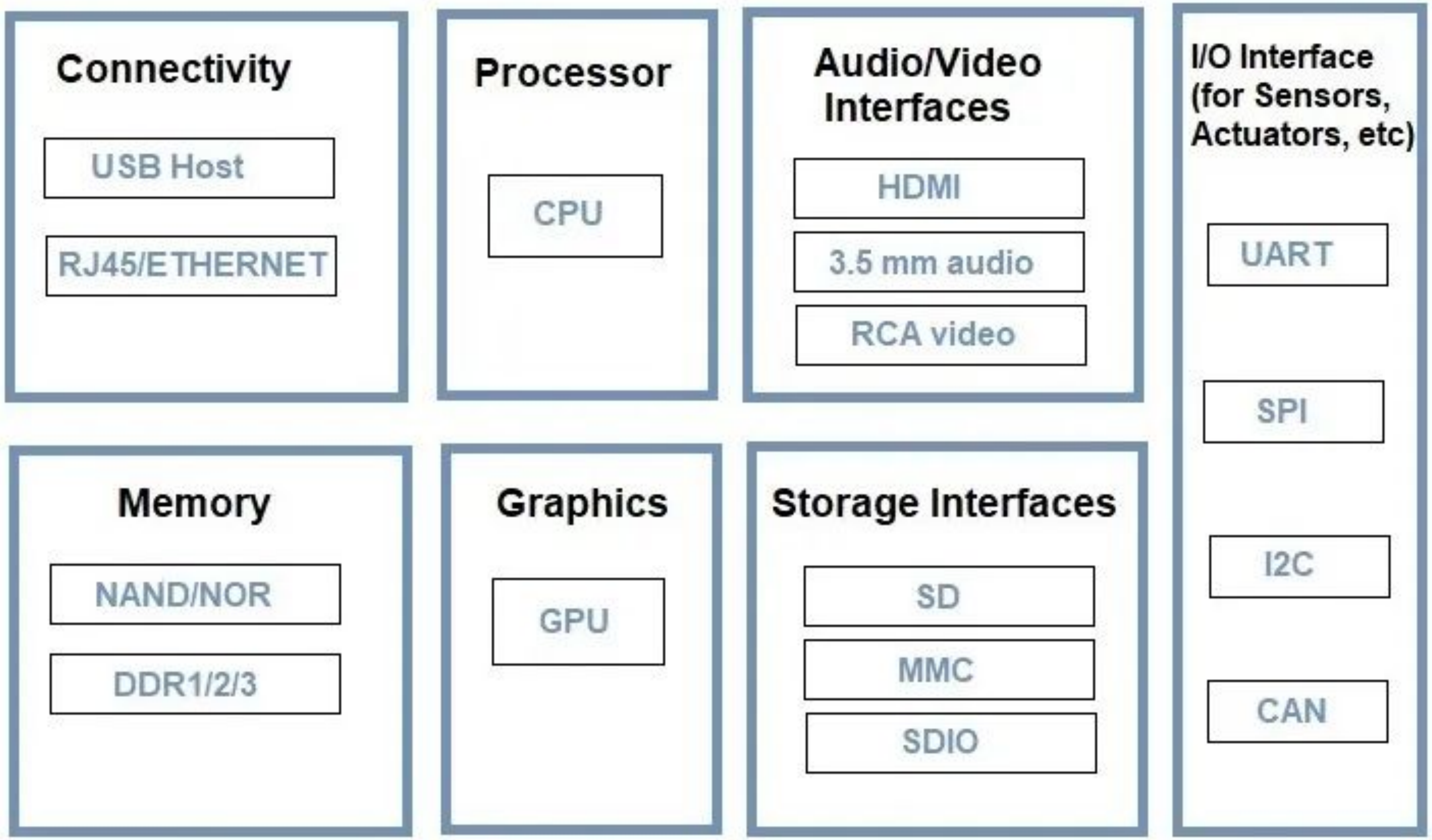
  I/O interface –sensors

  Internet connectivity interface

  Memory and storage interface

  Audio/Video interface
- IoT devices can be connected to actuators that allow them to interact with other physical entities

Block Diagram of an IoT Device

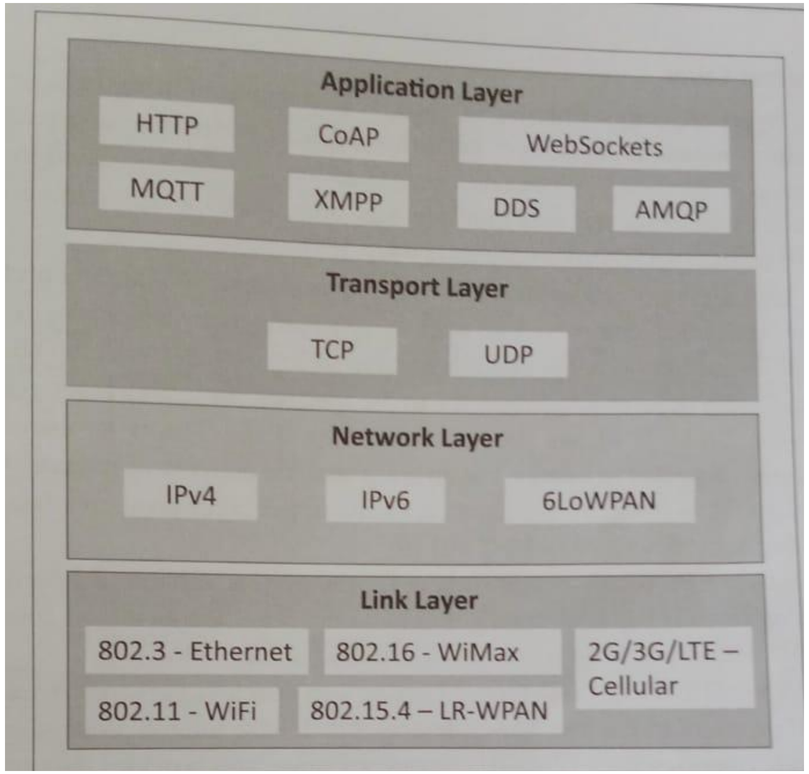


Generic Block Diagram of IoT Devices

List some of IoT devices

IoT Protocols

IoT Protocols Stack



IoT Protocols

Link Layer

- Determines how the data is physically sent over the network's physical layer or medium (Copper wire, coaxial cable, radio wave)
- Scope – Limited to the local network connection to which host is attached.
- Hosts on the same link exchange data packets over the link layer protocol
- Determines how the packets are coded and signalled by the hardware device over the medium to which the host is attached.

✔ 802.3 – Ethernet
✔ 802.11 – WiFi
✔ 802.16 – WiMax
✔ 802.15.4 – LR-WPAN
✔ 2G/3G/4G – Mobile Communication

Link Layer Protocols

- **802.3 – Ethernet** - Collection of wired Ethernet standards for the link layer
- Example: 10BASE5 Ethernet – Coaxial cable as a shared medium
- 802.3i is a standard for 10BASE-T Ethernet over copper twisted-pair connections
- 802.3j is a standard for 10BASE-F Ethernet over fiber optic connections
- 802.3ae is a standard for 10Gbit/s Ethernet over fiber
- The shared medium (broadcast) – carries communication for all the devices on the network
- Data sent by one device can be received by all devices subject to propagation conditions and transceiver capabilities

Link Layer Protocols (contd.,)

802.11 – WiFi - collection of wireless local area network (WLAN) communication standards

Operational Frequency

- ✔ 802.11a – 5GHz band
- ✔ 802.11b – 2.4 GHz
- ✔ 802.11g – 2.4 GHz
- ✔ 802.11n – 2.4/5 GHz
- ✔ 802.11ac – 5GHz
- ✔ 802.11ad – 60 GHz
- ✔ Data rate – 1 Mbps to 6.75 Gbps

Link Layer Protocols (contd.,)

802.16 – WiMax – collection of wireless broadband standards

- ✔ Data rate – 1.5 Mbps to 1 Gbps
- ✔ 802.16m – data rate of 100 Mbps for mobile stations and 1Gbps for fixed stations

802.15.4 – LR-WPAN – collection of standards for low-rate wireless personal area networks (LR-WPAs)

- ✔ Provides basis specification for high-level communication protocols such as ZigBee
- ✔ Data rate – 40 Kbps to 250 Kbps
- ✔ Provides low-cost and low-speed communication for power constrained devices

Link Layer Protocols (contd.,)

2G / 3G / 4G – Mobile Communication

- ✔ 2G – GSM & CDMA
- ✔ 3G – UMTS & CDMA200
- ✔ 4G – LTE
- ✔ IoT devices based on these standards can communicate over cellular networks
- ✔ Data rate – 9.6 Kbps (2G) to 100 Mbps (4G)

Network / Internet Layer

- Responsible for sending IP datagrams from the source network to the destination network
- Performs host addressing and routing
- Datagram – contain source and destination address
- Host identification – IP addressing scheme - IPv4 or IPv6

IPv4

✔ 32-bit address - address space 2^{32}

✔ More devices connected to Internet – addresses got exhausted – succeeded by IPv6

✔ IP – establish connections on packet network, but do not guarantee delivery of packets

✔ Guaranteed delivery and data integrity – handled by TCP

Network / Internet Layer (contd.,)

IPv6 – 128-bit address

6LoWPAN – IPv6 over Low Power Wireless Personal Area Networks)

- ✔ Brings IP protocol to low-power devices which have limited processing capability
- ✔ Operates in 2.4 GHz
- ✔ Data rate – 250 Kbps
- ✔ 6LoWPAN – works with 802.15.4 link layer protocol and defines compression mechanism for IPv6 datagrams over IEEE 802.15.4 based networks

Transport Layer

- End-to-end message transfer capability independent of underlying network
- Message transfer – set up connections – using handshakes (TCP) or without handshakes / acknowledgements (UDP)
- Provides error control, segmentation, flow control and congestion control

Transmission Control Protocol (TCP)

✔ Widely used by web browsers (HTTP, HTTPS) , e-mail (SMTP), file transfer(FTP)

✔ TCP is connection oriented and stateful protocol

✔ TCP ensures reliable transmission of packet in-order

✔ Provides – error detection capability – duplicate packets can be discarded and lost packets are retransmitted

✔ Flow control – rate at which the sender sends the data is not too high for the receiver to process

✔ Congestion control – avoiding network congestion and congestion collapse

Transport Layer (contd.,)

User Datagram Protocol (UDP)

- ✔ Connectionless protocol
- ✔ Useful for time-sensitive applications – very small data units to exchange – not overhead of connection set up
- ✔ UDP – transaction oriented and stateless protocol
- ✔ Does not provide guaranteed delivery, ordering of messages and duplicate elimination

Application Layer

- Application layer – how application interface with the lower layer protocols to send data over the network – process-to-process connections using ports
- Application (files) – encoded by application layer protocol – encapsulated in the transport layer protocol – can be connection or transaction oriented communication over the network
- Port numbers – application addressing (80 – HTTP, 22-SSH)

✔ HTTP
✔ CoAP
✔ WebSocket
✔ MQTT
✔ XMPP
✔ DDS
✔ AMQP

Application Layer (contd.,)

HTTP – Hyper Text Transfer Protocol – foundation for World Wide Web (WWW)

- ✔ Commands – GET, PUT, POST, DELETE, HEAD, TRACE, OPTIONS
- ✔ Follows – Request – Response model – client sends requests to a server using HTTP commands
- ✔ HTTP – stateless protocol – each HTTP request is independent of the other requests
- ✔ HTTP client can be a browser or an application running on the client
- ✔ HTTP protocol uses – Universal Resource Identifiers (URIs) to identify HTTP resources

Application Layer (contd.,)

CoAP – Constrained Application Protocol – application layer protocol for machine-to-machine application, constrained environments with constrained devices and constrained networks.

