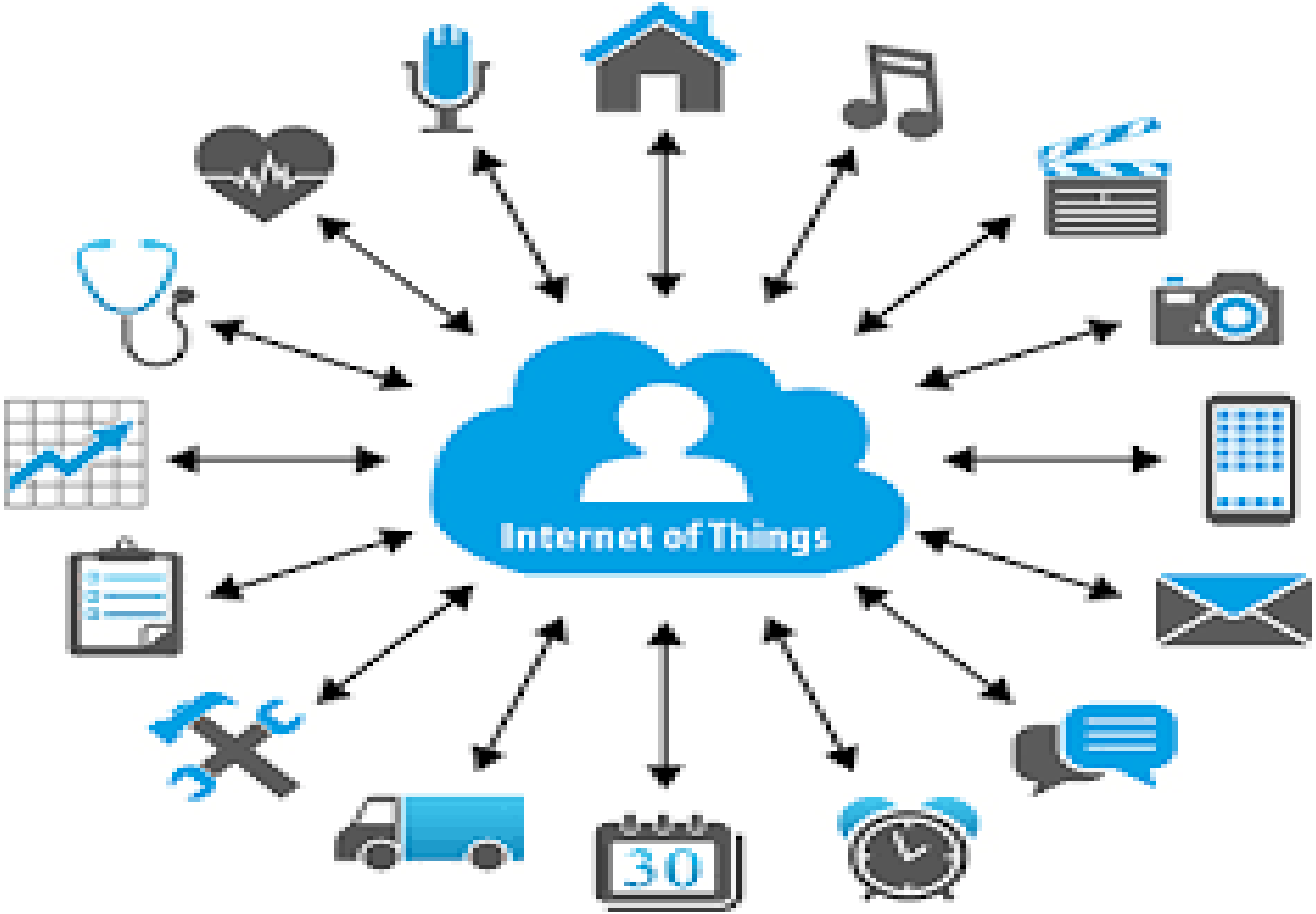
Fundamentals of IoT

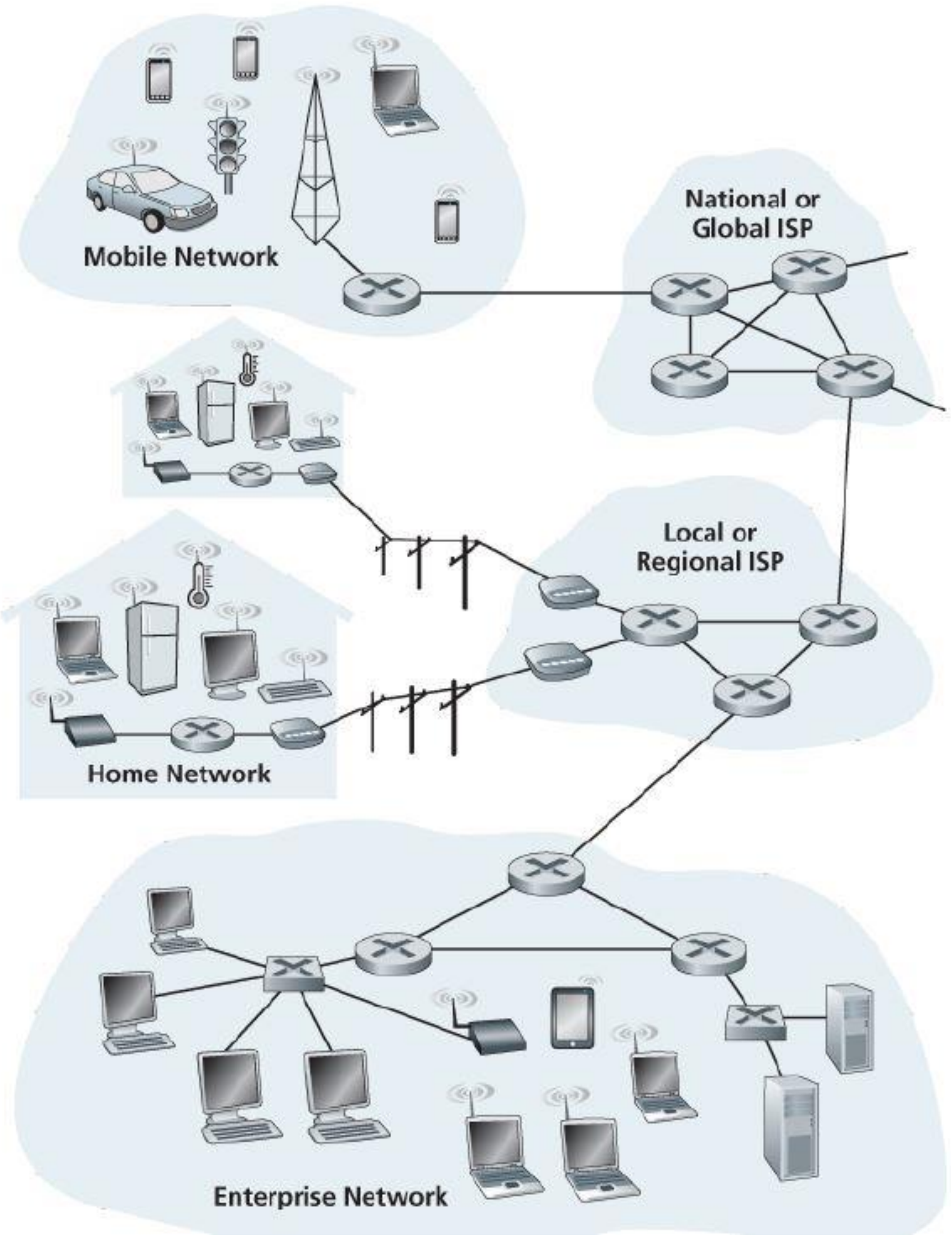
Dr. Shridhar Sanshi


Internet of Things
30

What is Internet?

- A global network of computing resources.
- Physical collection of routers and circuits as a set of shared resources.
- Common definition

A network of networks based on the TCP/IP communications protocol.



Internet-Based Services

Some of the services offered by the Internet are:

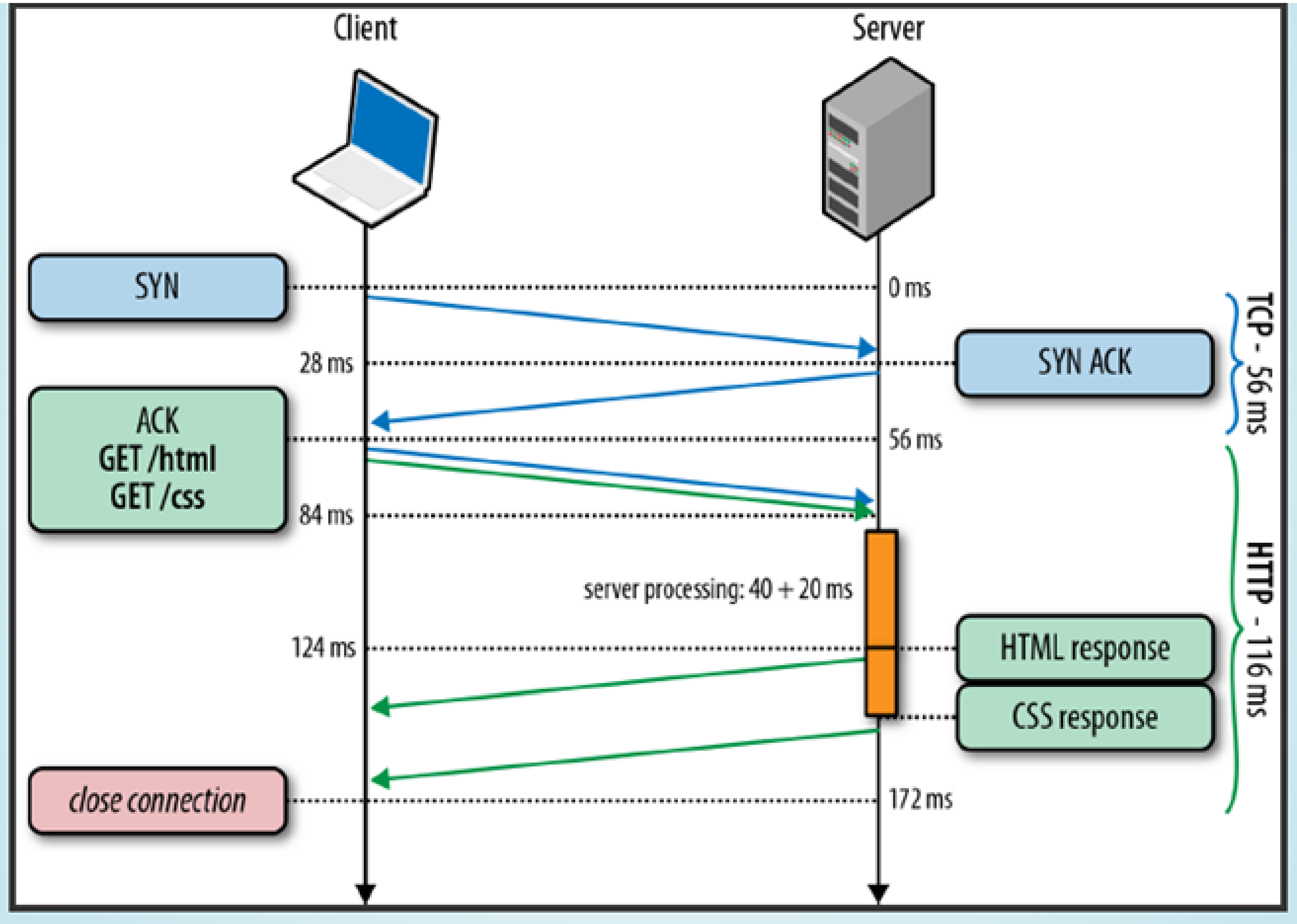
- Email
- Telnet
- FTP
- UseNet news
- World Wide Web (WWW)

What is WWW?

- The World Wide Web—commonly referred to as WWW, W3, or the Web—is a system of interconnected public webpages accessible through the Internet.
- "the Web" consists of several components:
    - The HTTP protocol governs data transfer between a server and a client.
    - To access a Web component, a client supplies a unique universal identifier, called a URL (uniform resource locator) or URI (uniform resource identifier) (formally called Universal Document Identifier (UDI)).
    - HTML (hypertext markup language) is the most common format for publishing web documents.

What is HTTP?

- The HyperText Transfer Protocol (HTTP) is the underlying network protocol that enables transfer of hypermedia documents on the Web, typically between a browser and a server



- One-to-one communication
- Uni-Directional
- Synchronous request-response
- Scalability
- request-response based communication
- High Power Consumption

What is URL?

- Uniform Resource Locator and is used to specify addresses on the World Wide Web.
- A URL is the fundamental network identification for any resource connected to the web
- A URL will have the following format –

  protocol://hostname/other_information
- The protocol specifies how information is transferred from a link.
- The domain name is the computer on which the resource is located.
- Links to particular files or subdirectories may be further specified after the domain name.
- The directory names are separated by single forward slashes.

What is HTML?

- HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content.
- Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript).
- "Hypertext" refers to links that connect web pages to one another, either within a single website or between websites.

- What is **Web Server?**
- Every Website sits on a computer known as a Web server
- This server is always connected to the internet.
- Every Web server that is connected to the Internet is given a unique address

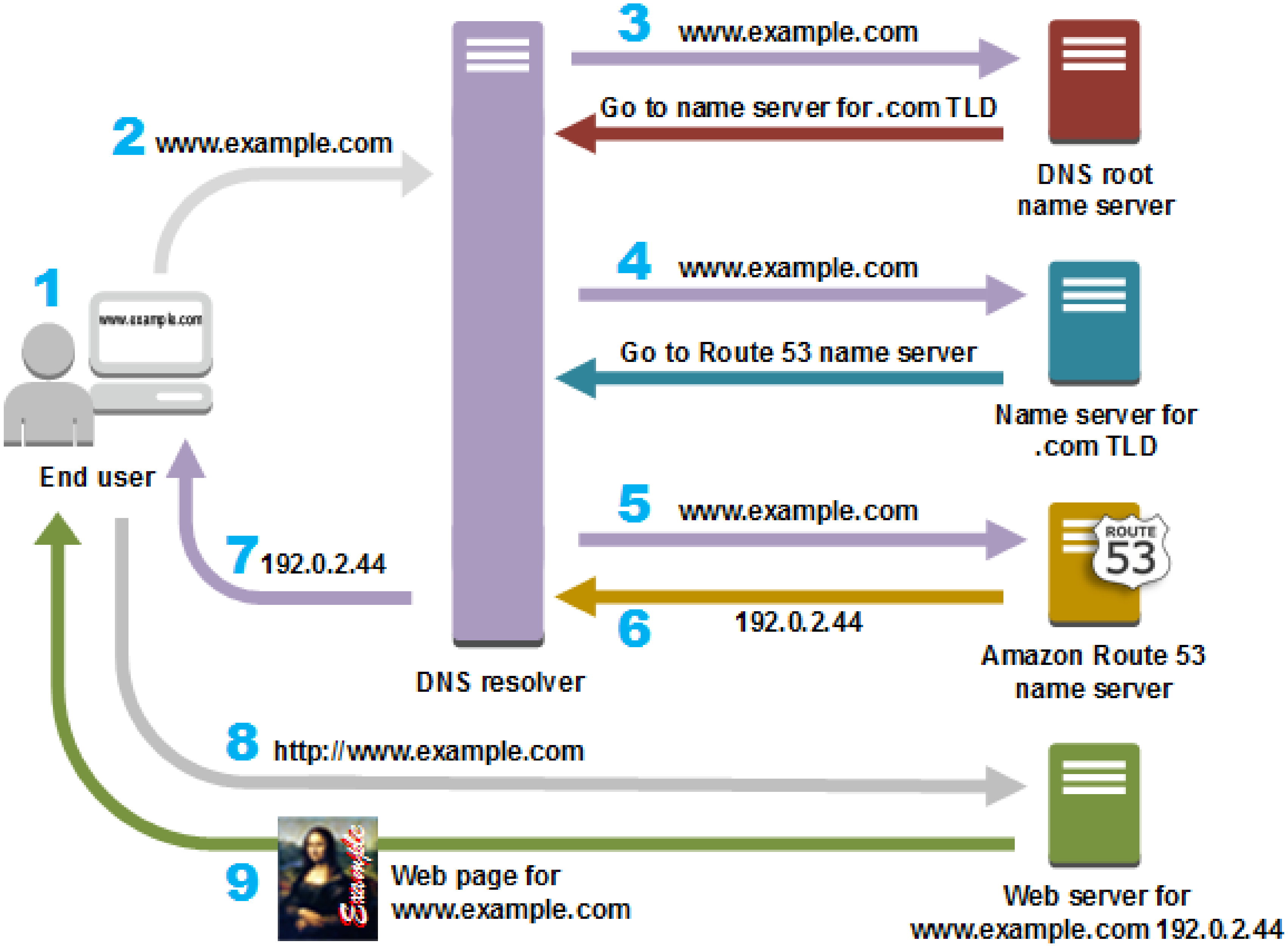
  For example, 68.178.157.132 or 68.122.35.127.
- When you register a Web address, also known as a **domain name**, such as example.com you have to specify the IP address of the Web server that will host the site.

What is Web browser?

- Web Browsers are software installed on your PC.
- To access the Web you need a web browsers
- On the Web, when you navigate through pages of information this is commonly known as browsing or surfing.