- ✔ CoAP – web transfer protocol and uses a request-response model – runs on top of UDP
- ✔ Uses client-server architecture – using connectionless datagrams
- ✔ Can easily interface with HTTP
- ✔ Supports methods – GET, PUT, POST and DELETE

Application Layer (contd.,)

WebSocket – allows full-duplex communication over single socket connection for sending message between client and server

- ✔ It is based on TCP and allows streams of message to be sent back and forth between the client and server while keeping TCP connection open
- ✔ Client – browser, mobile application, IoT devices

Application Layer (contd.,)

MQTT – Message Queue Telemetry Transport – light-weight messaging protocol based on publish- subscribe model

- ✔ Uses client-server architecture – client(IoT devices) connect to the server(MQTT broker) and publishes message to topics on the server
- ✔ The broker forwards the messages to the client subscribed to topics
- ✔ MQTT – well suited for constrained environment – devices have limited processing and memory resources – low network bandwidth

Application Layer (contd.,)

XMPP – Extensible Messaging and Presence Protocol – real time communication – streaming XML data between network entities

- ✔ Supports wide range of applications – messaging, presence, data syndication, gaming, multi-party chat and voice/video calls
- ✔ Allows sending small chunks of XML data from one network entity to another in near real-time
- ✔ It is a decentralized protocol and uses client-server architecture
- ✔ Supports both client-to-server and server-to-server communication paths
- ✔ In IoT context, XMPP allows real-time communication between IoT devices

Application Layer (contd.,)

DDS – Data Distribution Service – data centric middleware standard for device-to-device or M2M communication

- ✔ Uses publish-subscribe model – publishers(devices that generate data) create topics to which subscribers(devices that want to consume data) can subscribe
- ✔ Publisher – object responsible for data distribution
- ✔ Subscriber – responsible for receiving published data
- ✔ DDS provides quality of service (QoS) control and configurable reliability

Application Layer (contd.,)

AMQP – Advanced Message Queuing Protocol – open application layer protocol for business messaging

- ✔ Supports both point-to-point and publisher/subscriber models, routing and queuing
- ✔ AMQP brokers receive messages from publishers (devices or applications that generate data) and route them over connections to consumers (application that process data)
- ✔ Publishers publish the messages to exchanges which then distribute message copies to queues
- ✔ Messages are either delivered by the broker to the consumers which have subscribed to the queues or the consumers can pull the messages from the queues

Thank you

IoT Communication Model

- Request-Response communication model
- Publish-Subscribe communication model
- Push-Pull communication model
- Exclusive Pair communication model

IoT Communication APIs

- REST – Based Communication APIs
- WebSocket – Based Communication APIs

Difference between REST and WebSocket-based Communication APIs

Comparison Based on	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 requests
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

IoT Enabling Technologies

- Wireless Sensor Network
- Cloud Computing
- Big Data Analytics
- Communication Protocols
- Embedded Systems

Wireless Sensor Network- WSN

- Distributed Devices with sensors used to monitor the environmental and physical conditions
- Consists of several end-nodes acting as routers or coordinators too
- Coordinators collects data from all nodes / acts as gateway that connects WSN to internet
- Routers route the data packets from end nodes to coordinators

Example of WSNs in IoT & Protocols used

Example

- Weather monitoring system
- Indoor Air quality monitoring system
- Soil moisture monitoring system
- Surveillance systems
- Smart Grids
- Health monitoring systems

Protocols

- Zigbee

Zigbee

- Based on IEEE 802.15.4
- Operates at 2.4 GHz frequency
- Offers data rate from 10 to 100 meters depending on power o/p and environmental conditions

- Power of WSN – number of low-cost and low-power sensing nodes
- Self organizing networks – makes n/w robust-recovers (reconfigure itself) from failures and adding new nodes.
- Since large no. of nodes in WSN – Manual config. not possible

Cloud Computing

- Deliver applications and services over Internet
- Provides computing, networking and storage resources on demand
- Cloud computing performs services:
  - IaaS (Infrastructure as a Service)
  - PaaS (Platform as a Service)
  - SaaS (Software as a Service)

Cloud Services

IAAS : Rent Infrastructure

- Cloud-based services, pay-as-you-go for services such as storage, networking, and virtualization.

PAAS:

- Supply an on-demand environment for developing, testing, delivering and managing software applications
- Hardware and software tools available over the Internet

SAAS :

- Method for delivering software applications over the Internet, on demand and typically on a subscription basis.
- Software that’s available via a third-party over the Internet

SAAS

- BigCommerce
- Google Apps
- Salesforce
- Dropbox
- MailChimp
- ZenDesk
- DocuSign
- Slack
- Hubspot

IAAS

- DigitalOcean
- Linode
- Rackspace
- **Amazon** Web Services (AWS)
- Cisco Metapod
- Microsoft Azure
- Google Compute Engine (GCE)

PAAS

- AWS **Elastic** Beanstalk
- Windows Azure
- Heroku
- Force.com
- Google App Engine
- Apache Stratos
- OpenShift

Big Data Analytics

- Collection of data whose volume, velocity or variety is too large and difficult to store, manage, process and analyze the data using traditional databases
- It involves data cleansing, processing and visualization
- Lots of data is being collected and warehoused
  - Web data, e-commerce
  - purchases at department/ grocery stores
  - Bank/Credit Card transactions
  - Social Network

Big Data Analytics

- **Variety includes different types of data**
  - Structured -Relational data
  - Unstructured – Word,PDF,Media logs
  - Semi-Structured - XML data
  - All of above (text data, image,audio,video,sensor data)

Big Data Analytics

-Velocity refers to speed at which data is processed

- Batch
- Real-time
- Streams

Big Data Analytics

- Volume refers to the amount of data
- Terabyte
- Records
- Transactions
- Files
- Tables

Communication Protocols

- Form the backbone of IoT systems and enable network connectivity and coupling of applications
- Protocols define
  - Data exchange formats, data encoding schemes, addressing schemes routing of packets from source to destination
  - Sequence control, flow control retransmission of packets

Embedded Systems

- Computer system that has computer hardware and software embedded to perform specific tasks
- Designed to perform specific tasks
- Components of embedded system
  - Microprocessor/Microcontroller
  - Memory(RAM, ROM,cache..)
  - Networking Units(Ethernet, Wi-Fi adapter)
  - I/O Units(display, Keyboard..)
  - Storage(Flash memory)

Embedded systems

- Digital watches
- Digital cameras
- Vending Machines,
- Appliances(Washing Machines, Microwave oven…)

IoT Levels and Deployment Templates

An IoT system comprises the following components:

1. Device – Identification, Remote Sensing, Actuating, Remote monitoring capabilities
2. Resource – Software Components on IoT devices for accessing, processing and storing sensor information or controlling actuators connected to the devices. Also include software component that enables network access for the devices
3. Controller Service – Sends data from the devices to the web services and receive commands from the application for controlling the devices
4. Database – Stores data generated by the devices
5. Web Service – Serves as a link between the IoT device, application, database and analysis component
6. Analysis Component – Analyse the data and generate the result
7. Application – Provides the interface that the users can use to control and monitor various aspects of IoT systems

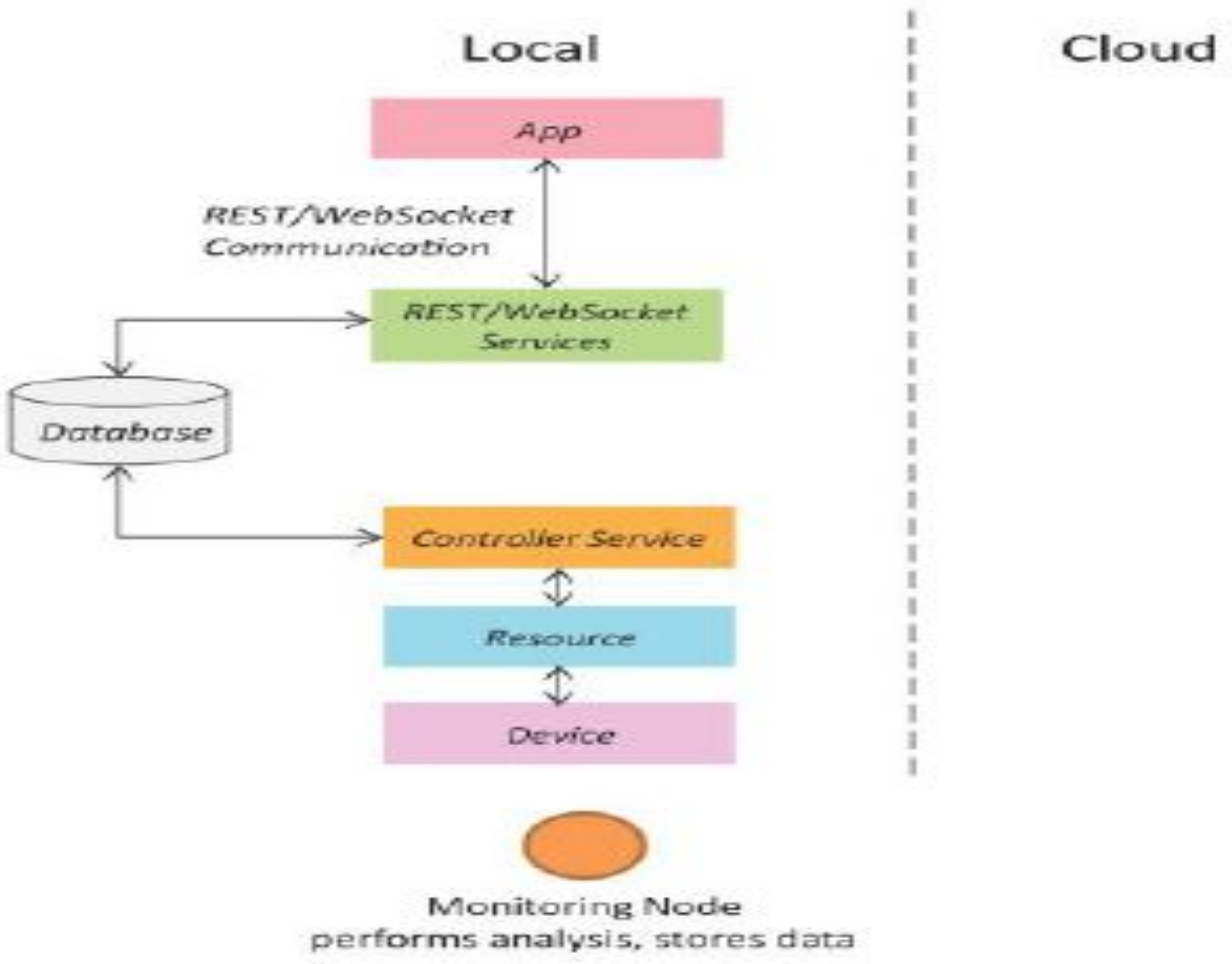
IoT Levels & Deployment Templates

- IoT Level 1
- IoT Level 2
- IoT Level 3
- IoT Level 4
- IoT Level 5
- IoT Level 6

IoT Level-1

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

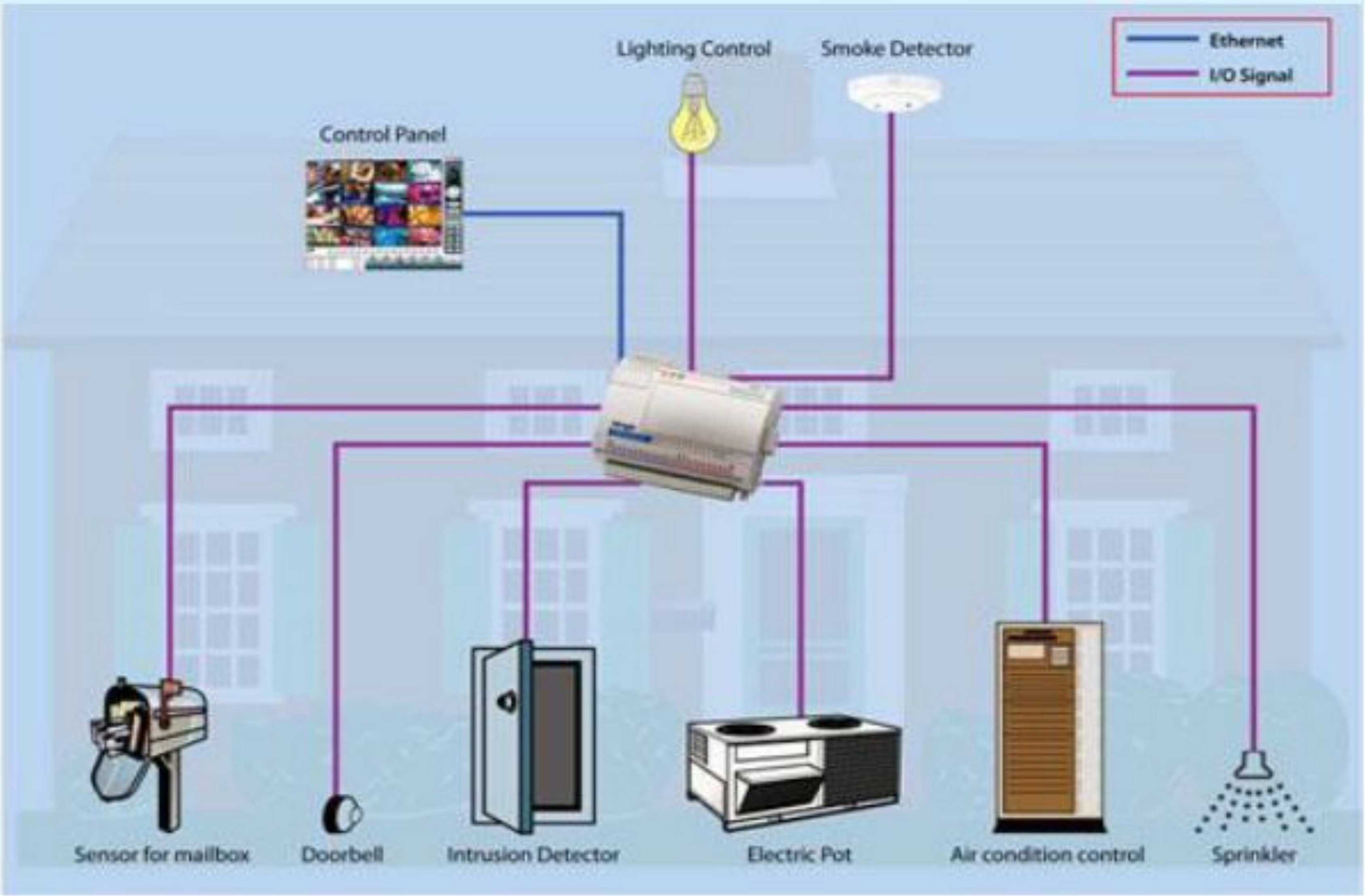
IoT Level-1


Local
Cloud
App
REST/WebSocket
Communication
REST/WebSocket
Services
Database
Controller Service
Resource
Device
Monitoring Node
performs analysis, stores data

IoT Level-1 Example – Home Automation

- The system consists of a **single node** that allows controlling the **lights** and **appliances** in home remotely
- **Electronic relay switch** is used to interface the devices
- Status information of each lights and appliances is maintained in a **local database**
- **Application** is deployed **locally**
- This level consists of **air conditioner, temperature sensor,** data collection and analysis and control & monitoring app
  - The data sensed in stored **locally**
  - The data analysis is done **locally**
  - Monitoring & Control is done using Mobile app or web app
  - The data generated in this level application is not huge
  - All the control actions are performed through internet
- **Example**
  - Room temperature is monitored using temperature sensor and data is stored/ analyzed locally.
  - Based on analysis made, control action is triggered using mobile app or it can just help in status monitoring.

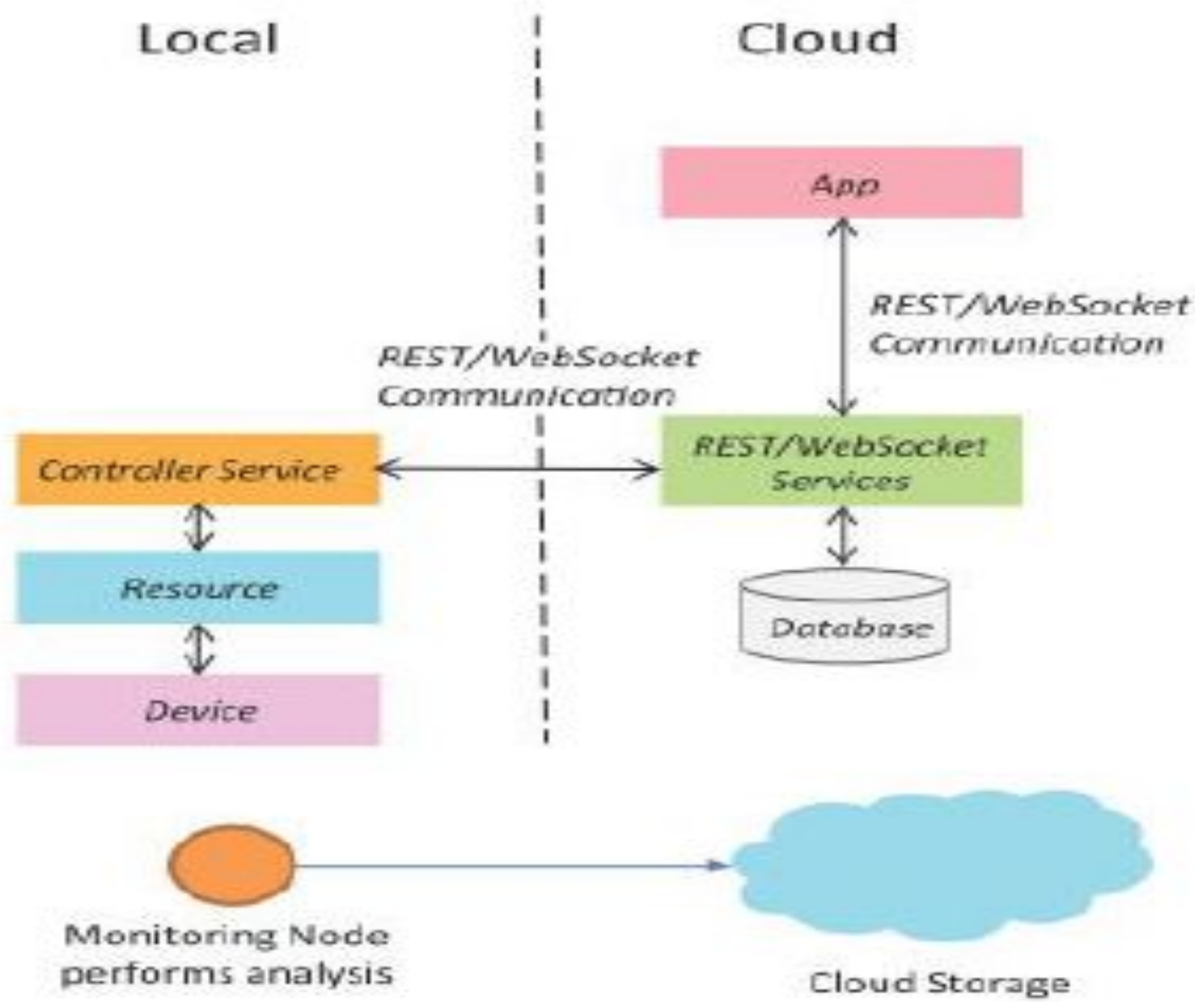
IoT Level-1 Example – Home Automation



IoT Level-2

- 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 **application** is usually **cloud-based**
- Level-2 IoT systems are suitable for solutions where
  - **the data involved is big**
  - the primary **analysis** requirement is **not computationally intensive** and **can be done locally itself.**

IoT Level-2

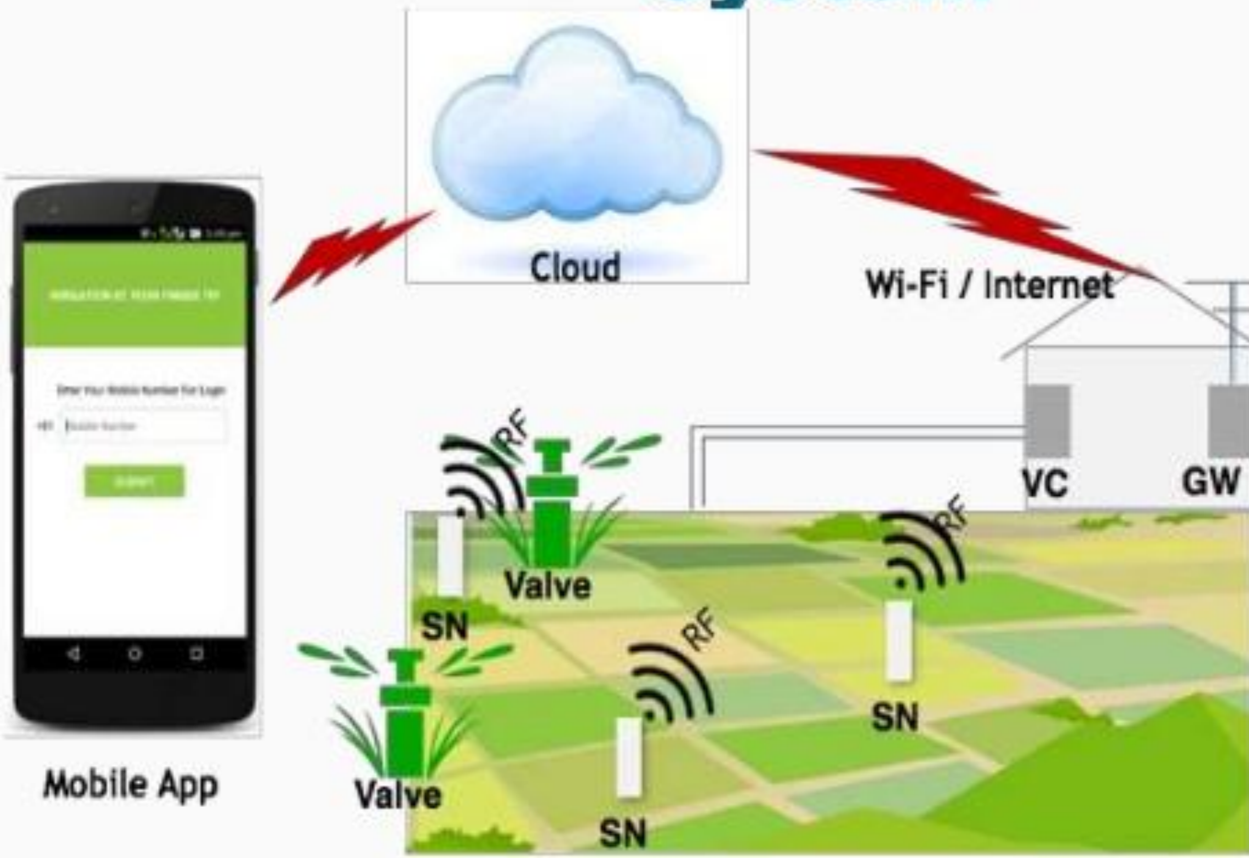

Local
Cloud
App
REST/WebSocket Communication
REST/WebSocket Communication
Controller Service
REST/WebSocket Services
Resource
Database
Device
Monitoring Node performs analysis
Cloud Storage

IoT Level-2

- It consists of air conditioner, temperature sensor, Big data (Bigger than level -1, data analysis done here) , cloud and control & monitoring app
- Level-2 is complex compare to level-1
- Rate of sensing is faster compare to level-1
- Level- 2 has voluminous size of data cloud storage is used
- Data analysis is carried out locally
- Cloud is used for only storage purpose
- Based on data analysis, control action is triggered using web app or mobile app
- **Examples:** Agriculture applications, room freshening solutions based on odour sensors etc.