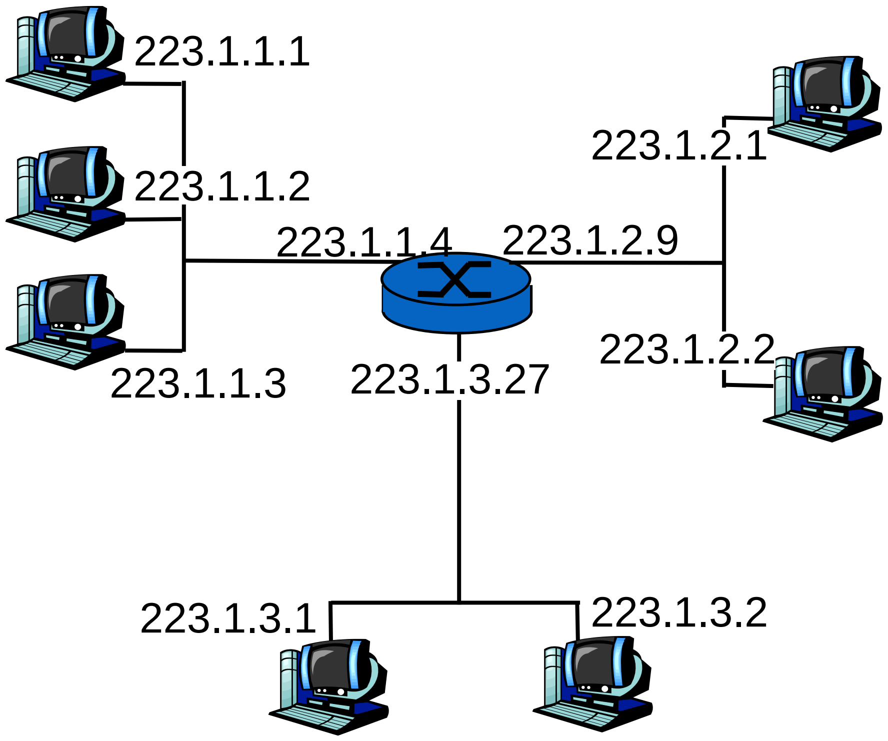
What is DNS?

- Domain Name System.
- When someone types your domain name, www.example.com
- your browser will ask the Domain Name System to find the IP that hosts your site.
- When you register your domain name, your IP address should be put in a DNS along with your domain name.
- Without doing it your domain name will not be functioning properly.


1
www.example.com
End user
2 www.example.com
DNS resolver
3 www.example.com
Go to name server for .com TLD
DNS root name server
4 www.example.com
Go to Route 53 name server
Name server for .com TLD
5 www.example.com
ROUTE 53
6 192.0.2.44
Amazon Route 53 name server
7 192.0.2.44
8 http://www.example.com
9
Example
Web page for www.example.com
Web server for www.example.com 192.0.2.44

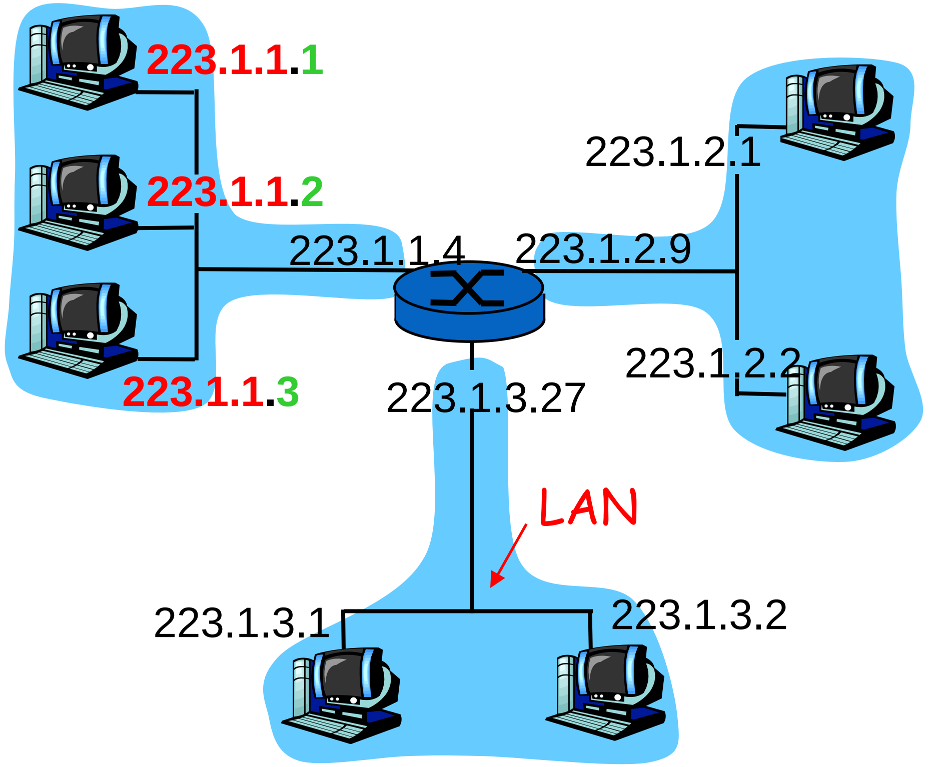
What is IP address?

- IP address: 32-bit identifier for host, router *interface*
- *interface:* connection between host/router and physical link
  - router's typically have multiple interfaces
  - host may have multiple interfaces
  - IP addresses associated with each interface

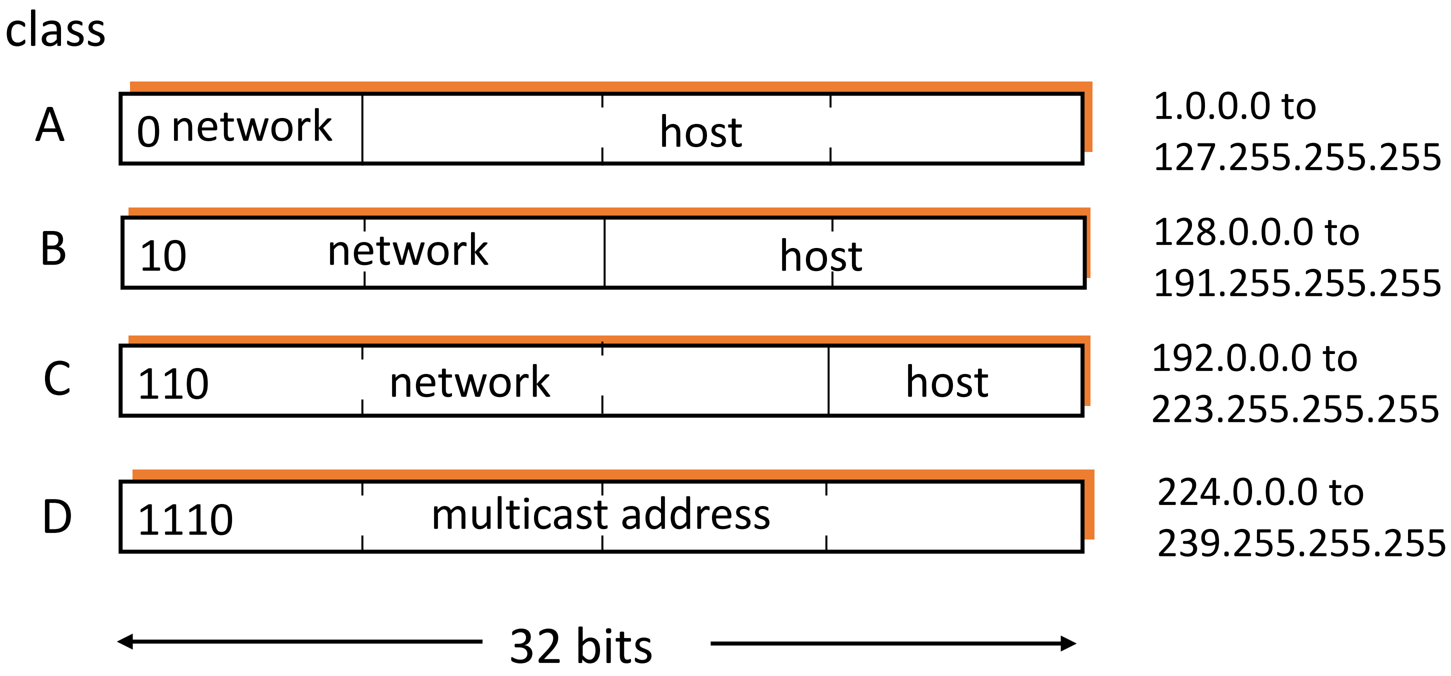


$$223.1.1.1 = \underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}$$

- IP address:
  - network part/prefix (high order bits)
  - host part (low order bits)
  - Additional hosts to 223.1.1 network would have address of 223.1.1.xxx
- *What's a network ?* (from IP address perspective)
  - device interfaces with same network part of IP address
  - can physically reach each other without intervening router



network consisting of 3 IP networks (for IP addresses starting with 223, first 24 bits are network address) “class-full” addressing: 4 shown, 5th was for future use beginning with 11110



class

A 0 network host 1.0.0.0 to 127.255.255.255

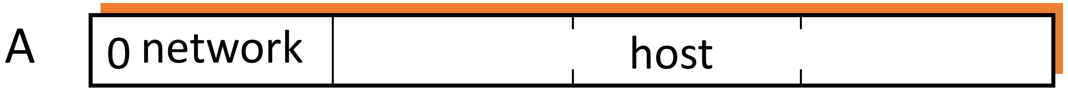
B 10 network host 128.0.0.0 to 191.255.255.255

C 110 network host 192.0.0.0 to 223.255.255.255

D 1110 multicast address 224.0.0.0 to 239.255.255.255

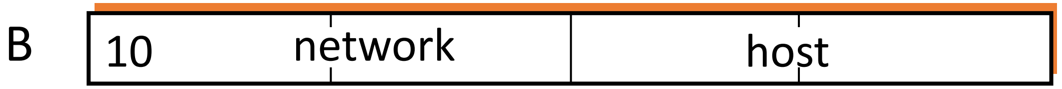
32 bits

IP Addresses (Class A, B, C. D later)



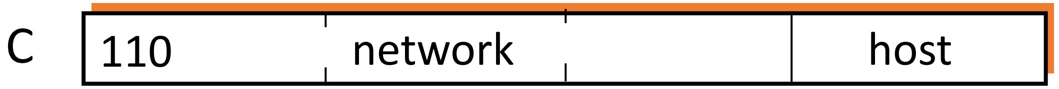
1.0.0.0 to
127.255.255.255

- 2^7 networks (first bit is 0)
- 2^(24) interfaces



128.0.0.0 to
191.255.255.255

- 2^(14) networks (first 2 bits are 10)
- 2^(16) interfaces



192.0.0.0 to
223.255.255.255

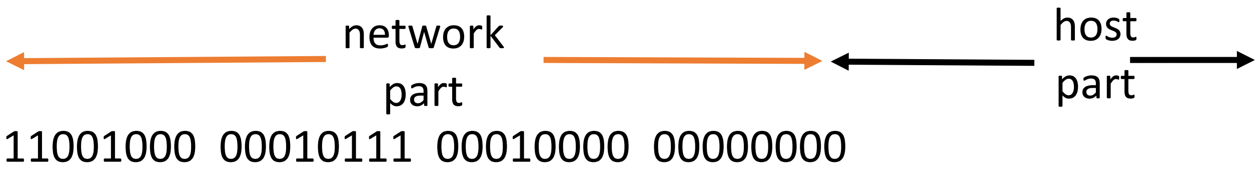
- 2^(21) networks (first 3 bits are 110)
- 2^(8) interfaces

Classful addressing

- ❑ Class A, B, C networks require 1, 2 and 3 bytes for the network portion.
- ❑ E.g., Class C networks can accommodate only 2^8-2 = 254 hosts (2 are reserved). Small for most medium to large organizations.
- ❑ However Class B supports 65,634 hosts – too large. An organization with 2000 hosts ended up with class B addressing – address space was ill used.
- ❑ Therefore in 1993, Classless Interdomain Routing (**CIDR**) was introduced.

IP addressing: CIDR (RFC 1519)

- CIDR: Classless InterDomain Routing
  - network portion of address of arbitrary length
  - address format: a.b.c.d/x, where x is # bits in network portion of address
- Classful/CIDR addressing example:
  - Prev. example with 2000 hosts. Therefore 2^16 – 2000 = 63K addresses were unused.
  - CIDR: Network part: 21 bits. Host part: 2^11 = 2048 hosts.



network part

host part

11001000 00010111 00010000 00000000

200.23.16.0/21

IP addresses: how to get one?

How does *host* get IP address?

- hard-coded by system admin in a file
  - Wintel: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from a server
  - “plug-and-play”

(more shortly)

IP addresses: how to get one?

How does *network* get network part of IP addr?

- gets allocated portion of its provider ISP’s address space

ISP's block	11001000 00010111 00010000 00000000	200.23.16.0/20

(**allocated to ISP**). **It is divided into 8 equal sized blocks**.

Organization 0	11001000 00010111 00010000 00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010 00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100 00000000	200.23.20.0/23
...	….. ….	….
Organization 7	11001000 00010111 00011110 00000000	200.23.30.0/23

IP addressing: the last word...

How does an ISP get block of addresses?

ICANN: Internet Corporation for Assigned Names and Numbers (guidelines in RFC 2050)

- allocates addresses
- manages DNS
- assigns domain names, resolves disputes

Internet of Things

- The network of physical objects or "things" embedded with electronics, software, sensors
- Connected so as to offer service by exchanging data with the other connected devices
- Each thing is uniquely identifiable
- Things embedded with sensors and computing system
- Able to interoperate within the existing Internet infrastructure

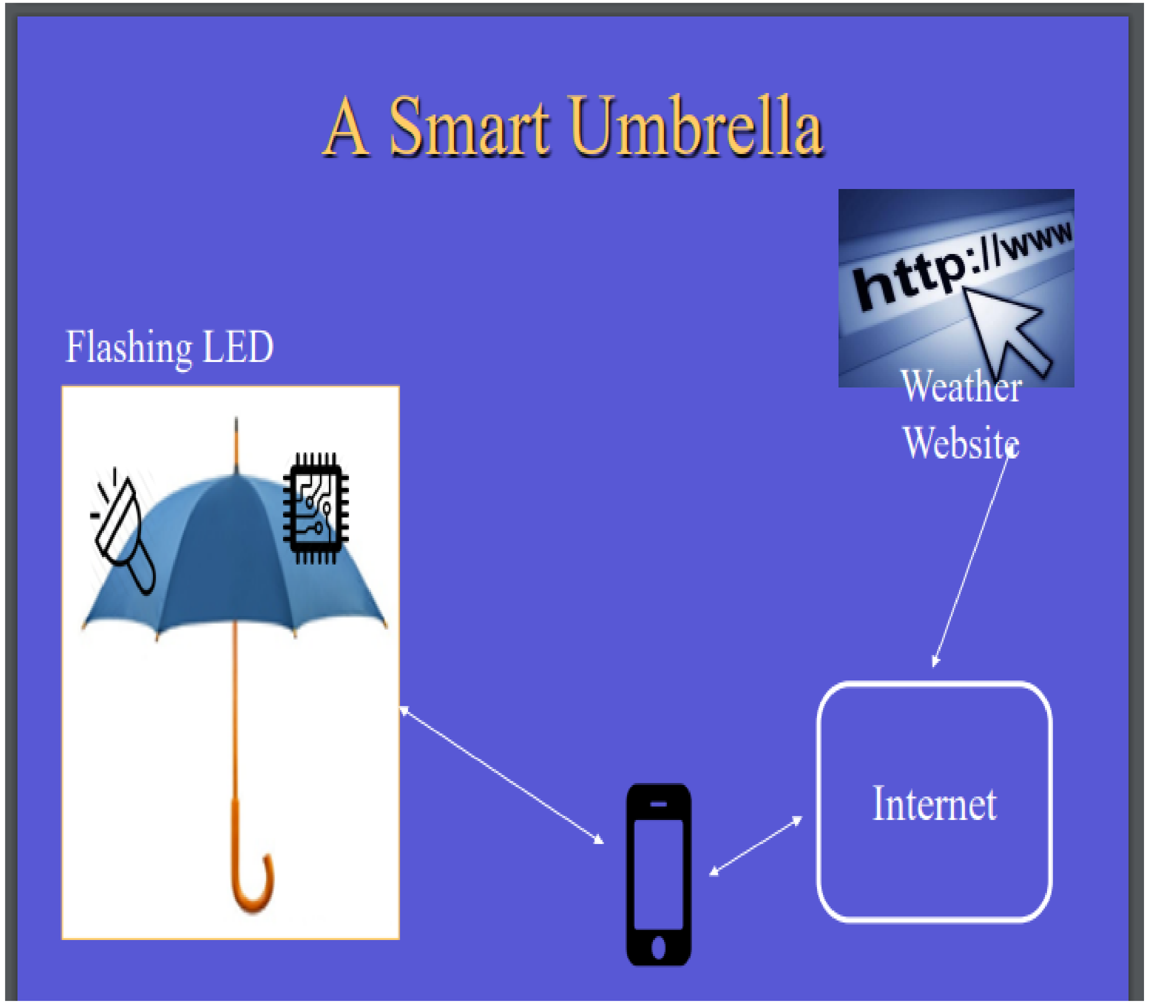
“A **dynamic** global network infrastructure with **self-configuring** capabilities based on standard and **interoperable communication protocols** where physical and virtual "things" have **identities**, physical attributes, and virtual personalities and use intelligent interfaces, and are **seamlessly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

Characteristics of IoT

- **Dynamic & Self-Adapting** : Adapt the changes w.r.t changing contexts
- **Self Configuring** : Ex. Fetching latest s/w updates without manual intervention.
- **Interoperable Communication Protocols** : Communicate through various protocols
- **Unique Identity** : Such as Unique IP Address or a URI
- **Integrated into Information Network** : This allows to communicate and exchange data with other devices to perform certain analysis.