IoT – Level 2 Example
Smart Irrigation

IOT based Smart Irrigation System

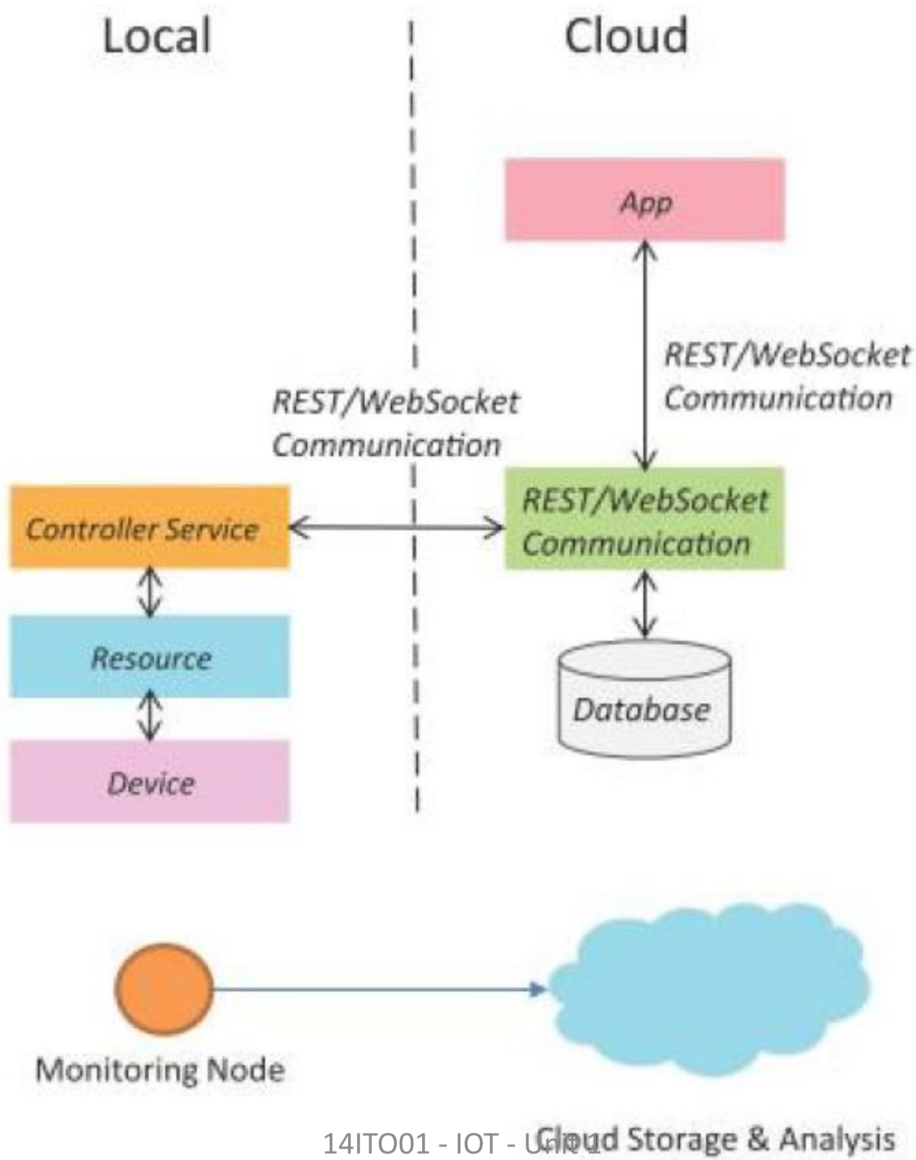


SN: Sensor Node; GW: Gateway; VC: Valve Controller; RF: Radio Frequency

Level 3

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

IoT Level-3


Local
Cloud
App
REST/WebSocket Communication
REST/WebSocket Communication
Controller Service
REST/WebSocket Communication
Resource
Database
Device
Monitoring Node
Cloud Storage & Analysis

Level 3 Example – Tracking Package Handling

- The system consists of a **single node** (**package**)
- That monitors the **vibration levels** for a package being sipped.
- The device in this system uses **accelerometer** and **gyroscope** sensors for monitoring vibration levels.
- The **controller system** sends the sensor data to the cloud using web sockets.
- The **data stored** in the cloud and **visualized** using cloud based application.
- The **analysis components** in the cloud can **trigger alerts** if the vibration levels greater than the threshold.

IoT – Level 3 Example: Tracking Package Handling

Sensors used

Accelrometer

sense movement or vibrations

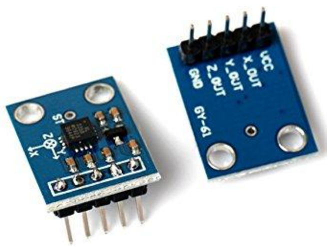
Gyroscope

Gives orientation info

Websocket service is used because sensor data can be sent in real time.

Accelerometer Sensor

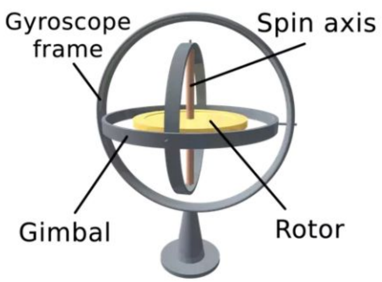
- Accelerometers are electromechanical devices that measure **acceleration**, the **rate of change in velocity of an object**. In other words, it's devices used to respond to any **vibrations** associated with movement.
- Uses:
  - **Compass/Map applications** on your smartphone devices (iPhones, Andriod, etc.) through axis based sensing
  - **Tilt sensing**; iPhone uses an accelerometer to sense whether the phone is being held in portrait or landscape mode
  - **Earthquake detection**
  - **Fall sensing**
  - **Medical devices** such as artificial body parts
  - **Fitness trackers/wearables**
  - **Games/applications t**hat require motion sensing (Wii, Kinect, etc.)

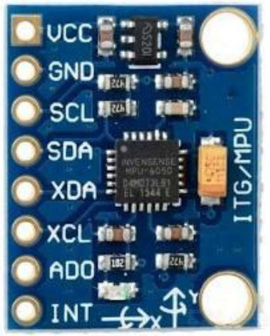


- Note: Accelerometers are most **commonly used** to detect position, velocity, vibration, and to determine orientation.

Gyroscope Sensor

- Gyroscope is a device used for measuring **rotational changes** or maintaining orientation
- It's based on the principle of **preserving angular momentum**.
- A typical gyroscope contains a **rotor** that's suspended inside three rings called the **gimbals**.
- It works through the **precession effect**, allowing gyroscopes to defy gravity when the **spin-axis** is rotated
- This means that instead of falling over from the force of gravity, it automatically adjusts itself sideways
- Uses:
  - **Aircrafts**
  - **Space stations**
  - **Stability in vehicles**; motorcycles, ships
  - **Inertial guidance systems**
  - Consumer electronics through MEMS gyroscopes (Most mid-range to higher-end **Andriod phones**)

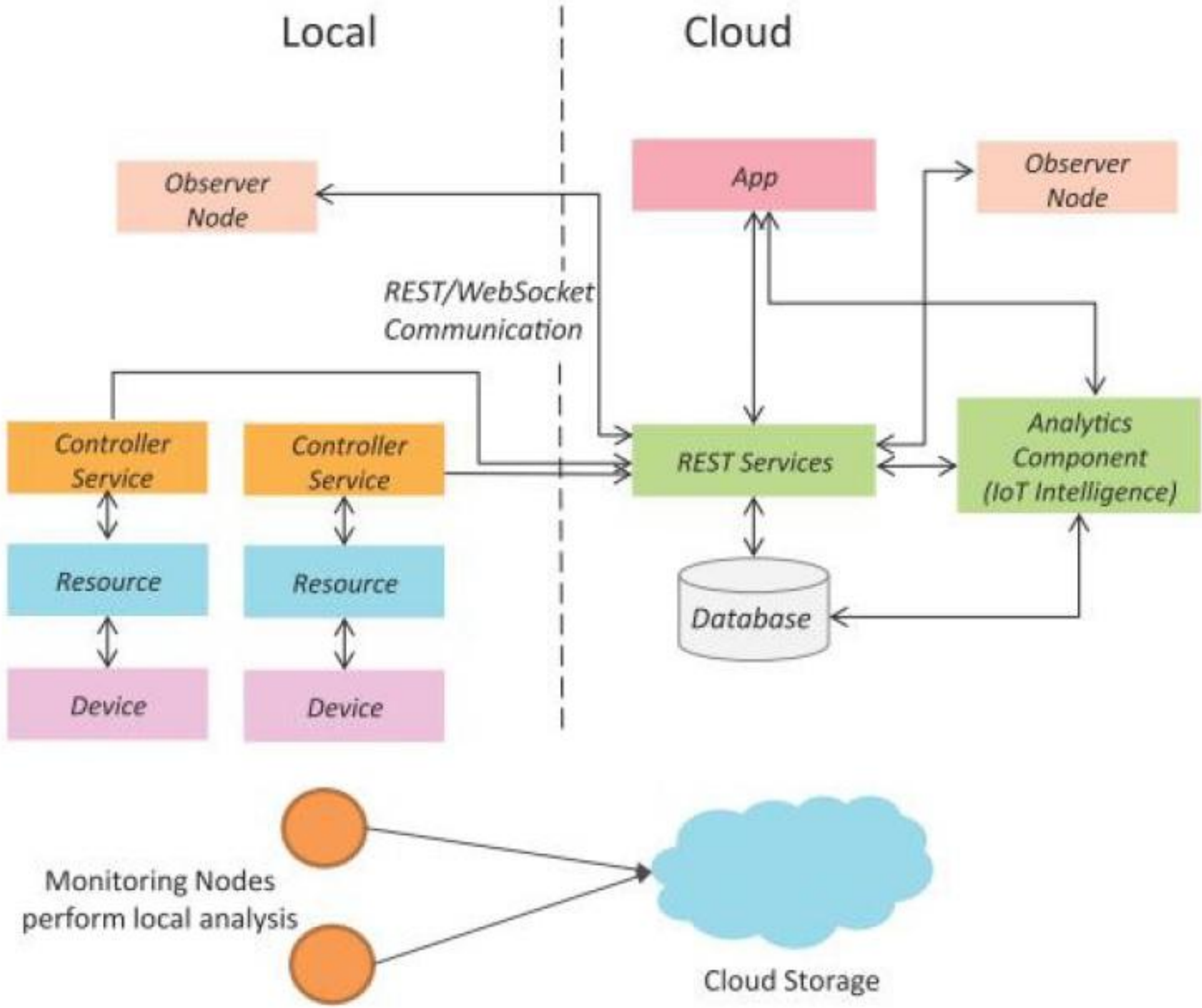




Level 4

- A level-4 IoT system has **multiple nodes** that perform **local analysis**
- **Data** is stored in the **cloud** and application is **cloud-based**
- 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
  - **Data** involved is **big** and
  - **Analysis** requirements are **computationally intensive**

IoT Level-4


Local
Cloud
Observer Node
App
Observer Node
REST/WebSocket Communication
Controller Service
Controller Service
REST Services
Analytics Component (IoT Intelligence)
Resource
Resource
Database
Device
Device
Monitoring Nodes perform local analysis
Cloud Storage

Level 4 Example – Noise Monitoring

- The system consists of **multiple nodes** placed in different locations.

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

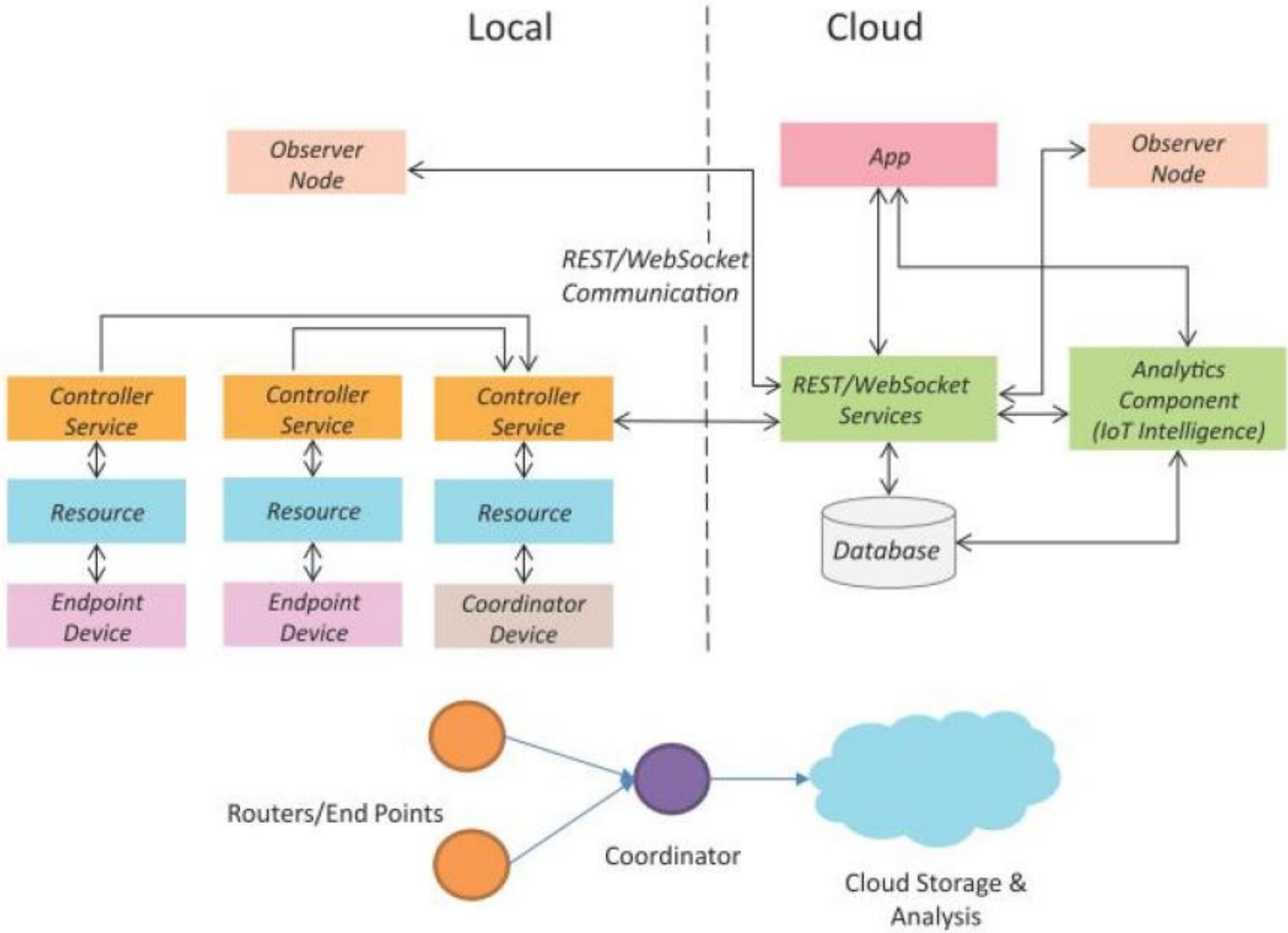
IoT – Level 4 Example: Noise Monitoring

Sound Sensors are used

Level 5

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

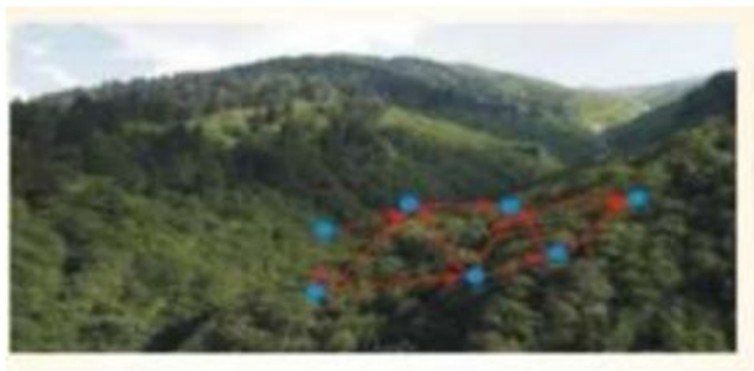
IoT Level-5

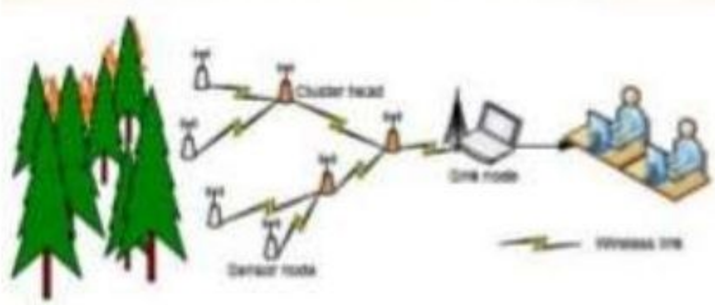


Local
Cloud
Observer Node
App
Observer Node
REST/WebSocket Communication
Controller Service
Controller Service
Controller Service
REST/WebSocket Services
Analytics Component (IoT Intelligence)
Resource
Resource
Resource
Database
Endpoint Device
Endpoint Device
Coordinator Device
Routers/End Points
Coordinator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

IoT – Level 5 Example: Forest Fire Detection

Detect forest fire in early stages to take action while the fire is still controllable.
Sensors measure the temperature, smoke, weather, slope of the earth, wind speed, speed of fire spread, flame length

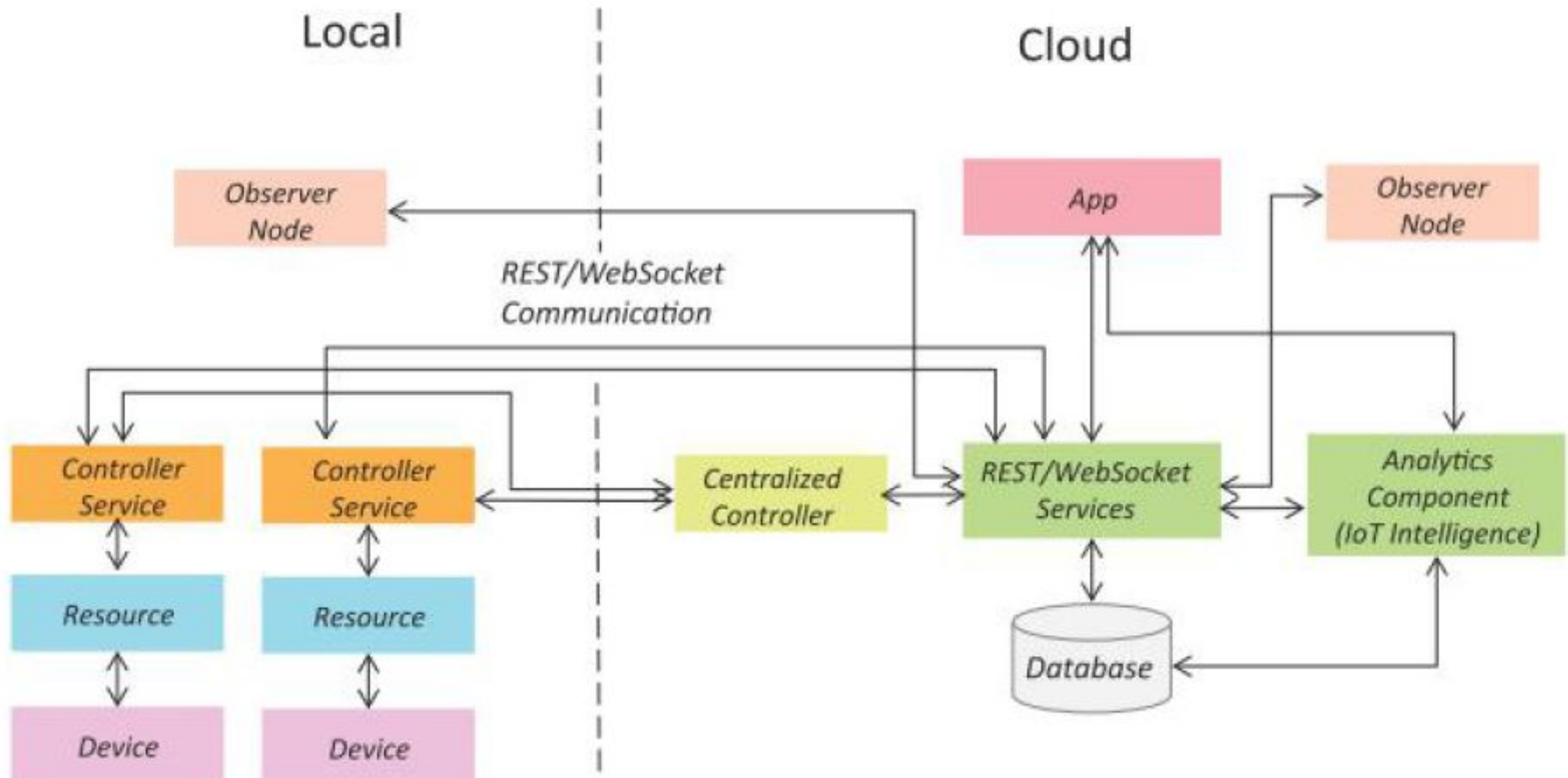


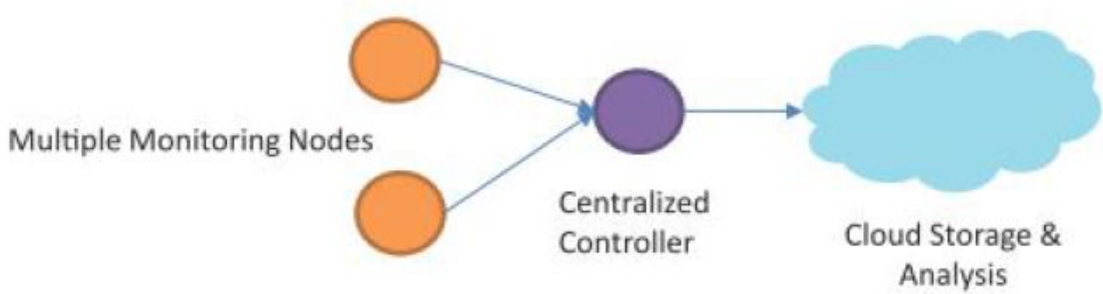


Level 6

- A level-6 IoT system has **multiple independent end nodes** that perform sensing and/or actuation and send data to the cloud
- **Data** is stored in the **cloud** and **application** is **cloud-based**
- 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**

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

Level 6 Example – Weather Monitoring System

- The system consists of **multiple nodes** placed in different locations for monitoring **temperature**, **humidity** and **pressure** in an area
- The end nodes are equipped with various sensors,
  - **Temperature**
  - **Pressure**
  - **Humidity**
- The end nodes send the data to the cloud in real time using **websockets**
- The **data stored** in **cloud database**
- The **analysis of data** is done in the **cloud** to aggregate the data and make predictions
- **Cloud based application** is used for visualizing the data