“Internet of Things” Devices Concept

- Connected devices could be tracked, controlled or monitored using remote computers, Applications, Business Processes
- Smart Devices - Embedded devices with computing and communicating capabilities



Smart Hyper-connected Devices

- Hyperconnectivity: use of multiple systems and devices to remain constantly connected to networks, social networks and streams of information
- Smart devices constantly connect to networks
- For example:
  - a streetlights network constantly connected to a central controlling station/server

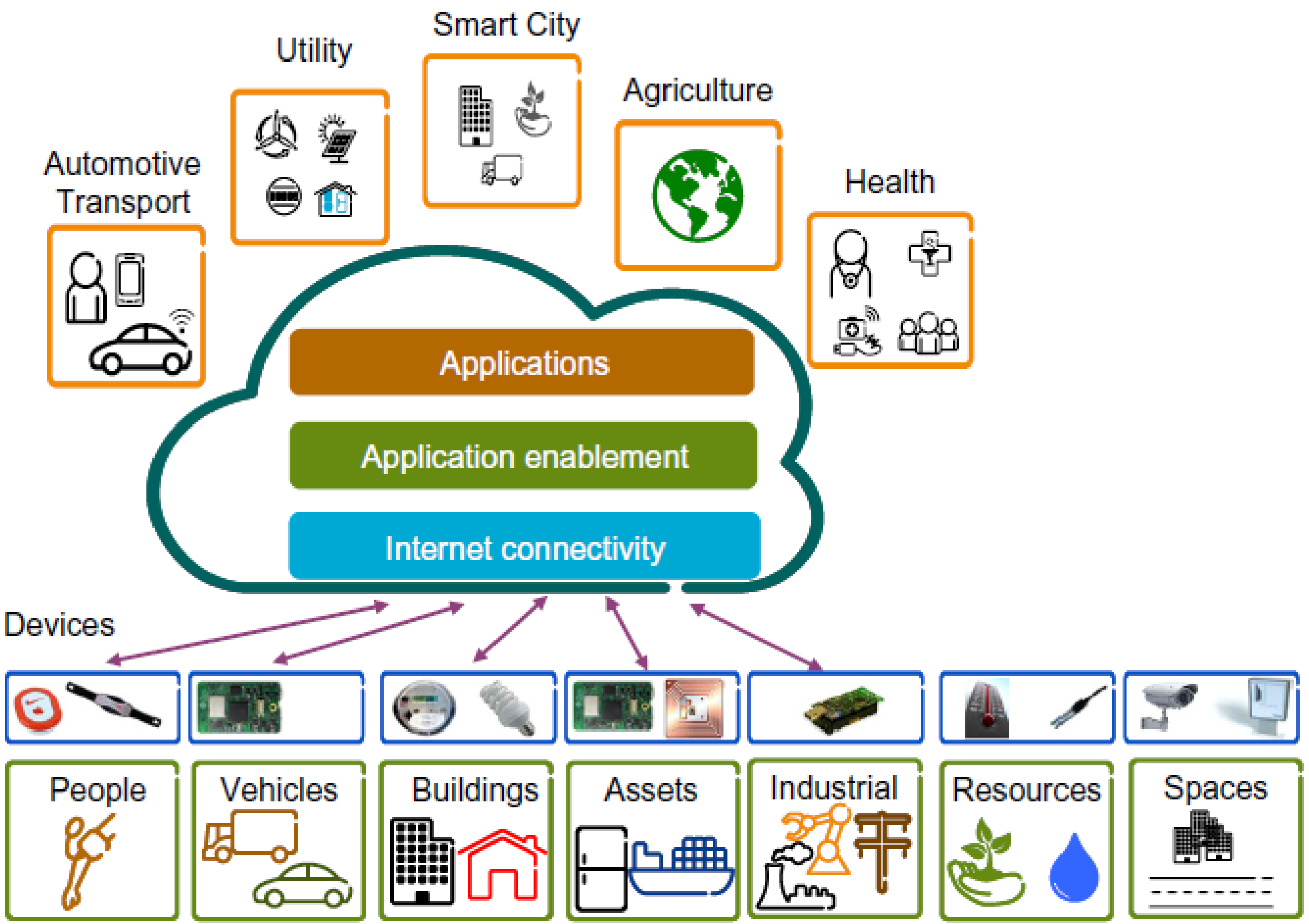
IoTs Vision

- A vision where things (wearable, watch, alarm clock, home devices, surrounding objects with) become smart and behave alive through sensing, computing and communicating systems
- A vision where embedded devices interact with remote objects or persons through connectivity, for examples, using Internet or Near Field Communication or other technologies.

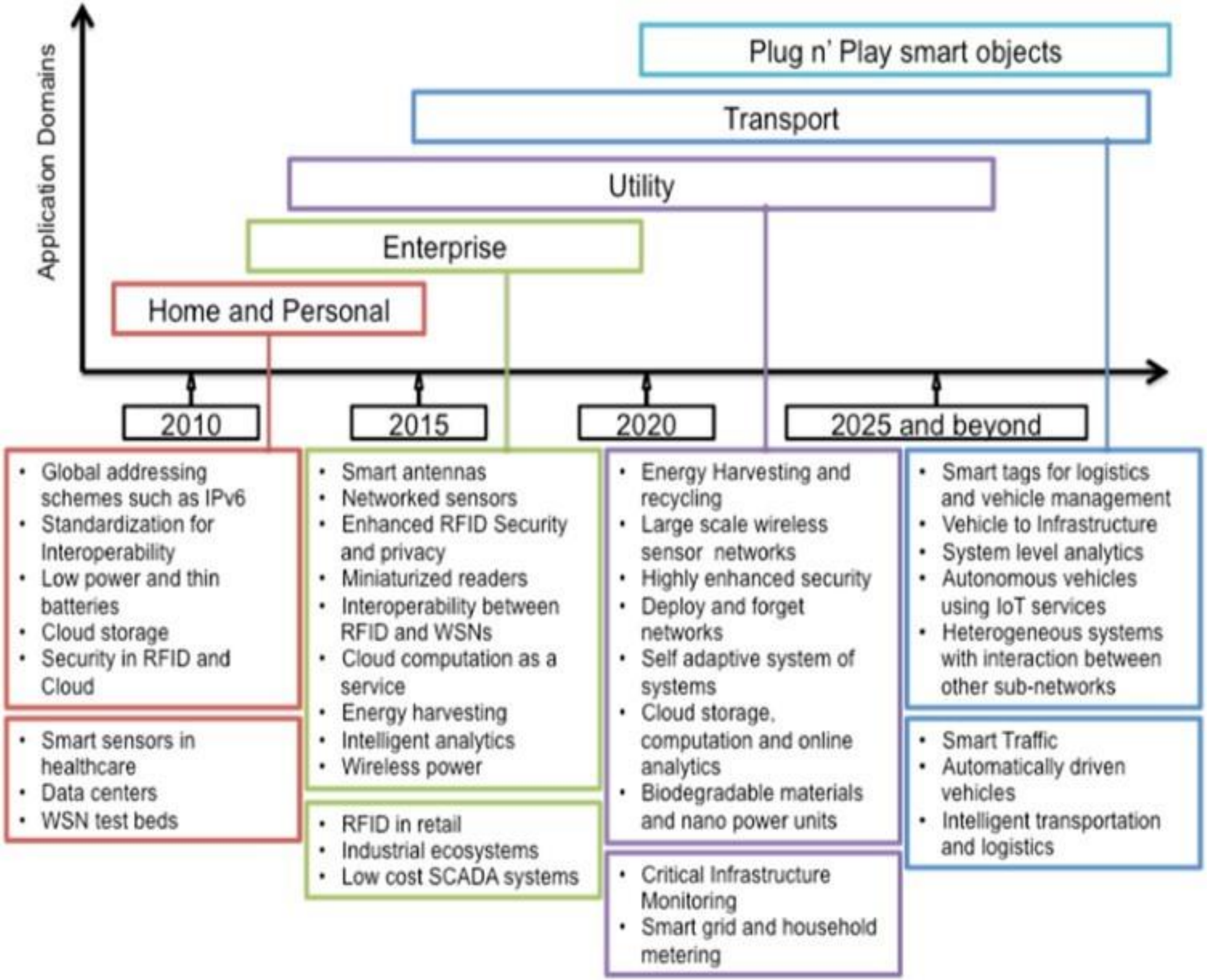
APPLICATIONS

Applications of IoT

- Smart Homes
- Smart City
- Self-driven Cars
- IoT Retail Shops
- Farming
- Wearables
- Smart Grids
- Industrial Internet
- Telehealth
- Smart Supply-chain Management
- Traffic management
- Water and Waste management



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.



- 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

- 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

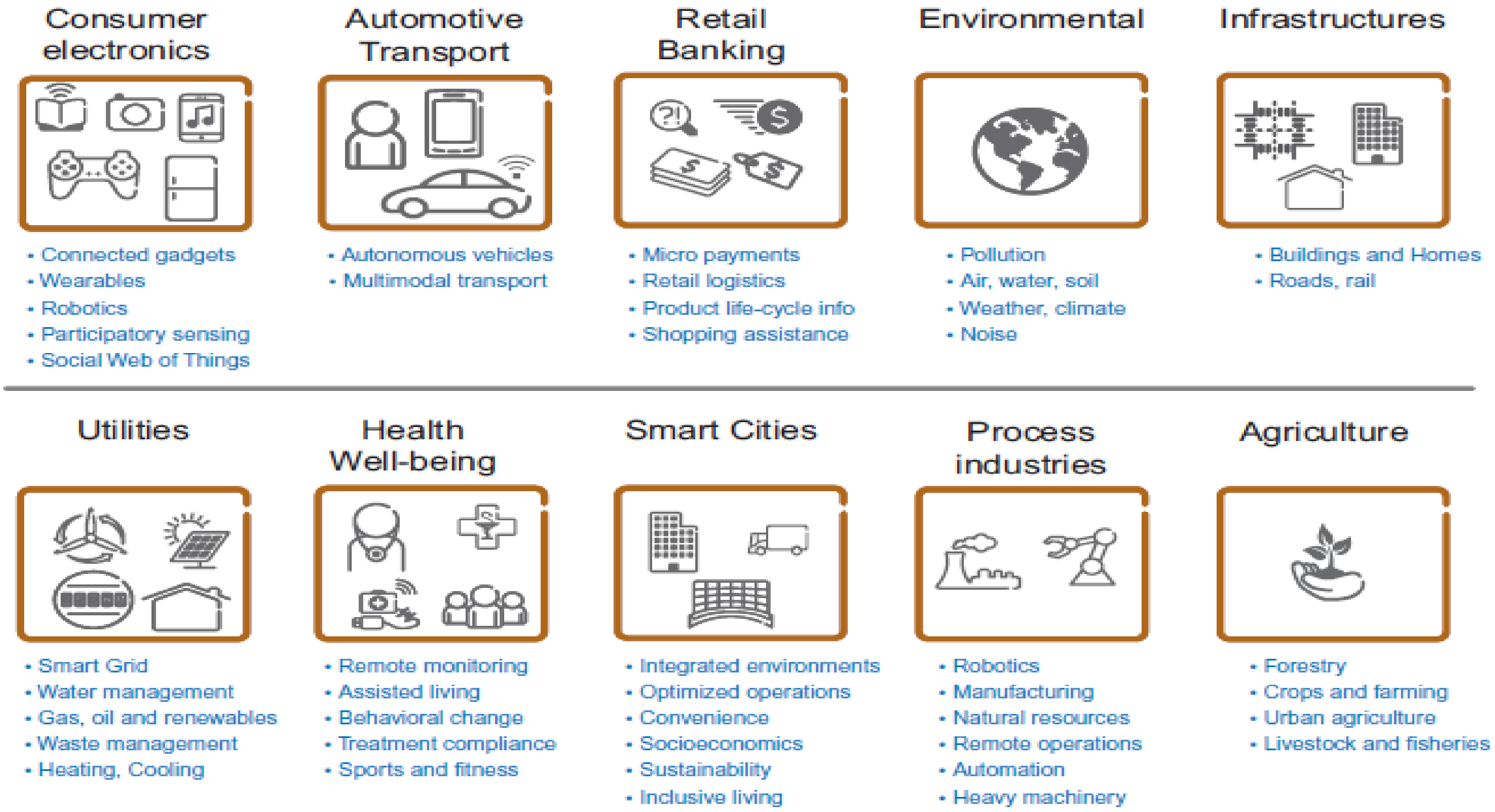
- Energy Harvesting and recycling
- Large scale wireless sensor networks
- Highly enhanced security
- Deploy and forget networks
- Self adaptive system of systems
- Cloud storage, computation and online analytics
- Biodegradable materials and nano power units

- Critical Infrastructure Monitoring
- Smart grid and household metering

- 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

Applications of IoT



Consumer electronics

- Connected gadgets
- Wearables
- Robotics
- Participatory sensing
- Social Web of Things

Automotive Transport

- Autonomous vehicles
- Multimodal transport

Retail Banking

- Micro payments
- Retail logistics
- Product life-cycle info
- Shopping assistance

Environmental

- Pollution
- Air, water, soil
- Weather, climate
- Noise

Infrastructures

- Buildings and Homes
- Roads, rail

Utilities

- Smart Grid
- Water management
- Gas, oil and renewables
- Waste management
- Heating, Cooling

Health Well-being

- Remote monitoring
- Assisted living
- Behavioral change
- Treatment compliance
- Sports and fitness

Smart Cities

- Integrated environments
- Optimized operations
- Convenience
- Socioeconomics
- Sustainability
- Inclusive living

Process industries

- Robotics
- Manufacturing
- Natural resources
- Remote operations
- Automation
- Heavy machinery

Agriculture

- Forestry
- Crops and farming
- Urban agriculture
- Livestock and fisheries

Smart Home

SMART HOME
17:30
23°
13°
Good Morning!

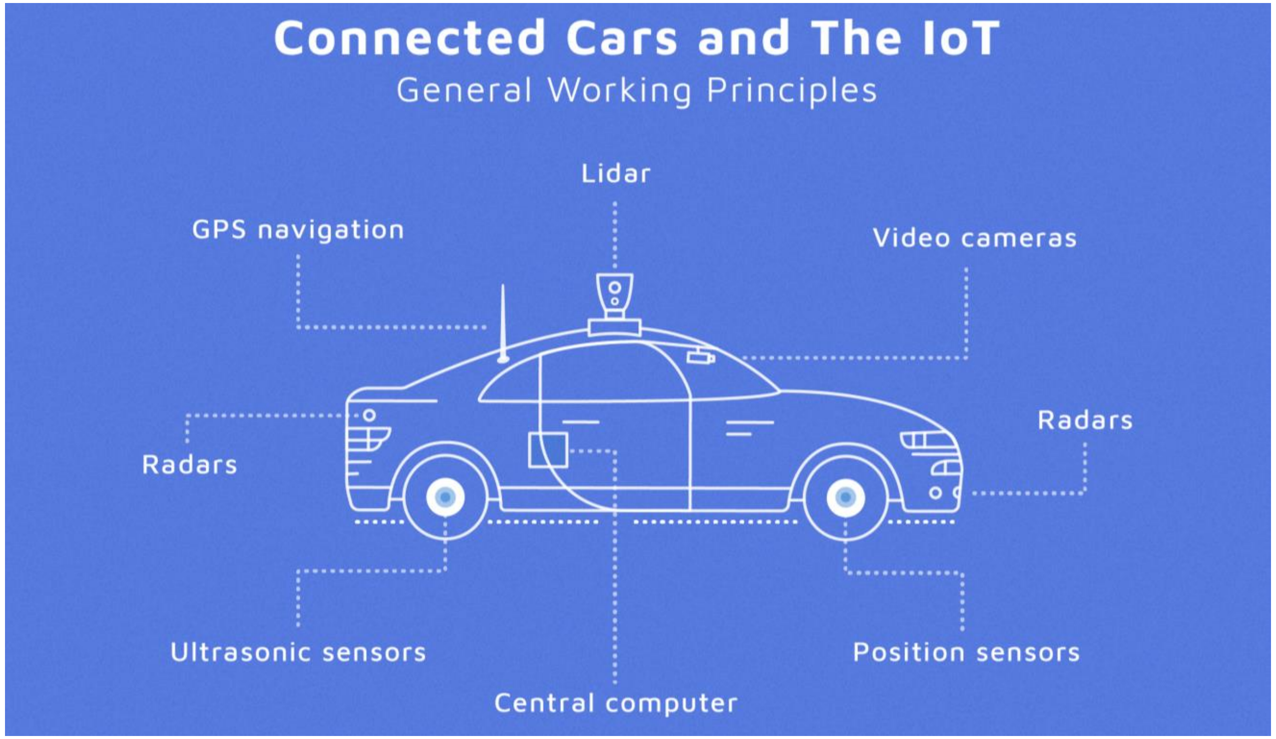
Home applications like lighting, air conditioners, locks, thermostat, etc. into a single system that can be controlled from your smartphone

Smart city

- To prevent environmental deterioration, avoid sanitation problems, mitigate traffic congestion, and thwart urban crime, municipalities turn to the Internet of Things (IoT).
- **Some Use Cases:**
  - Road traffic
  - Smart Parking
  - Public Transport
  - Street lighting
  - Waste management
  - Environment
  - Public safety

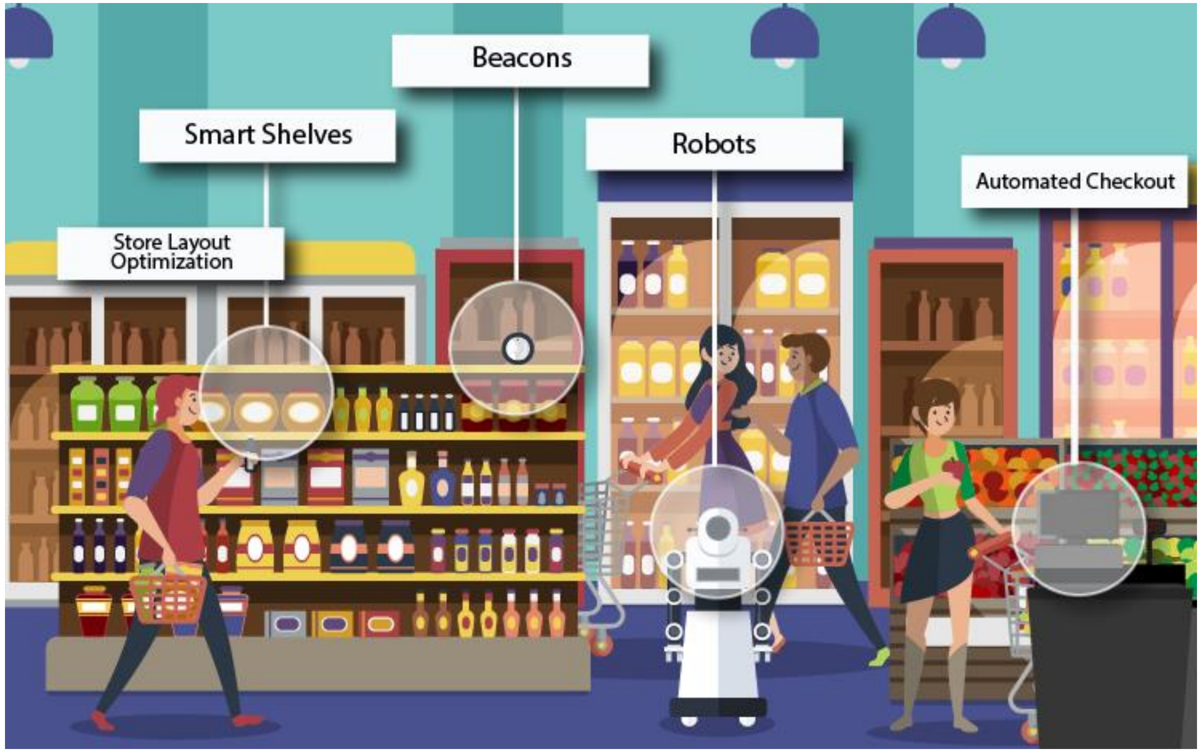
Self-driven Cars in IoT

Autonomous vehicles are thus connected to share information from the on-board sensors, as well as from smart phones of pedestrians and cyclists, traffic sensors, parking detectors, etc.