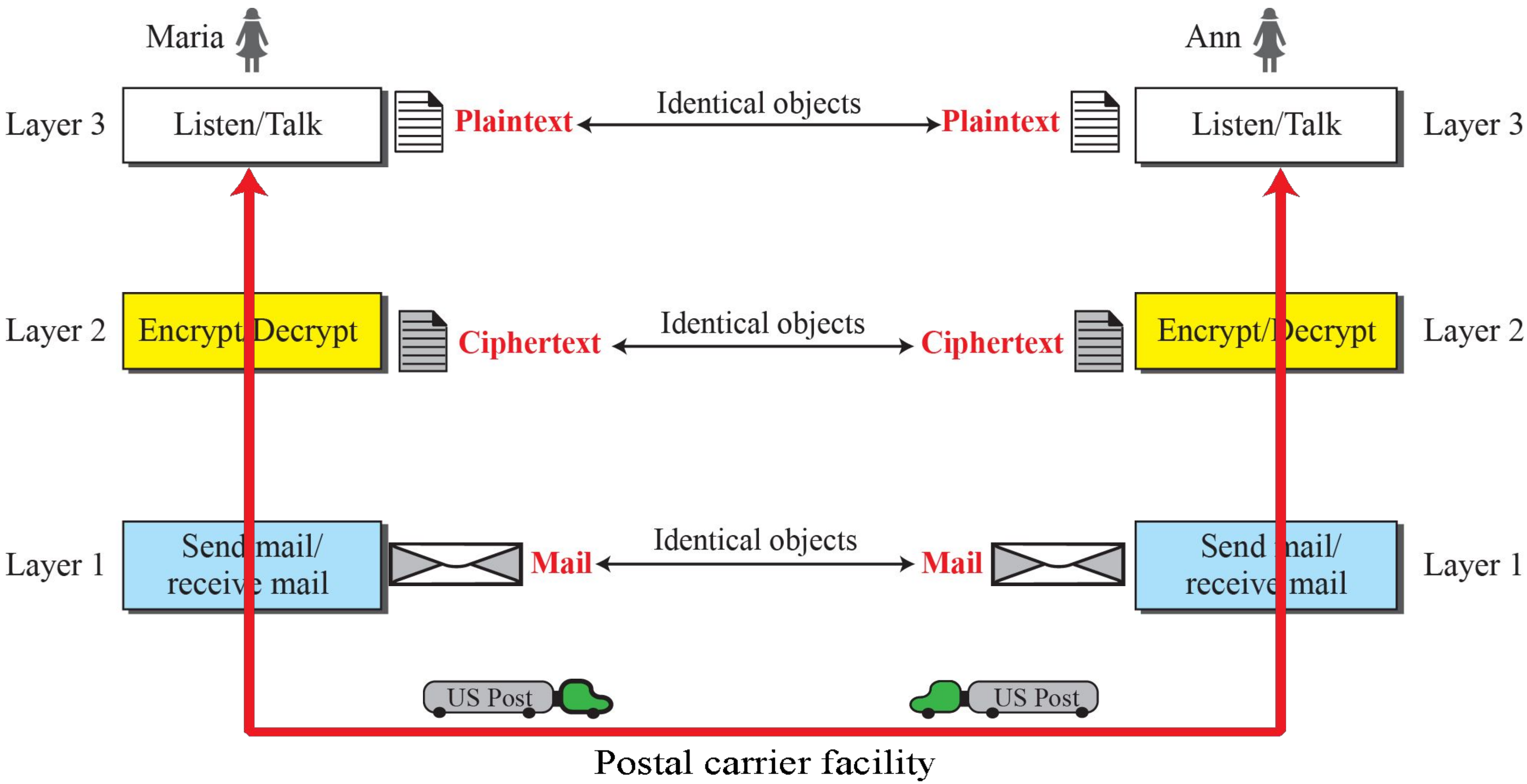
IoT – Level 6 Example: Weather Monitoring System

Sensors used

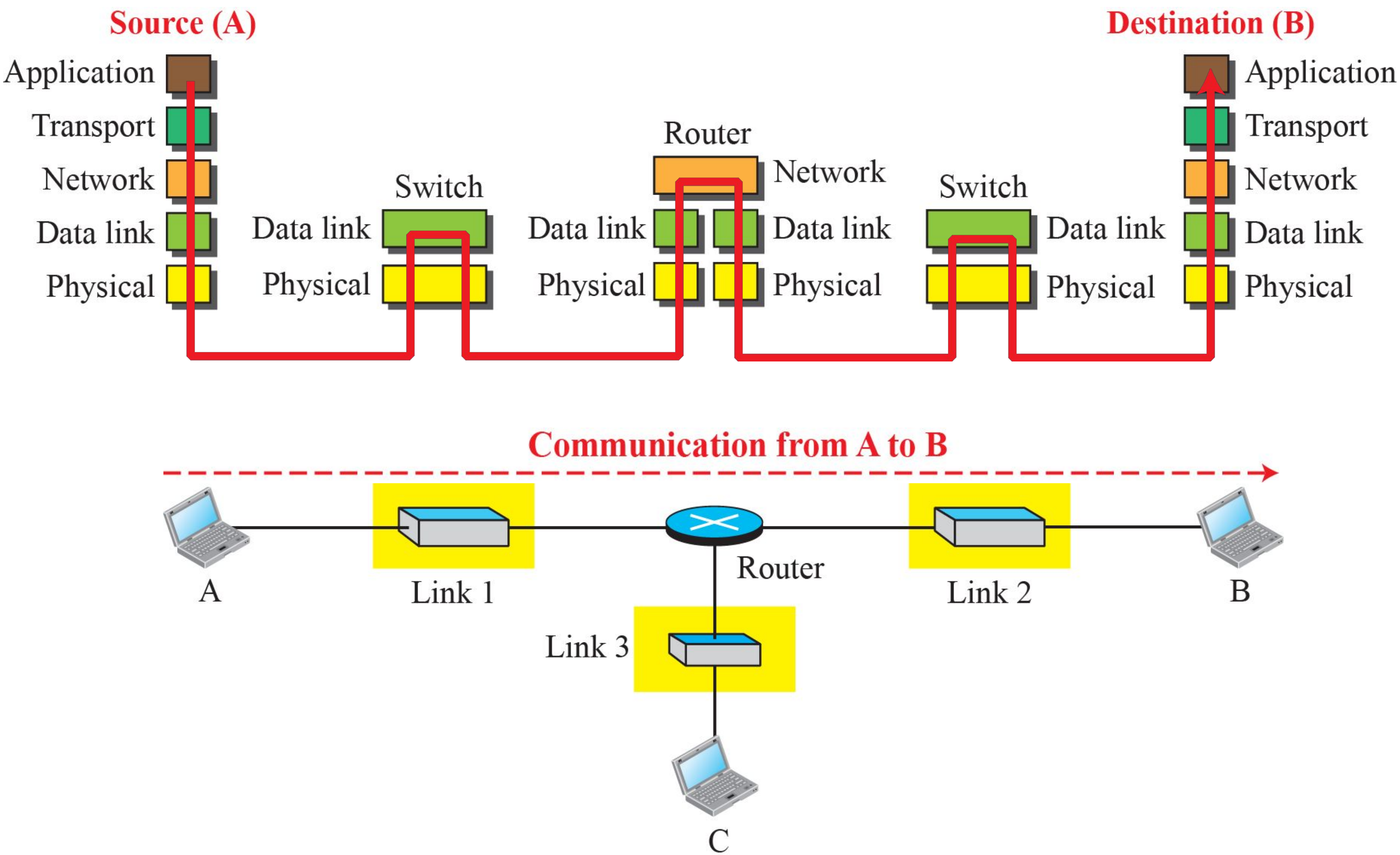
- Wind speed and direction
- Solar radiation
- Temperature (air, water, soil)
- Relative humidity
- Precipitation
- Snow depth
- Barometric pressure
- Soil moisture