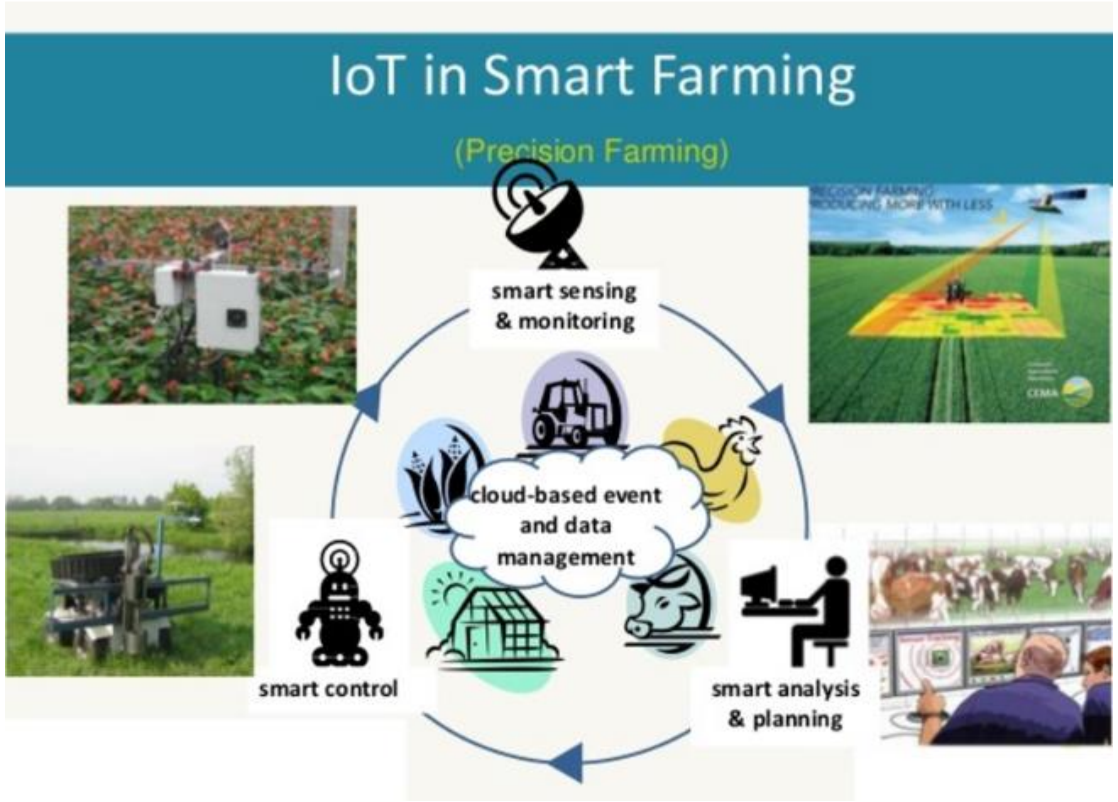


IoT Retail Shops

IoT allows us to monitor sales opportunities in real time and track missed in-store sales, taking demand planning to the next level

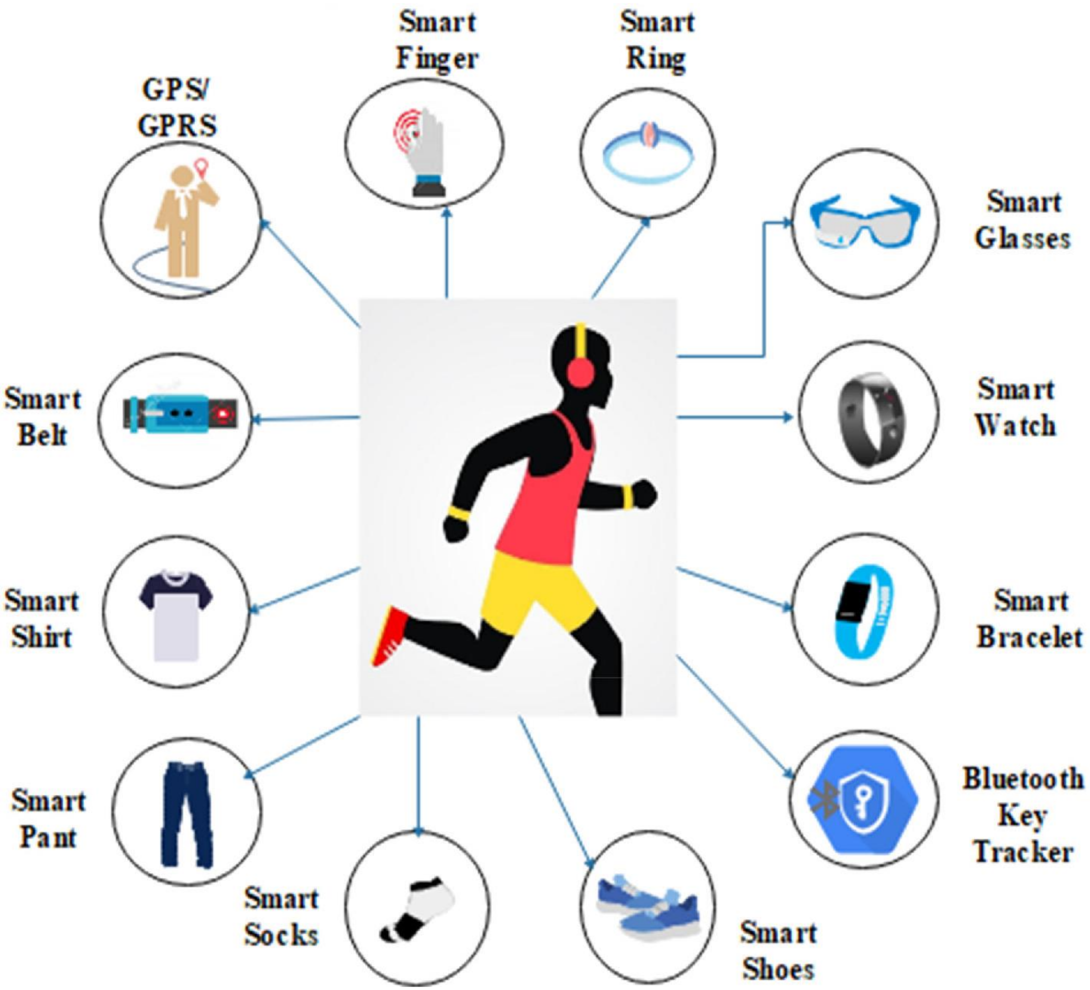


Farming



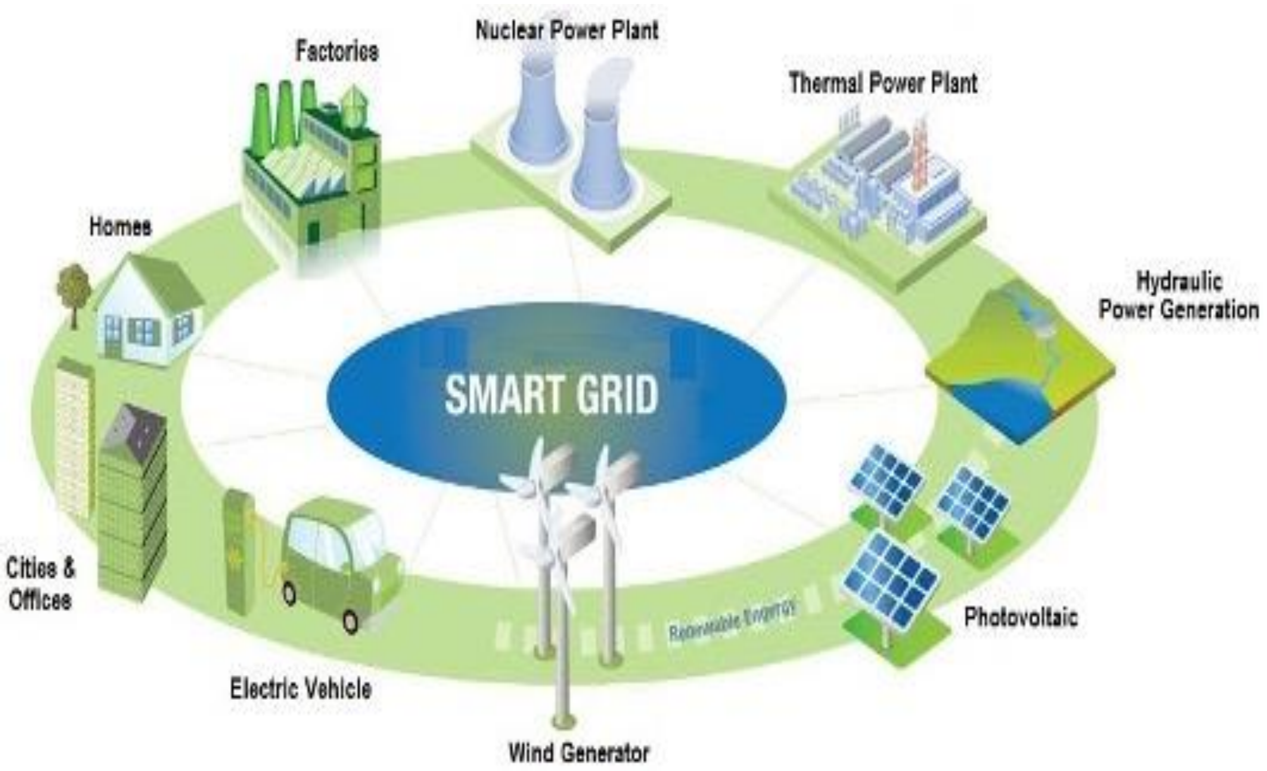
A system that is built for monitoring the crop field with the help of sensors (light, humidity, temperature, soil moisture, crop health, etc.) and automating the irrigation system

Smart Wearable



These wearable devices are used for tracking information on real time basis

Smart Grid

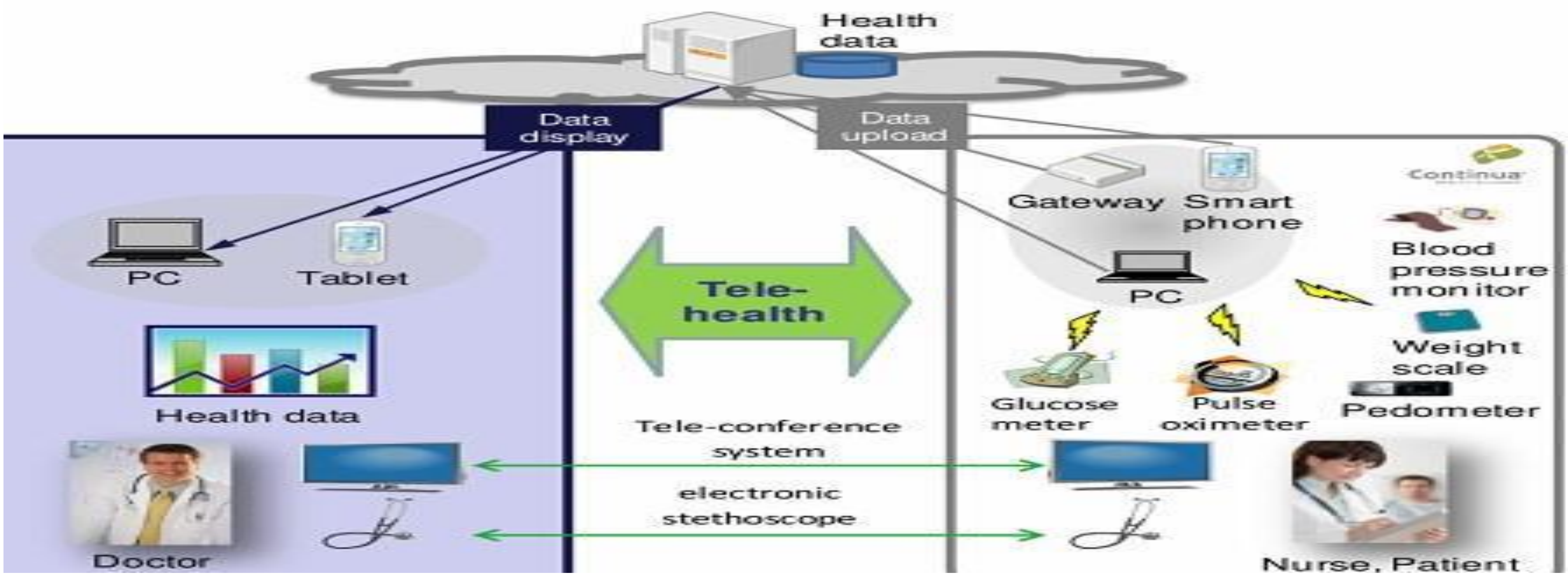


The Smart Grid is a unique chance to transform the energy sector into a new age of dependability, availability, and efficiency, which will contribute to our economic and environmental health.

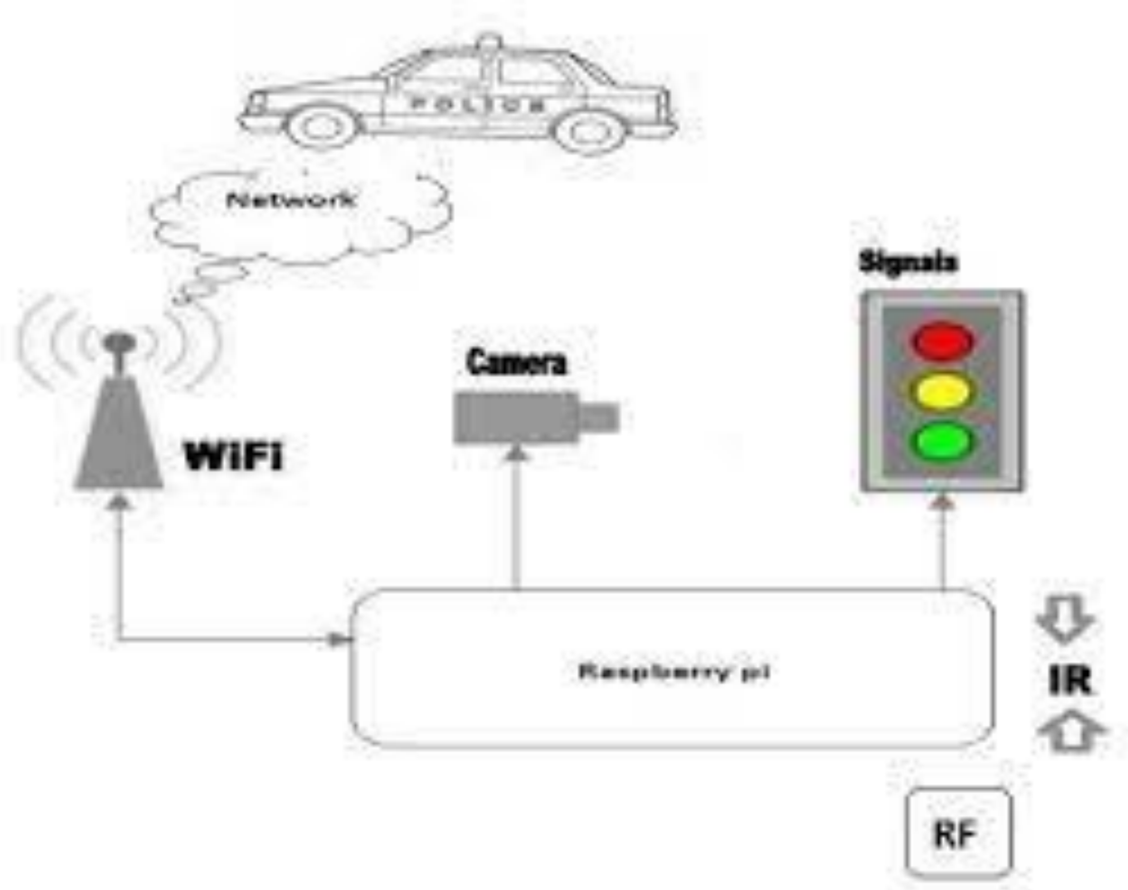
Industrial Internet

Industrial IoT is part of this larger concept known as the Internet of Things (IoT).