A single-layer protocol

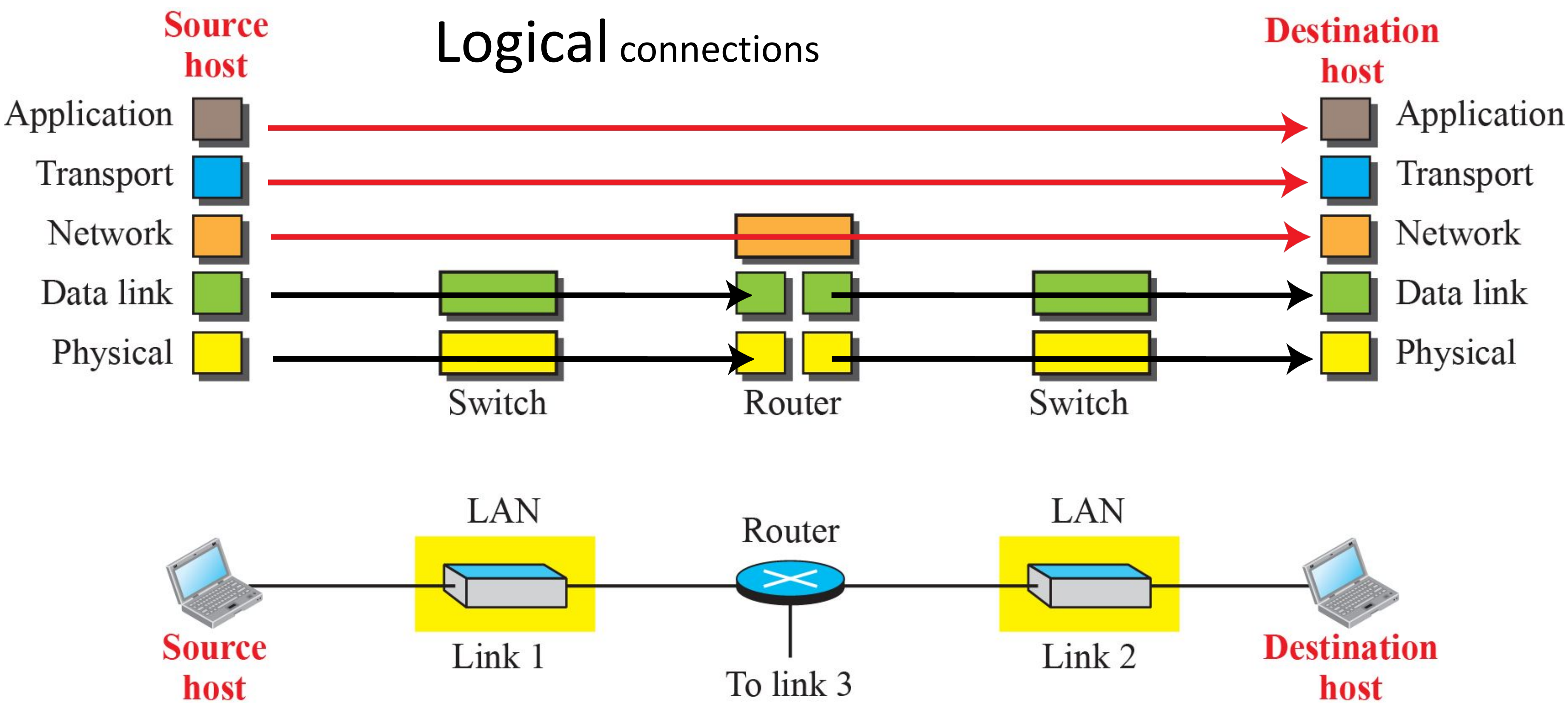
A three-layer protocol



Communication through an internet

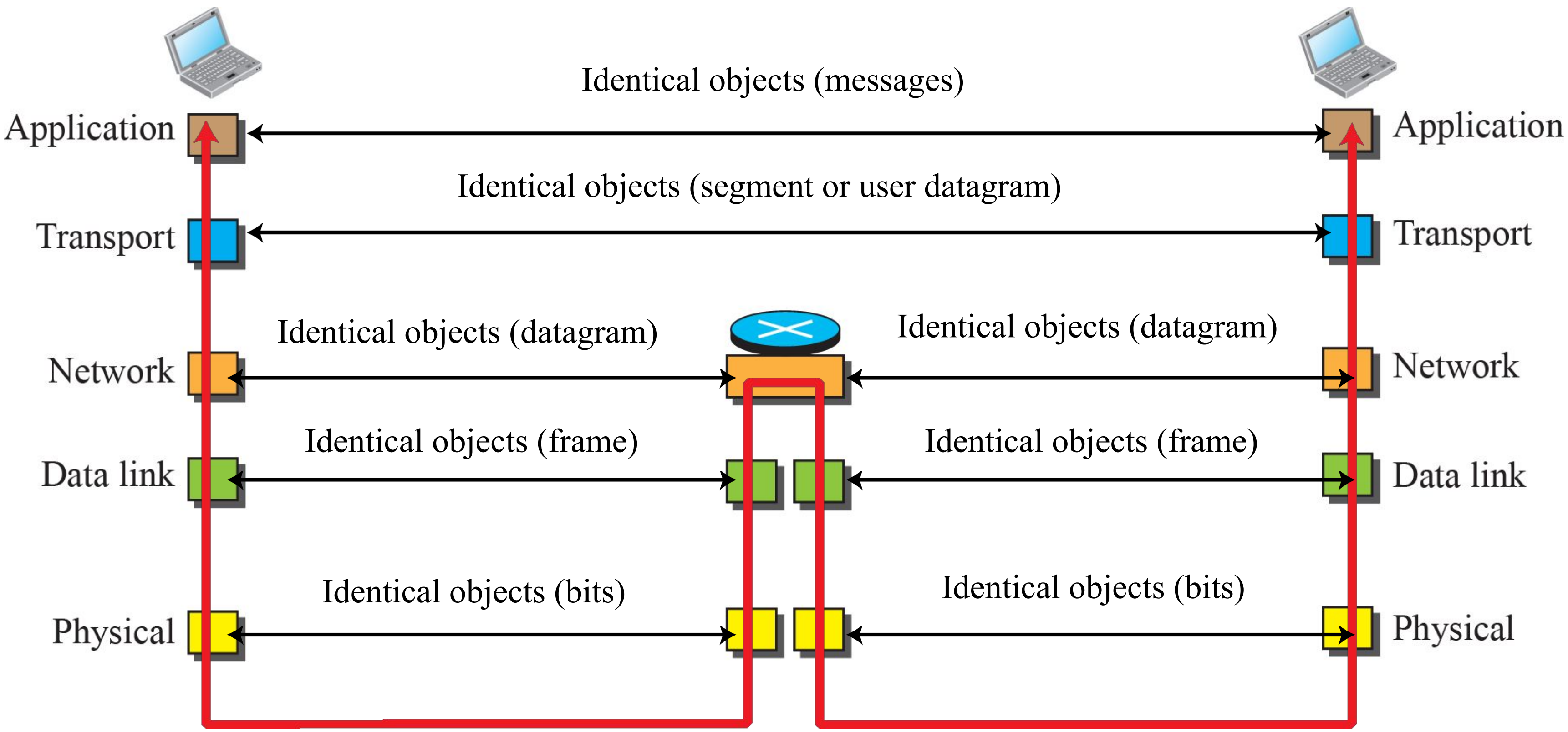


Logical connections between layers in TCP/IP



Identical objects in the TCP/IP protocol suite

Notes: We have not shown switches because they don't change objects.



Addressing in the TCP/IP protocol suite

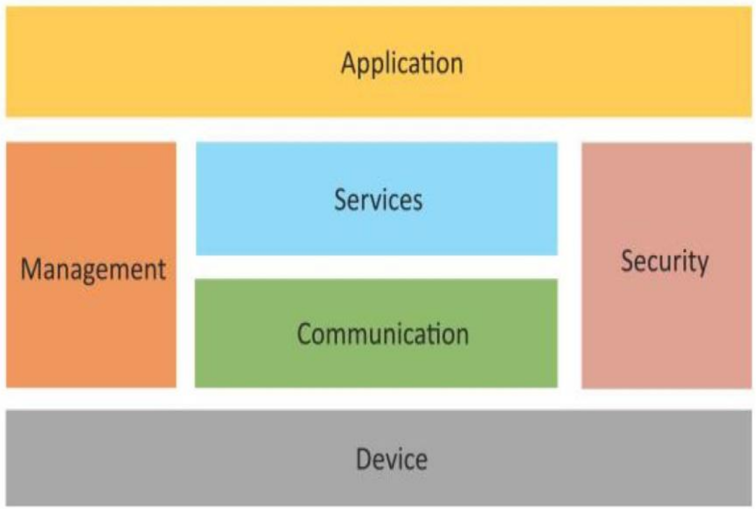
Packet names	Layers	Addresses
Message	Application layer	Names
Segment / User datagram	Transport layer	Port numbers
Datagram	Network layer	Logical addresses
Frame	Data-link layer	Link-layer addresses
Bits	Physical layer	

Logical Design of IoT

- Abstract representation of the entities and processes without going into the low-level specifics of the implementation
  - Functional Blocks of IoT System
  - IoT Communication Models
  - IoT Communication APIs

Functional Blocks of IoT

- An IoT system comprises of a number of functional blocks that provide the system the capabilities for
  - **Identification**
  - **Sensing**
  - **Actuation**
  - **Communication and**
  - **Management.**



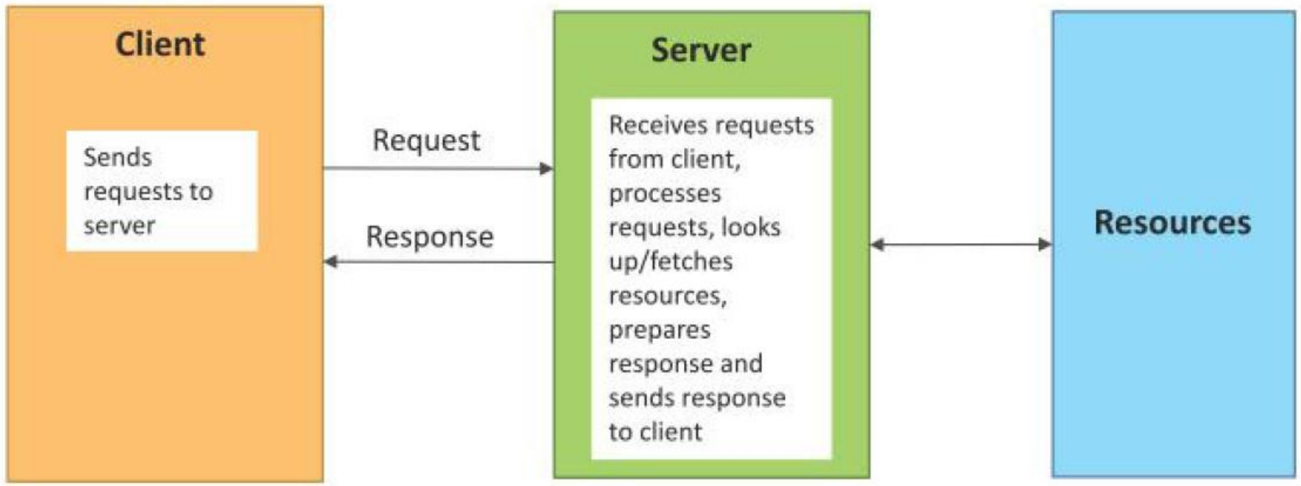
- **Device:** Provides **sensing**, **actuation**, **monitoring** and control functions
- **Communication:** It handles the **communication** for the IoT System
- **Services:** Services for device **monitoring**, device **control**, data **publishing** and device **discovery**
- **Management:** It provides various **functions** to govern the IoT system.
- **Security:** Secures the IoT system and provide functions such as **authentication**, **authorization**, **message integrity** and **data security**.
- **Application:** It provides an **interface** that the user can use to control and monitor various aspects of IoT System.

IoT Communication Models

- There are **four types** of IoT communication models available.
  - **Request-Response** communication model
  - **Publish-Subscribe** communication model
  - **Push-Pull** communication model
  - **Exclusive Pair** communication model

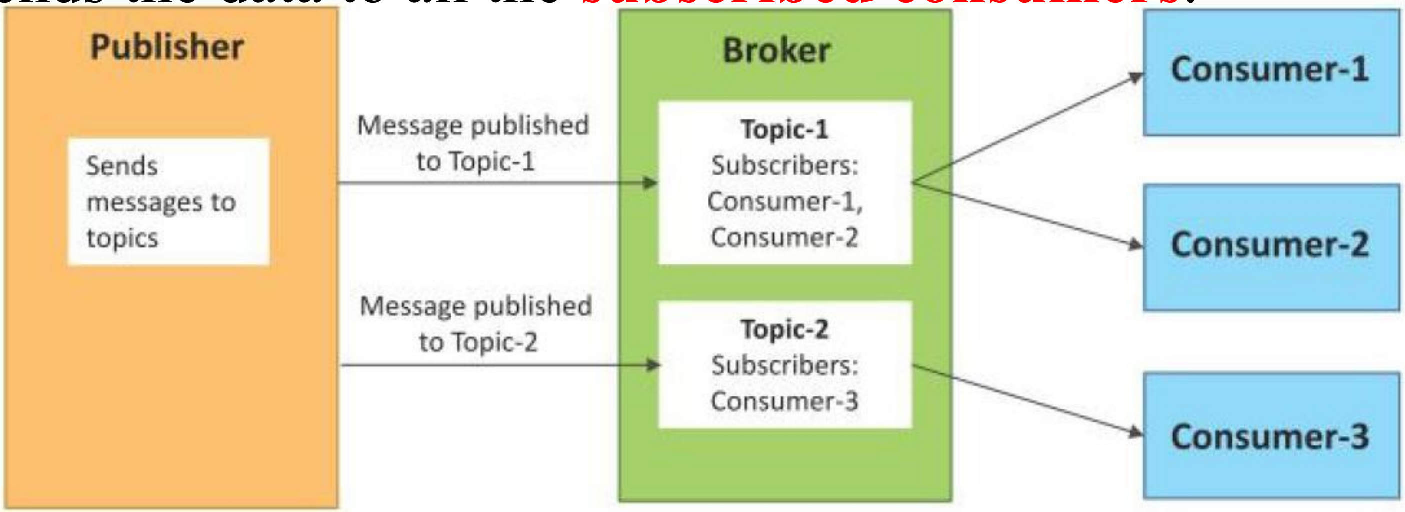
Request-Response communication model

- **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
  - Sends the response to the client.
- It is a **stateless communication** model.



Publish-Subscribe communication model

- This model 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**.



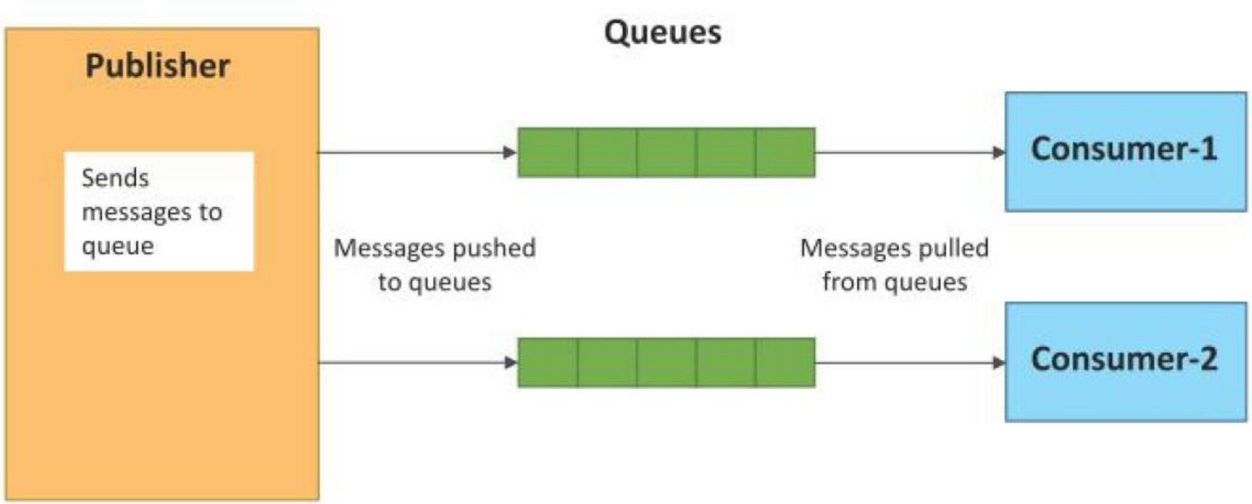
Publish-Subscribe Communication Model

- **Example: Home Heating System**

  It has a temperature sensor, a heating unit, and a central control unit.
- Temperature in the house. Published by the temperature sensor and subscribed to by the central control unit. Has the value of the current in-house temperature.
- Heating Request. Published by the control unit and subscribed to by the heating unit. Has the value of either "On" or "Off".