Telehealth

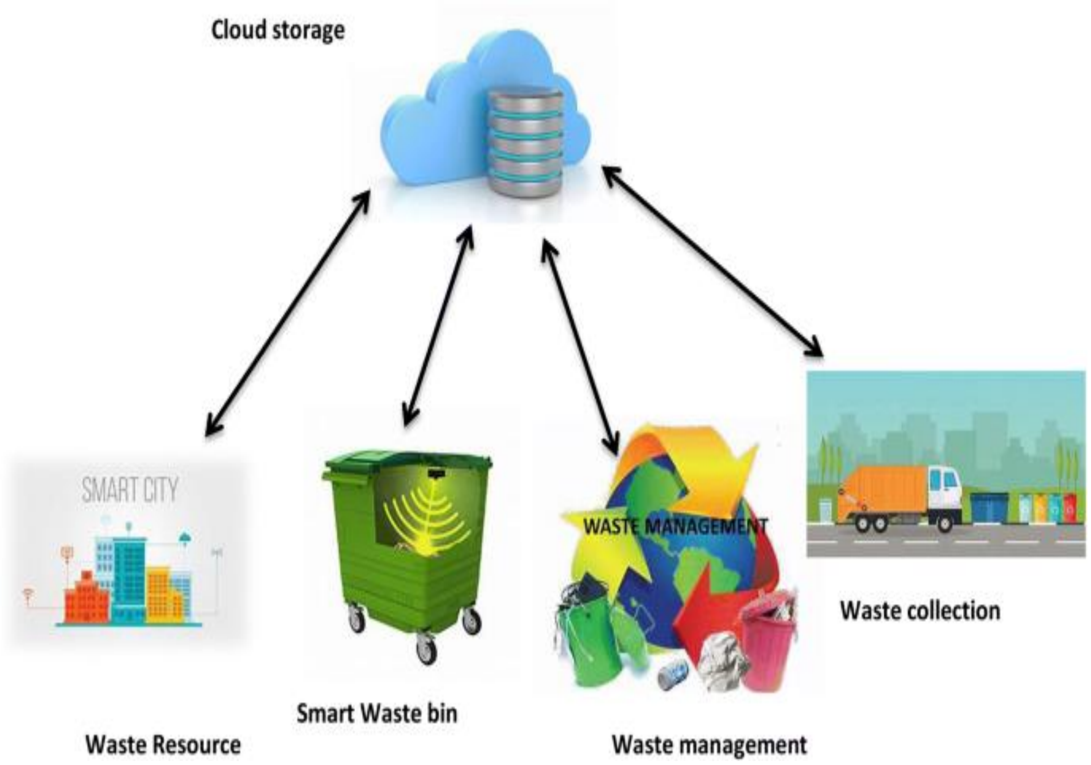

Health data
Data display
Data upload
PC
Tablet
Health data
Doctor
Tele-health
Tele-conference system
electronic stethoscope
Gateway
Smart phone
Continua
Blood pressure monitor
PC
Weight scale
Glucose meter
Pulse oximeter
Pedometer
Nurse, Patient

Smart Traffic



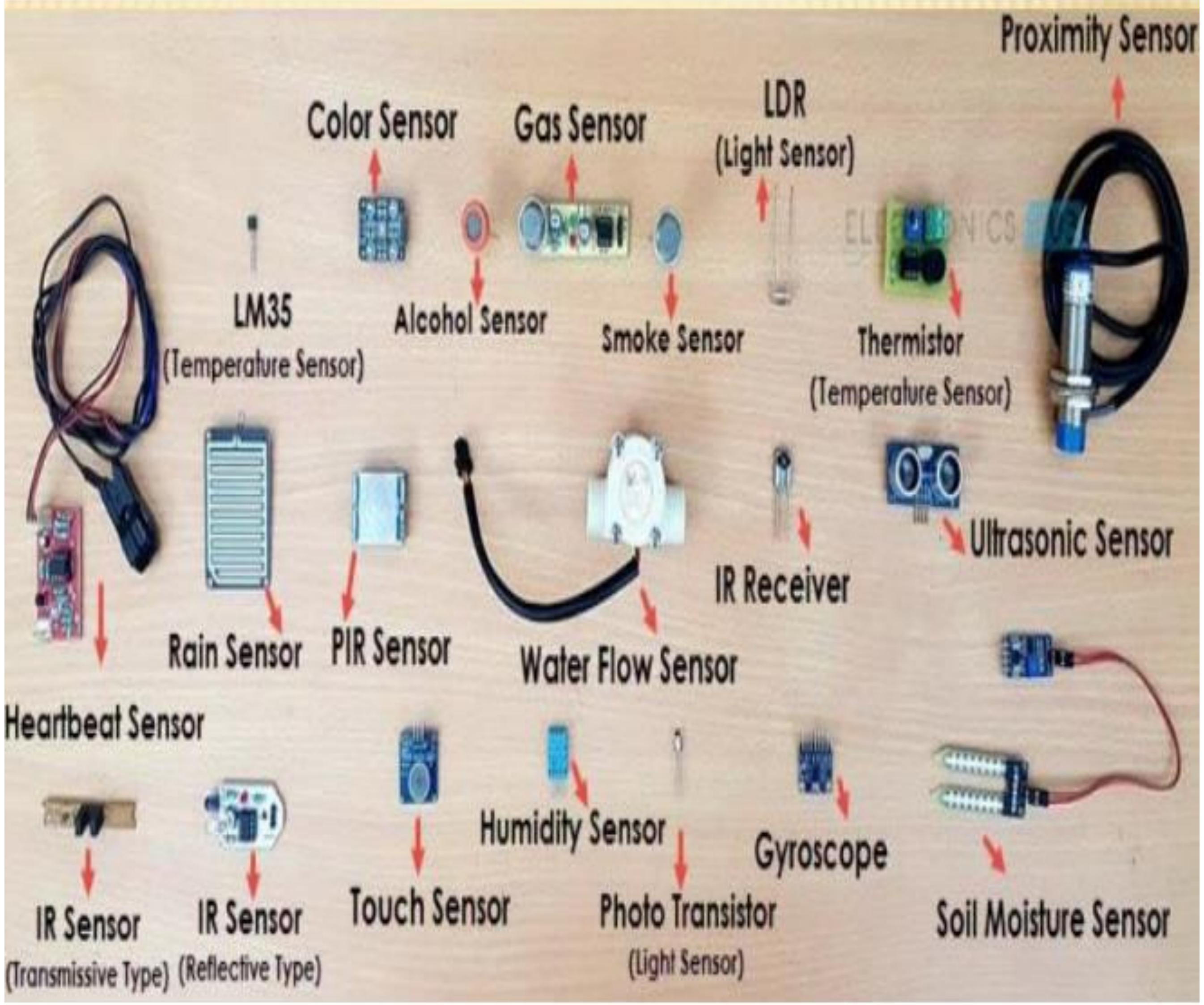
Smart Traffic Management is a system where centrally-controlled traffic signals and sensors regulate the flow of traffic through the city in response to demand.

Waste Management

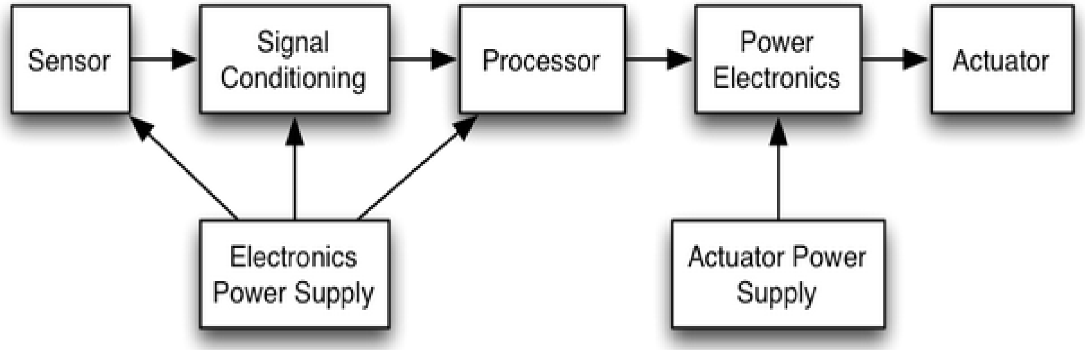


Sensors and Actuators

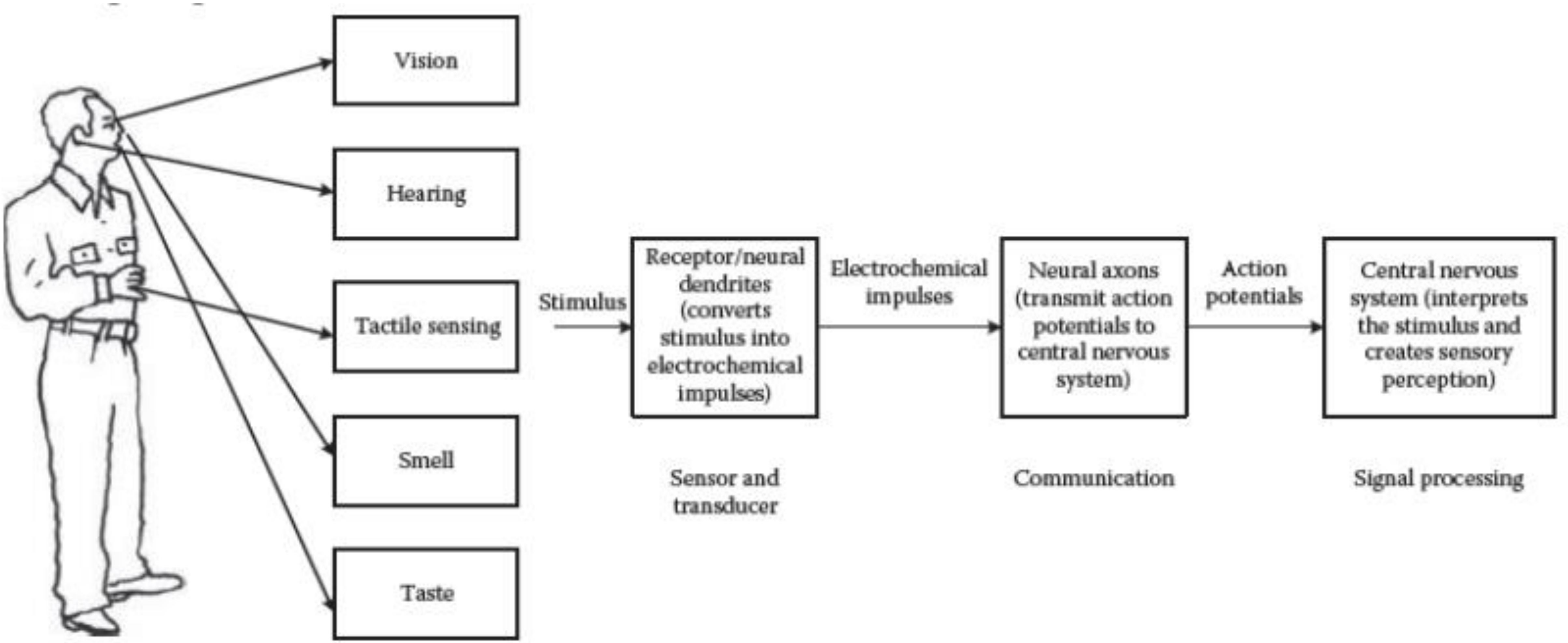
Sensors


Proximity Sensor
Color Sensor
Gas Sensor
LDR
(Light Sensor)
LM35
(Temperature Sensor)
Alcohol Sensor
Smoke Sensor
Thermistor
(Temperature Sensor)
Rain Sensor
PIR Sensor
Water Flow Sensor
IR Receiver
Ultrasonic Sensor
Heartbeat Sensor
Humidity Sensor
Gyroscope
IR Sensor
(Transmissive Type)
IR Sensor
(Reflective Type)
Touch Sensor
Photo Transistor
(Light Sensor)
Soil Moisture Sensor

Detects
Light
Detects
Sound
Detects
Certain Chemicals
Detects
Pressure & Temperature


Sensor
Signal
Conditioning
Processor
Power
Electronics
Actuator
Electronics
Power Supply
Actuator Power
Supply

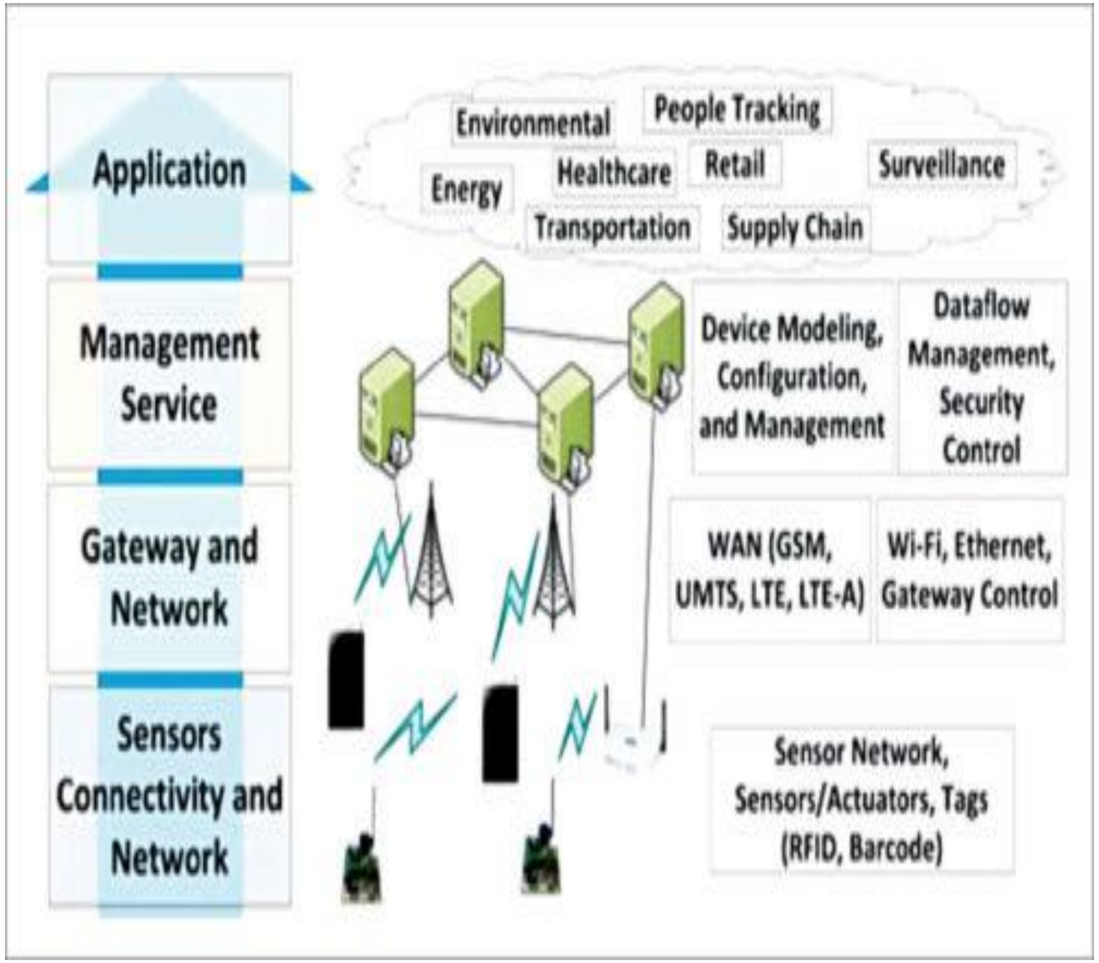
Sensing Process


Vision
Hearing
Tactile sensing
Smell
Taste
Stimulus
Receptor/neural dendrites (converts stimulus into electrochemical impulses)
Electrochemical impulses
Neural axons (transmit action potentials to central nervous system)
Action potentials
Central nervous system (interprets the stimulus and creates sensory perception)
Sensor and transducer
Communication
Signal processing

- Sensors have been around for a long time.
- The first thermostat was introduced in the late 1880s and infrared sensors have been around since the late 1940s.
- Sensors are devices that detect and respond to changes in an environment.
- Inputs can come from a variety of sources such as light, temperature, motion and pressure.
- Sensors output valuable information and if they are connected to a network, they can share data with other connected devices and management systems.

Importance of Sensor in IoT

- The sensor and network connectivity in the IoT mainly located in the bottom layer.
- The main purpose of sensors is to collect data from the surrounding environment. Sensors, or 'things' of the IoT system, form the front end.
- These are connected directly or indirectly to IoT networks after signal conversion and processing. But all sensors are not the same and different IoT applications require different types of sensors.


Application
Management Service
Gateway and Network
Sensors Connectivity and Network
Environmental
People Tracking
Energy
Healthcare
Retail
Surveillance
Transportation
Supply Chain
Device Modeling, Configuration, and Management
Dataflow Management, Security Control
WAN (GSM, UMTS, LTE, LTE-A)
Wi-Fi, Ethernet, Gateway Control
Sensor Network, Sensors/Actuators, Tags (RFID, Barcode)

- Generally speaking, a sensor is a device that is able to detect changes in an environment. By itself, a sensor is useless, but when we use it in an electronic system, it plays a key role.
- A sensor is able to measure a physical phenomenon (like temperature, pressure, and so on) and transform it into an electric signal.

These three features should be at the base of a good sensor:

  - It should be sensitive to the phenomenon that it measures
  - It should not be sensitive to other physical phenomena
  - It should not modify the measured phenomenon during the measurement process

Properties

A sensor can be described using several properties, the most important being:

- Range: The maximum and minimum values of the phenomenon that the sensor can measure.
- Sensitivity: The minimum change of the measured parameter that causes a detectable change in output signal.
- Resolution: The minimum change in the phenomenon that the sensor can detect.

Ultrasonic Sensor

- Ultrasonic sensors are devices that use ultrasonic waves for various applications, including distance measurement, object detection, and proximity sensing.

Working Principle:

  - Ultrasonic sensors have a transducer that generates ultrasonic waves (sound waves with a frequency above the audible range of human hearing, typically above 20 kHz).
  - The sensor emits a burst of ultrasonic waves in a specific direction. These waves travel through the air until they encounter an object.
  - When the ultrasonic waves hit an object, they reflect off the object's surface. The sensor's transducer then detects the reflected waves.
  - The sensor measures the time it takes for the ultrasonic waves to travel to the object and back. This time of flight is used to calculate the distance between the sensor and the object.
  - The distance (D) can be calculated using the formula: D = (Speed of Sound × Time of Flight) / 2.

- Applications: Distance Measurement, Object Detection, Liquid Level Measurement, Intruder Detection

Smoke Sensor

- Smoke sensors, also known as smoke detectors or smoke alarms, are devices designed to detect the presence of smoke in the air.
- Working Principles:
  - Smoke sensors operate based on the detection of smoke particles in the air. When a fire produces smoke, small particles are released into the surrounding environment.
  - Photoelectric sensors use a light source and a photosensitive receiver. When smoke enters the chamber, it scatters light, causing a reduction in the light reaching the receiver and triggering the alarm
- Applications: Residential Use, Industrial Facilities, Aircraft and Vehicles, Fire Alarm Systems

Motion Sensor

- A PIR (Passive Infrared) sensor is a type of motion sensor that detects changes in infrared radiation in its field of view.
- All objects with a temperature above absolute zero emit infrared radiation. PIR sensors detect changes in the infrared radiation patterns caused by the movement of warm objects, such as humans or animals, within their detection range.
- PIR sensors often include adjustments for time delay and sensitivity. Time delay settings control how long the sensor remains activated after detecting motion, and sensitivity settings determine the range and size of the objects that trigger the sensor.
- Applications: Lightning Control, Security, Automatic Door opener

Humidity Sensors

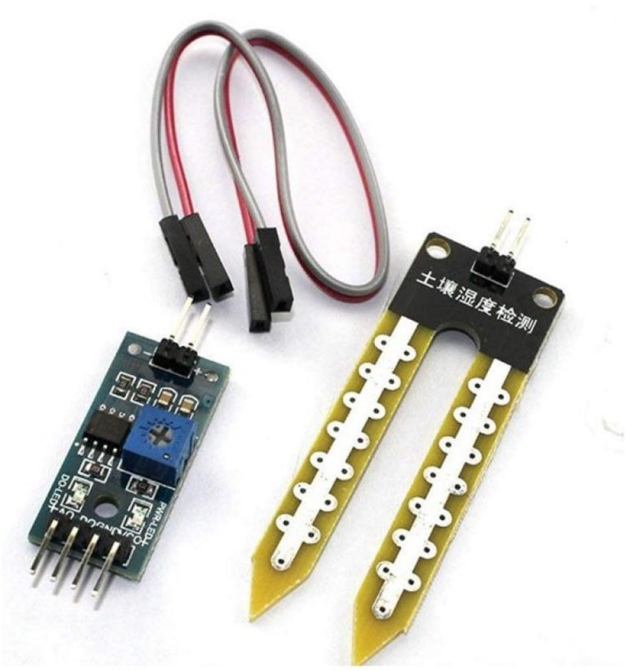
- A humidity sensor, also known as a hygrometer or humidity transducer, is a device designed to measure the moisture content or relative humidity in the air.
- Working Principle:
  - Capacitive humidity sensors are based on the principle that the capacitance of a material changes with variations in humidity. These sensors typically consist of a hygroscopic material that absorbs or releases water vapor, causing changes in capacitance.
- Applications: Weather Stations, Industrial Processes, Medical Devices

Gas sensors

- Gas sensors are mainly used for detecting toxic gases.
- Working Principle:
  - Gas sensors operate based on chemical detection mechanisms. They contain a sensing element or material that reacts with the target gas, leading to a measurable change in electrical properties or other physical characteristics.
  - The sensing element is connected to a transducer, which converts the detected change into an electrical signal.
  - The output signal from the transducer is then processed and measured by the sensor's electronics. The signal strength correlates with the concentration of the target gas.
- Applications: Industrial Safety, Environmental Monitoring, Medical Applications

Soil Moisture Sensor

- A soil moisture sensor is a device designed to measure the moisture content in the soil.
- Working Principle:
  - Capacitive soil moisture sensors operate based on the capacitance changes in the soil. The sensor has electrodes that measure the dielectric constant of the soil, which varies with moisture content. Higher moisture levels increase the capacitance.
- Applications: Agriculture, Gardening, Environmental Monitoring, Construction and Civil Engineering

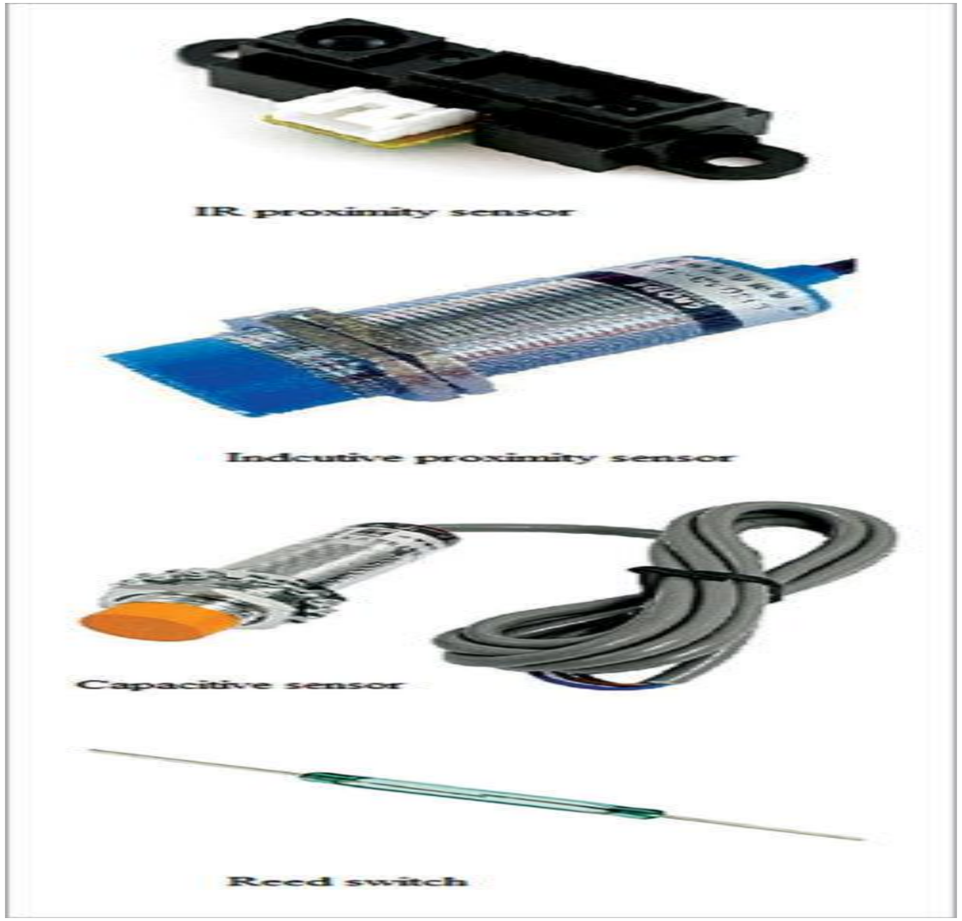


LDR Sensor

- An LDR (Light Dependent Resistor), also known as a photoresistor, is a type of resistor whose resistance varies with the amount of light falling on it.
- Working Principle:
  - LDRs are typically made of semiconductor materials.
  - The resistance of an LDR decreases with an increase in the intensity of light falling on it. This is known as photoconductivity. When exposed to light, photons are absorbed by the semiconductor material, generating electron-hole pairs and reducing the resistance.
  - In the absence of light, an LDR has a high resistance, referred to as dark resistance. As the intensity of light increases, the resistance decreases.
- Application: Light Sensing, Brightness Control, Automotive Lighting

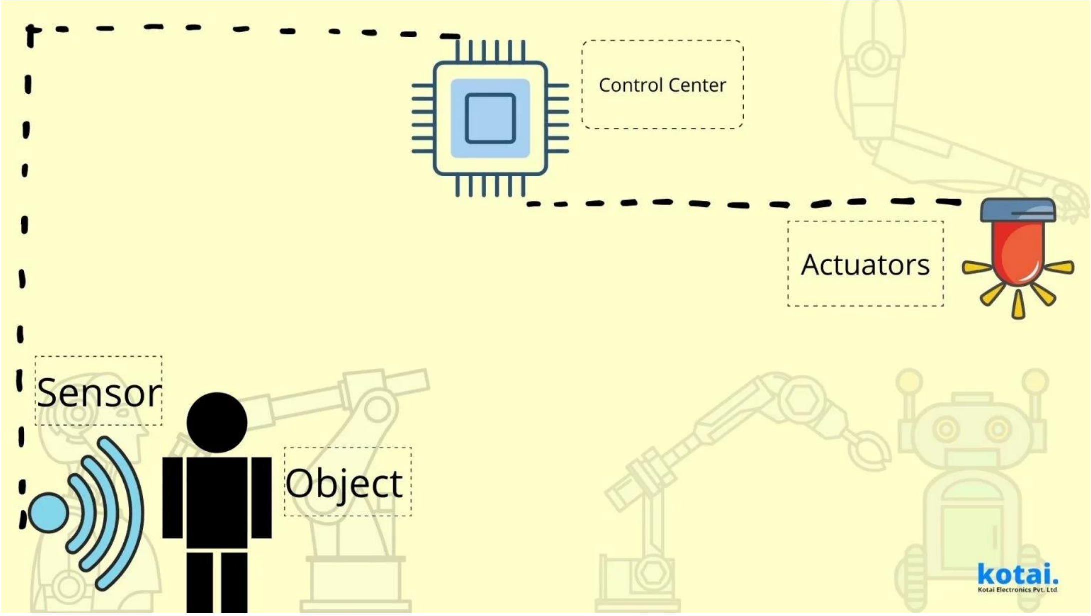
Proximity sensors

- These sensors detect the presence or absence of a nearby object without any physical contact.
- Different types of proximity sensors are inductive, capacitive, photoelectric, ultrasonic and magnetic.
- These are mostly used in object counters, process monitoring and control.



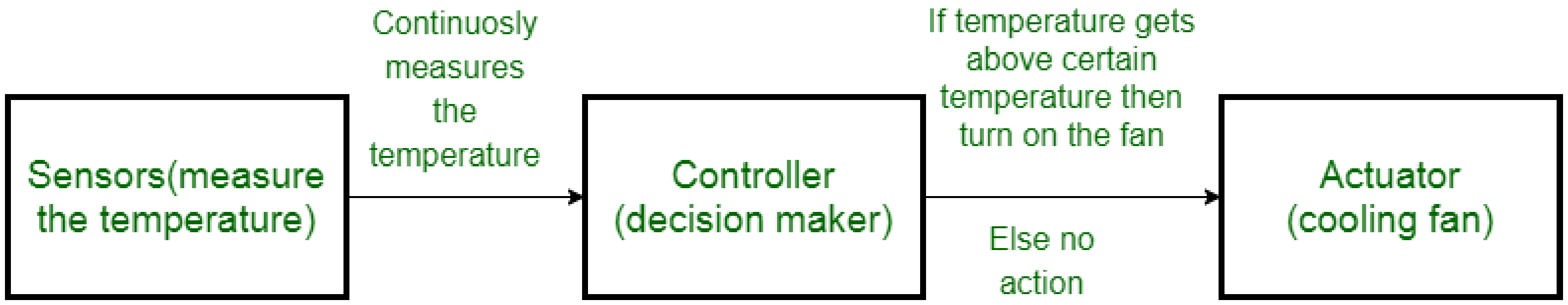
Actuators

- It takes the electrical signal and converts it into certain physical actions
- difference between sensors and actuators in IoT is sensor track the output from the environment whereas the actuator track the output from the control center.



What Connects Sensors and Actuators in IoT Device

- In a smart IoT system, The sensor collects data and sends it to the control center.
- The control center processes the data depending on what they are programmed to do and then it commands the actuators to perform certain tasks.
- Basically, if the sensor is the brain and the actuator is the limb that performs the tasks it’s the main difference between sensors and actuators in IoT.


Sensors(measure the temperature)

Continuosly measures the temperature

Controller (decision maker)

If temperature gets above certain temperature then turn on the fan

Else no action

Actuator (cooling fan)

Types of Actuators

Hydraulic Actuators

- This Actuator operates by converting hydraulic power to do the mechanical tasks. Here the mechanical power is converted into rotary, leaner, and oscillatory motion. The actuator here captures output from motors and uses liquid as the pressure generator.

Pneumatic Actuators

- This actuator works similar to the hydraulic actuator but it uses vacuum or compressed air to convert it into mechanical power they are weaker compared to hydraulic actuators.

Electrical Actuators

- Similarly, electrical actuators use electrical energy to turn it into mechanical torque. The mechanical actuators are used in industrial places

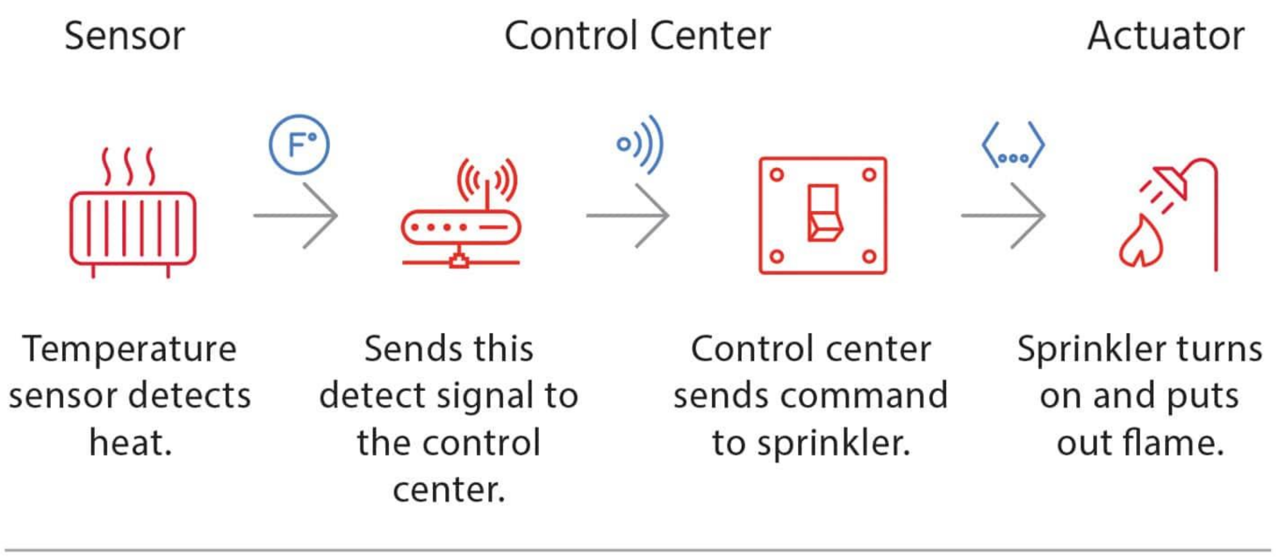
Thermal Actuators

- This can be called an electric less motor. It’s equipped with thermal-sensitive material that’s capable of producing linear motion in response to temperature changes. Opposite to another actuator, it does not need an external power source. It is used to release latches, operate switches, and open or close valves.

Magnetic Actuators

- This kind of actuator changes electronic magnetic current to mechanical output They operate in either a rotary or linear direction and can have continuous or limited motion. Magnetic actuators are used within the aerospace, automotive industry, healthcare, computers, and many other industries

Actuators



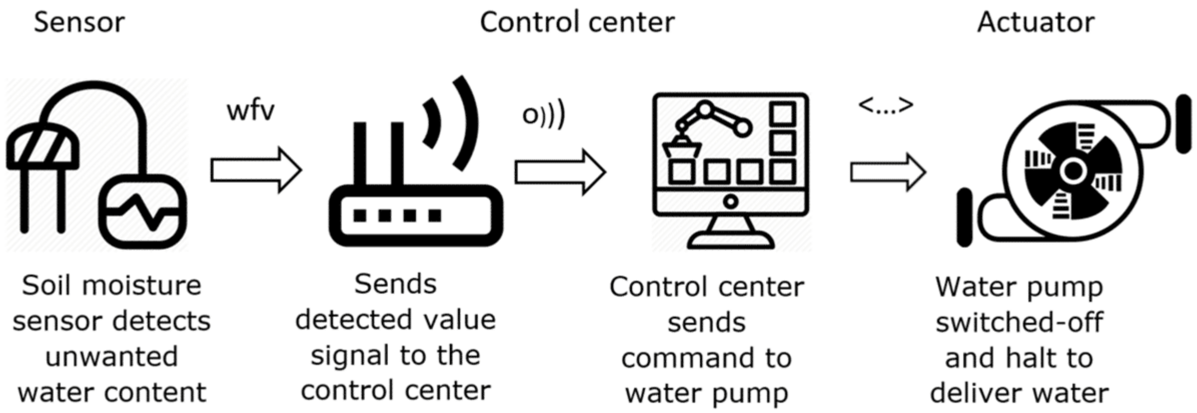
Temperature sensor detects heat.

Sends this detect signal to the control center.

Control center sends command to sprinkler.

Sprinkler turns on and puts out flame.

Sensor to **Actuator** Flow



Soil moisture sensor detects unwanted water content

Sends detected value signal to the control center

Control center sends command to water pump

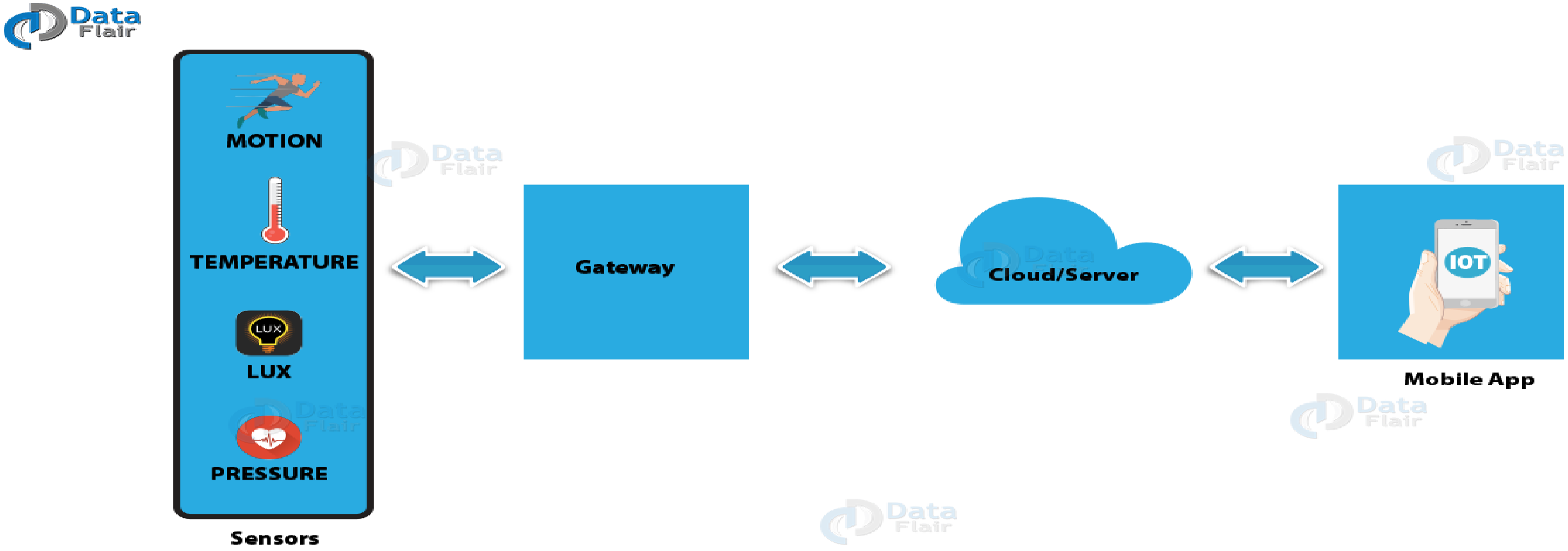
Water pump switched-off and halt to deliver water