Push-Pull communication model

- In this 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 : **Decouples** the message between the producers and consumers.
- Queues also act as a **buffer** (Mismatch between the rate at which the producers push data and the rate at which the consumers pull data)

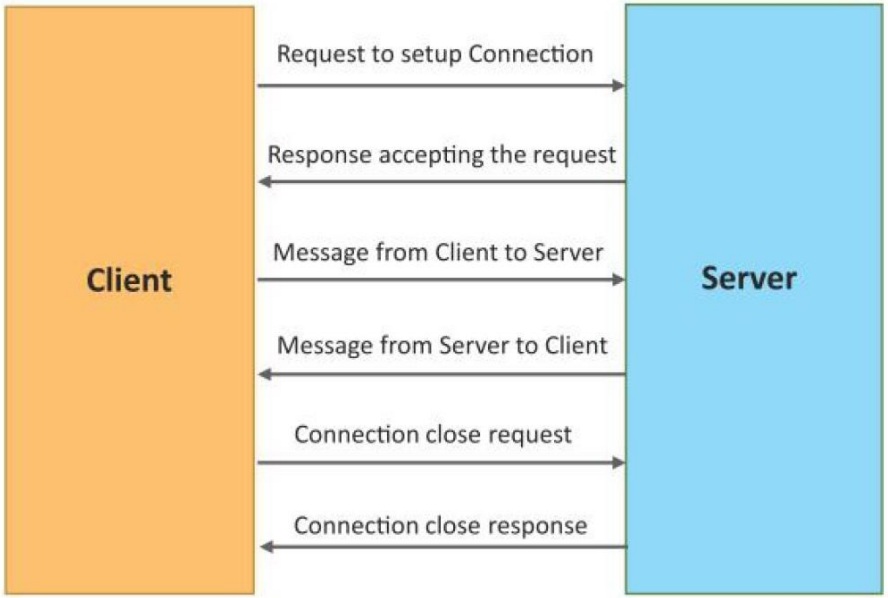


PUSH –PULL Example

- Push communication : Sending information to stakeholders. This method is useful for time sensitive information and when feedback is expected.
- Push is a kind of broadcast and as the sender control it, determine which stakeholders receive it, and at what time
- Push communication ---Send information to a large pool of stakeholders quickly.
- Push : memos, letters, voice mails, etc
- Pull communication method ----Stakeholders pull the information as per their requirements.
- Pull : blog, intranet, bulletin board, etc.

Exclusive Pair communication model

- Exclusive Pair is a **bidirectional**, **fully duplex** communication model that uses a **persistent connection** between the client and server.
- **Once the connection is setup 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.

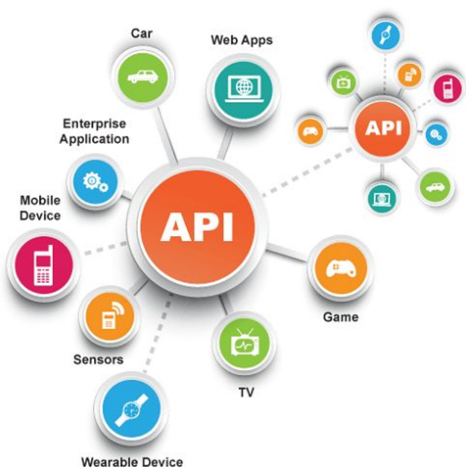


IoT Communication APIs

- An Application Program Interface (API) is a set of **routines, protocols** and **tools** for building software applications.
- API specifies how software components should interact.
- Two Communication APIs,

  REST – Based Communication APIs

  WebSocket – Based Communication APIs



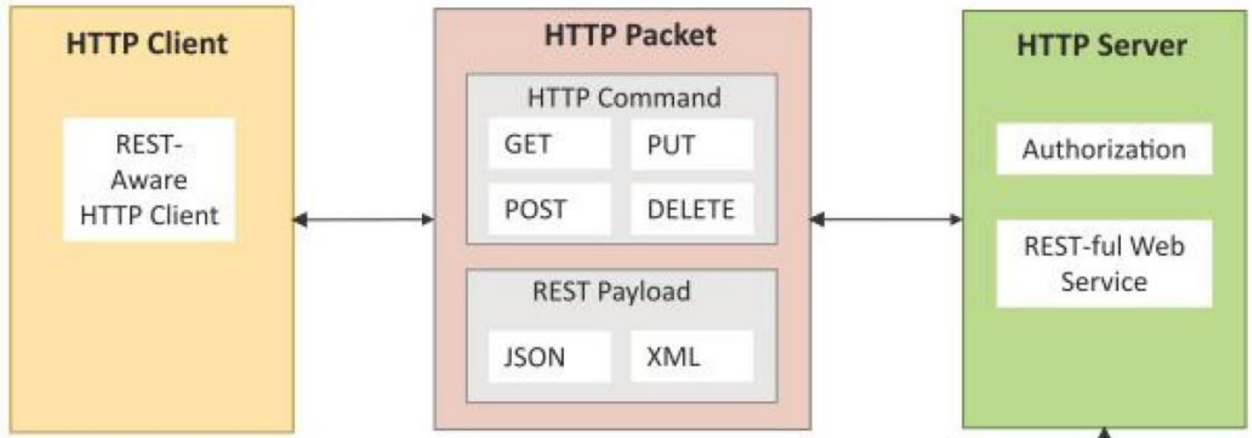
REST – Based Communication APIs

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

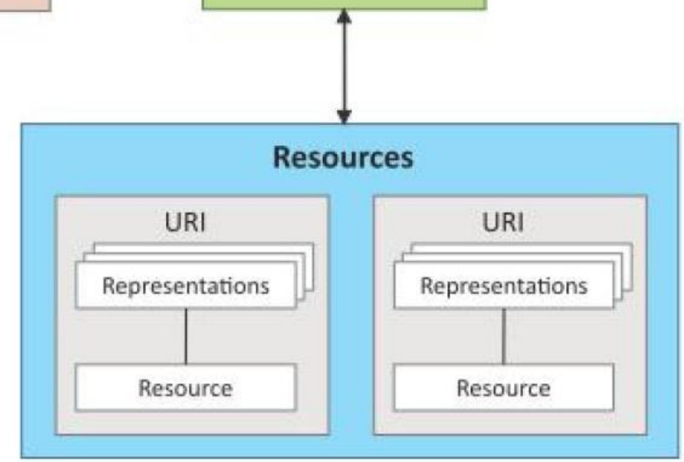
REST – Based Communication APIs

- The REST architectural constraints are as follows:
  - **Client – Server** : Separation, Independent
  - **Stateless** : Session state is kept entirely on the client
  - **Cache-able** : Improve efficiency and scalability.
  - **Layered System** : Intermediate Systems
  - **Uniform Interface** : Method of communication
  - **Code on demand** : Server can provide executable code for clients. (**It is optional**)

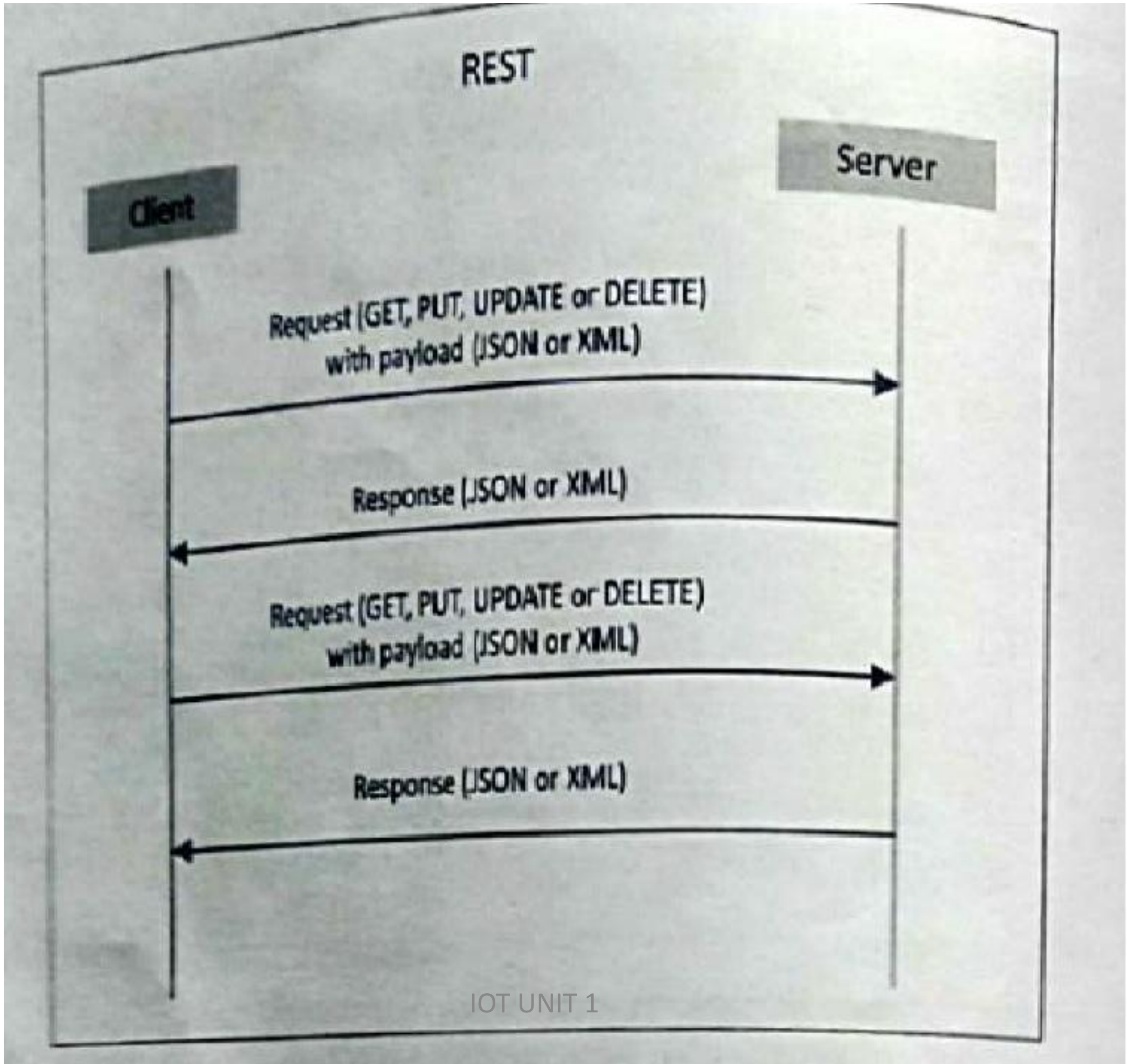
Communication b/w Client and Server using REST APIs



- RESTful Web Service:
  - It is a web API implemented using **HTTP** and **REST** principles
  - It is a collection of **resources** which are represented by **URIs**
  - It has a **base URI** (Ex: http://example.com/api)
  - Client send requests to these URIs using **HTTP methods** (**GET,PUT, POST** or **DELETE**)
  - It can support various **Internet media types** (**JSON**)
  - **IPSO Alliance** has published and application framework that defines a RESTful Design.



Interaction in the Request – Response Model used by REST



HTTP Methods

HTTP Method	Resource Type	Action	Example
GET	Collection URI	List all the resources in a collection	http://example.com/api/tasks/ (list all tasks)
GET	Element URI	Get information about a resource	http://example.com/api/tasks/1/ (get information on task-1)
POST	Collection URI	Create a new resource	http://example.com/api/tasks/ (create a new task from data provided in the request)
POST	Element URI	Generally not used	
PUT	Collection URI	Replace the entire collection with another collection	http://example.com/api/tasks/ (replace entire collection with data provided in the request)
PUT	Element URI	Update a resource	http://example.com/api/tasks/1/ (update task-1 with data provided in the request)
DELETE	Collection URI	Delete the entire collection	http://example.com/api/tasks/ (delete all tasks)
DELETE	Element URI	Delete a resource	http://example.com/api/tasks/1/ (delete task-1)

Websocket – Based Communication APIs

- WebSocket APIs allow **bidirectional**, **full duplex** communication between clients and servers
- WebSocket APIs follow the **exclusive pair communication** model
- Websockets reduces the **network traffic** and **latency** (**no overhead for connection setup and termination for each message**)
- Websockets is suitable for **IoT applications** that have **low latency** or **high throughput** requirements

Websocket – Based Communication APIs