Physical Design of IoT

- Things in IoT
- IoT Protocols

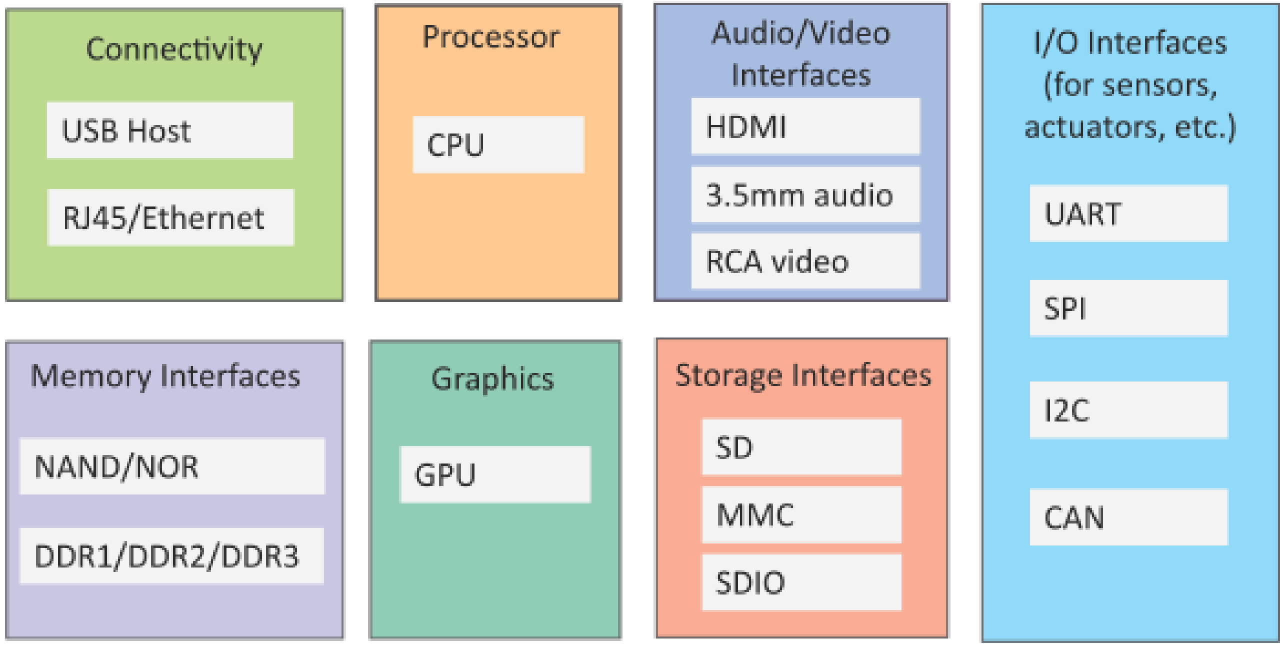
Components of IOT

- IoT is an ecosystem of connected physical objects that are accessible through the Internet



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



Connectivity
USB Host
RJ45/Ethernet

Processor
CPU

Audio/Video Interfaces
HDMI
3.5mm audio
RCA video

I/O Interfaces (for sensors, actuators, etc.)
UART
SPI
I2C
CAN

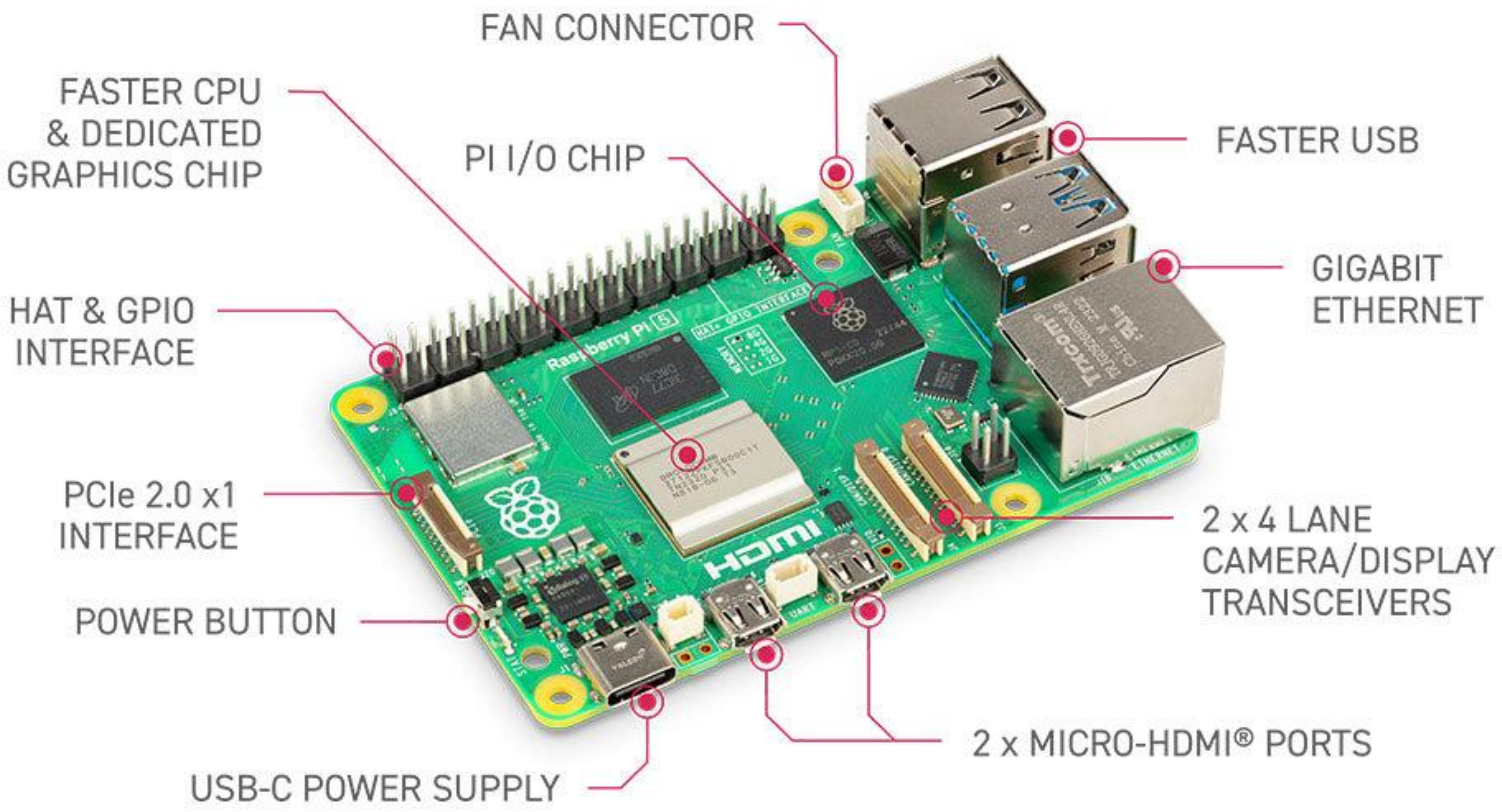
Memory Interfaces
NAND/NOR
DDR1/DDR2/DDR3

Graphics
GPU

Storage Interfaces
SD
MMC
SDIO

Generic block diagram of a IoT Device

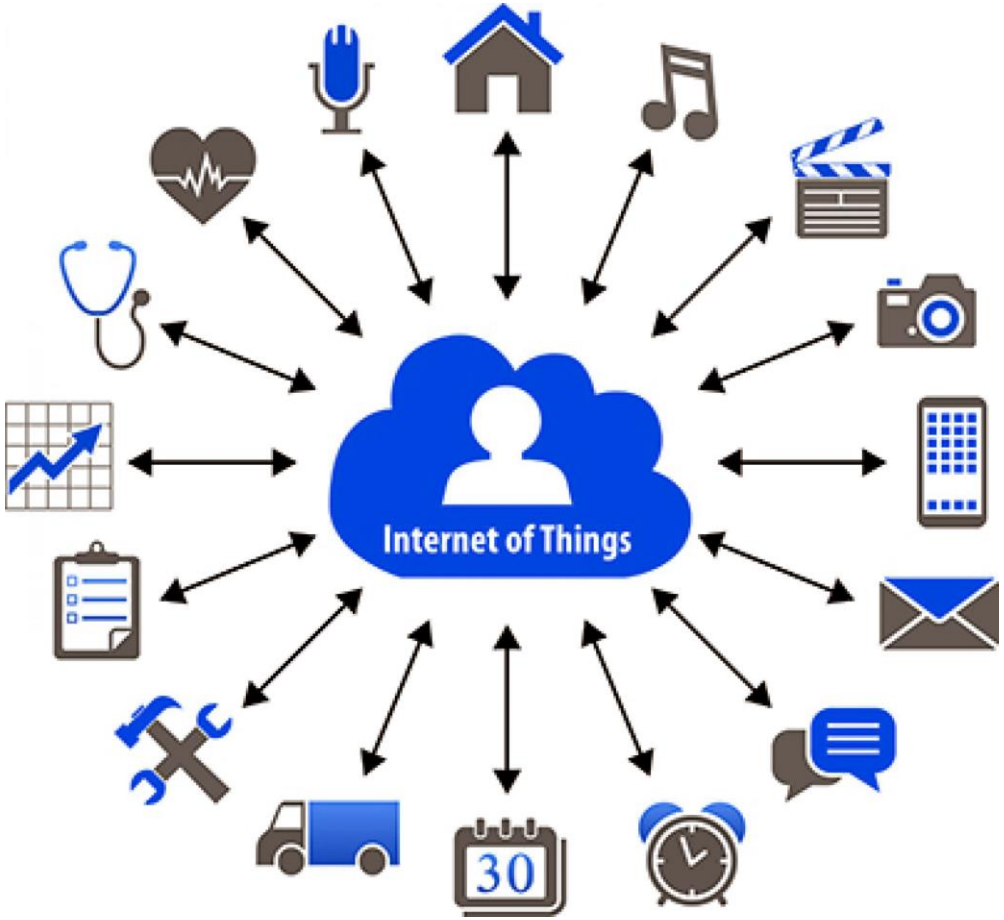
- Raspberry Pi 5



Components of IoT

- **Microcontroller** : 8-, 16-, or 32-bit working memory and storage
- **Power Source** : Fixed, battery, energy harvesting, or hybrid.
- **Sensors and Actuators** : Onboard sensors and actuators
- **Communication** : Cellular, wireless, or wired for LAN and WAN communication.
- **Operating System (OS)** : Main-loop, event-based, real-time, or full featured OS.
- **Applications** : Simple sensor sampling or more advanced applications.
- **User Interface** : Display, buttons, or other functions for user interaction.
- **Device Management (DM):** Provisioning, firmware, bootstrapping, and monitoring.
- **Execution Environment (EE):** Application lifecycle management and Application Programming Interface (API).

Sensors



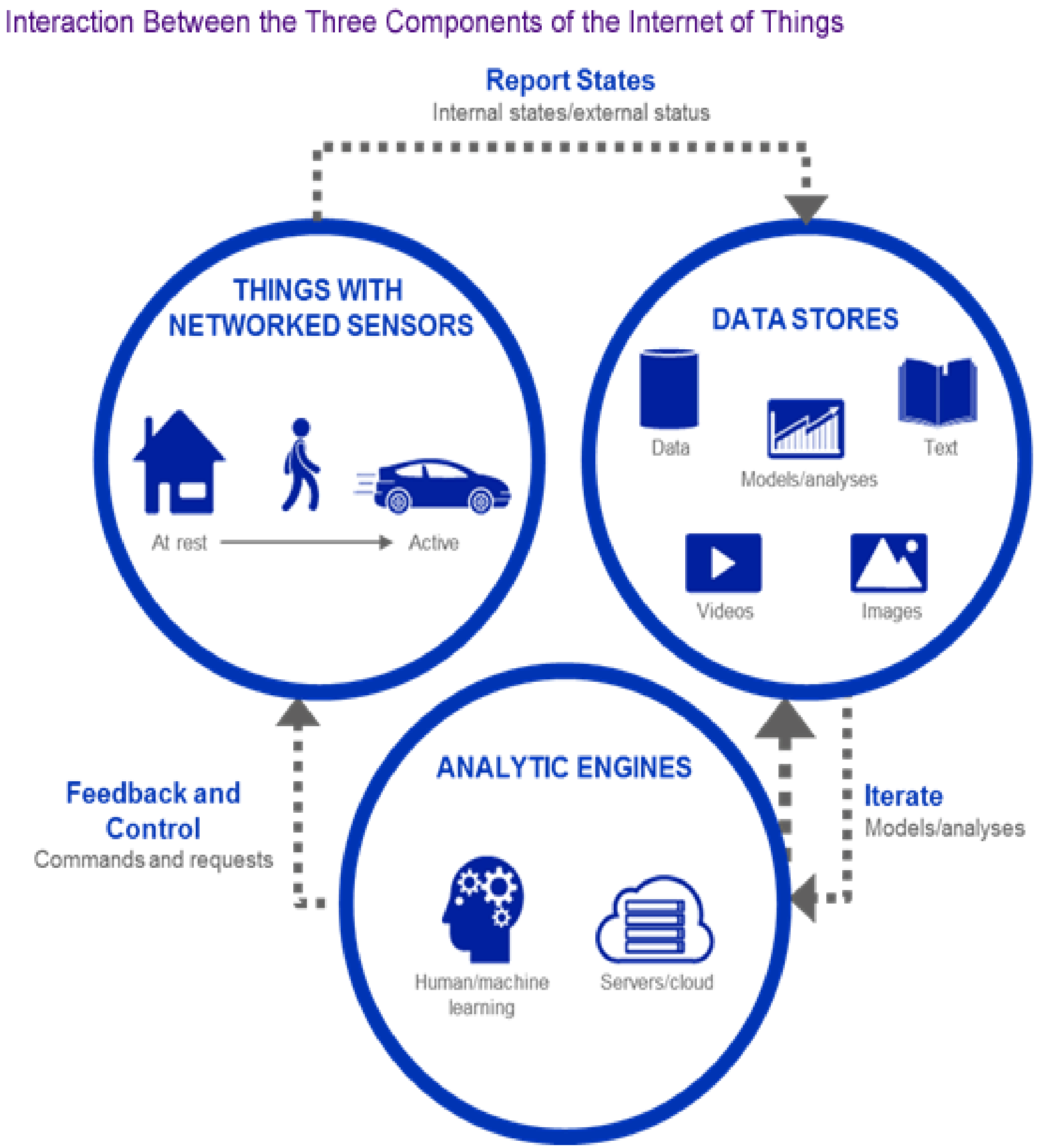
- A device can have multiple sensors that can bundle together to do more than just sense things.
- For example, our phone is a device that has multiple sensors such as GPS, accelerometer, camera but our phone does not simply sense things.

Connectivity

- The sensors can be connected to the cloud through various mediums of communication and transports such as cellular networks, satellite networks, Wi-Fi, Bluetooth, wide-area networks (WAN), low power wide area network and many more

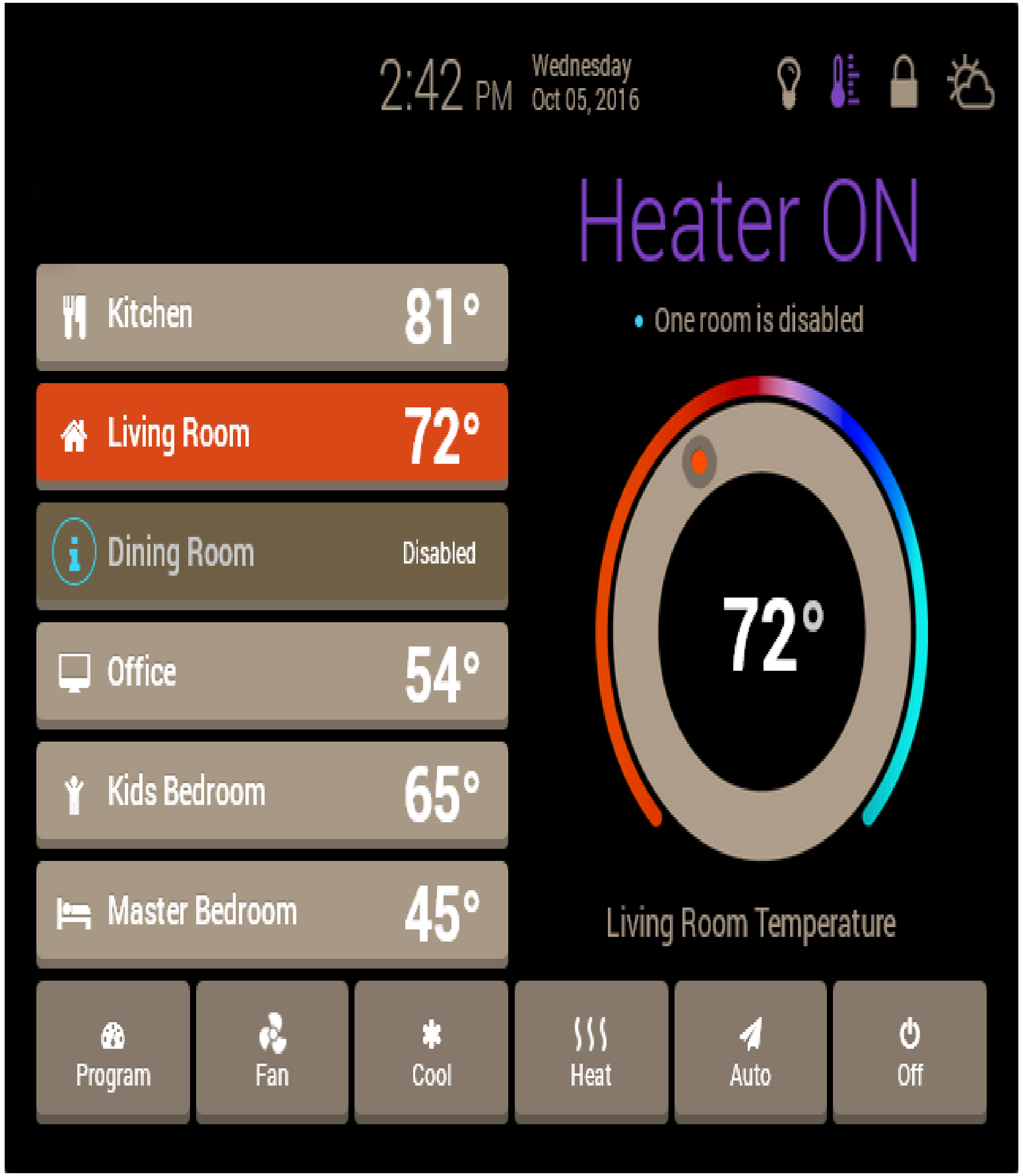
Data Processing

- Once the data is collected and it gets to the cloud, the software performs processing on the acquired data.



User Interface

- For example, if a user detects some changes in the refrigerator, the user can remotely adjust the temperature via their phone.
- There are also cases where some actions perform automatically.



2:42 PM Wednesday Oct 05, 2016

Heater ON

• One room is disabled

Kitchen 81°

Living Room 72°

Dining Room Disabled

Office 54°

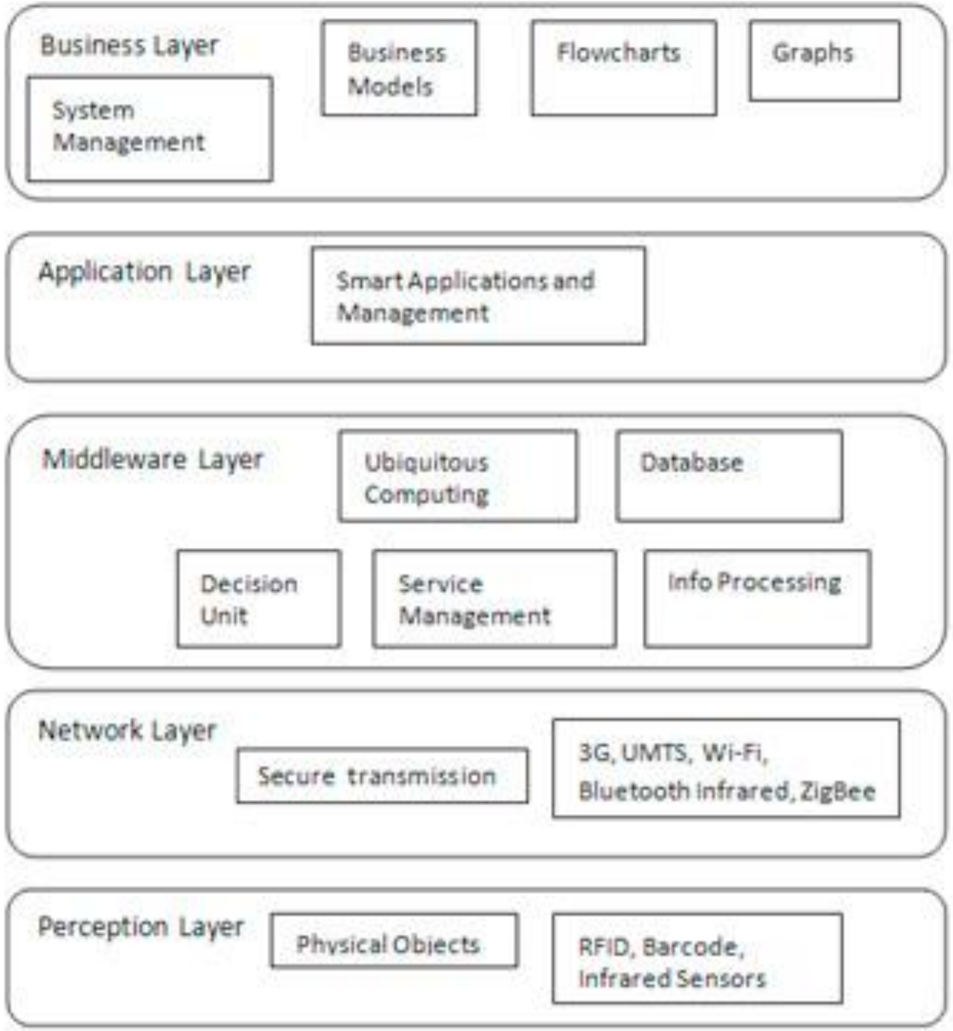
Kids Bedroom 65°

Master Bedroom 45°

72°

Living Room Temperature

IoT Architecture



Business Layer

System Management

Business Models

Flowcharts

Graphs

Application Layer

Smart Applications and Management

Middleware Layer

Ubiquitous Computing

Database

Decision Unit

Service Management

Info Processing

Network Layer

Secure transmission

3G, UMTS, Wi-Fi, Bluetooth Infrared, ZigBee

Perception Layer

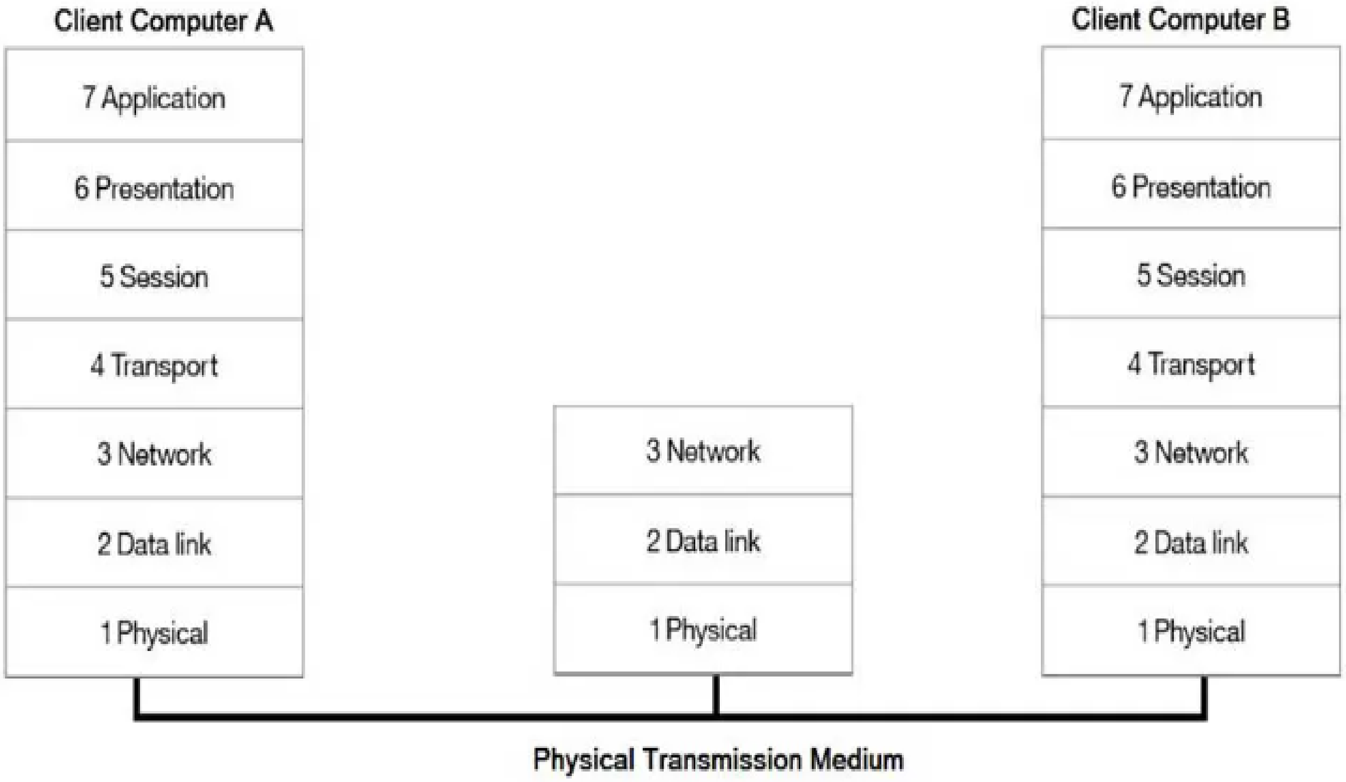
Physical Objects

RFID, Barcode, Infrared Sensors

- The **Perception layer** also called as ‘Device Layer’, is composed of physical devices and sensors. This layer works on identifying and collecting the information via sensor devices
- **Network Layer:** The Network layer is also known to be the ‘Transmission Layer’. This layer guarantees for the secure transfer of the information gathered from sensors to the information processing system
- **Middleware Layer:** Each smart object communicates with other devices only if they implement same service type. It takes the data from Network layer and stores it in the database. It processes information and decides the solution by analyzing the results.
- **Application Layer:** This layer is responsible for managing the application globally depending on the processing of objects’ information in the Middleware layer.
- **Business Layer:** This layer manages the complete IoT system in terms of the applications and services. It makes business models, graphs, flowcharts etc. on the basis of data obtained from previous layer. Depending on the result analysis, this layer will predict the future actions.

Architectural Overview of ISO/OSI

OSI model	TCP/IP model
7 Application	Application
6 Presentation	
5 Session	
4 Transport	Transport
3 Network	Internet
2 Data link	Network access & physical
1 Physical	



Client Computer A

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data link

1 Physical

3 Network

2 Data link

1 Physical

Client Computer B

7 Application

6 Presentation

5 Session

4 Transport

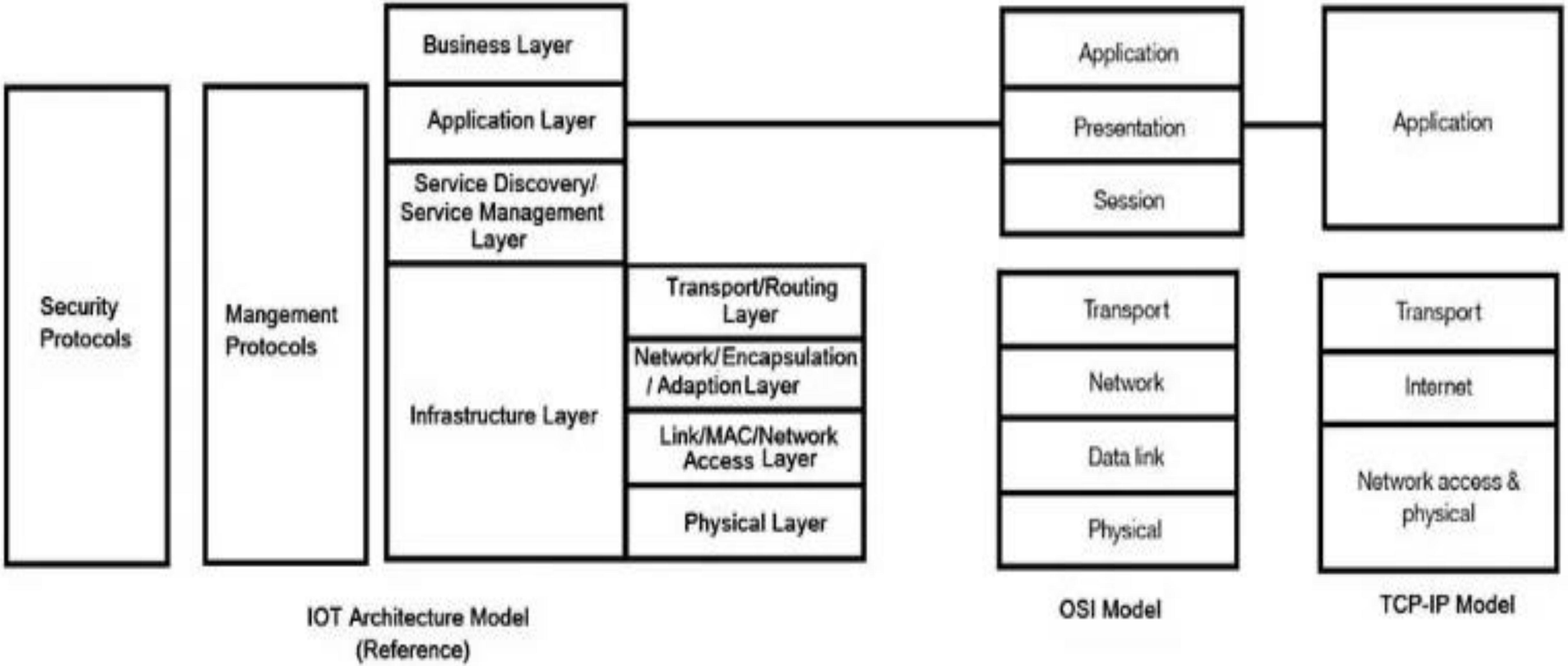
3 Network

2 Data link

1 Physical

Physical Transmission Medium

ISO/OSI vs IoT



Physical and Network Access Layer

Ethernet, Bluetooth Low Energy, Wireless HART, Zigbee, Z-wave, RFID, IEEE 802.11.ah, IEEE 802.15.4e, LoRaWAN, DASH7, Weightless, HomePlug, G.9959, LTE-A, DECT/ULE, ISA 100.11a, ANT, NFC, EPC Global, EddyStone, EnOcean, WiMax, NB-IOT, EC-GSM-IOT (Extended Coverage GSM-IOT), RPMA, LTE-MTC (LTE-Machine Type Communication), Cellular (GPRS/2G/3G/4G/5G), CDMA, Thread, INSTEON, DigiMesh

Network or Network Encapsulation or Adaption Layer

- IPv4
- IPv6
- 6LoWPAN
- 6Lo
- 6TiSCH
- IPv6 over Bluetooth Low Energy
- IPv6 over G.9959

Transport Layer

- TCP, UDP, DTLS, TLS, RPL, CARP, CORPL, QUIC, uIP, ROLL, Aeron, CCN (Content Centric Networking), NanoIP, TSMP (Time Synchronized Mesh Protocol)

Service Discovery or Service Management Layer

DNS-SD (DNS-Service Discovery), mDNS (Multicast Domain Name System)

uPnP, Simple Discovery Service Protocol,

Some of the currently available service discovery platforms and technologies are as follow –

- HyperCat
- Physical Web
- Wi-Fi Aware
- Bluetooth Beacons
- Open Hybrid
- Chirp

Application Layer

MQTT, SMQTT, CoAP, DDS , XMPP, AMQP, RESTful HTTP, MQTT-SN, STOMP, SMCP, LLAP, SSI, LWM2M, M3DA, XMPP-IOT, ONS 2.0, SOAP, Websocket, Reactive Streams, HTTP/2, JavaScript IOT

Business Layer and others

- Some of the device management protocols are as follow –

  OMA-DM

  TR-069

  OMA-CP

- Some of the data security protocols are as follow –

  Open Trust Protocol (OTrP)

  X.509

- Some of the popular Semantic Protocols are as follow –

  SensorML

  IOTDB

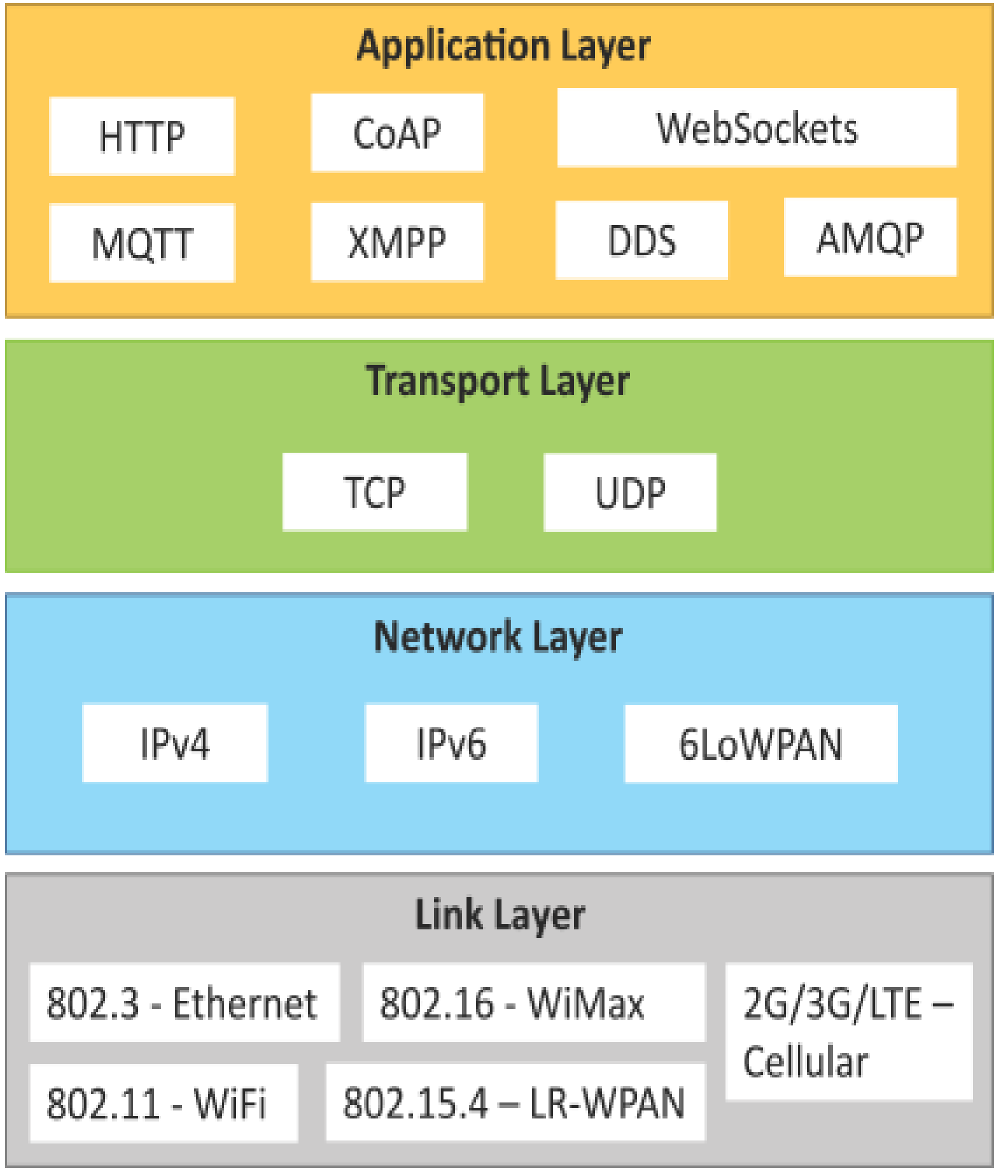
  RAML

  SENML

  Semantic Sensor Net Ontology, LsDL, Wolfram Language Connected Devices

IoT Protocols

- Link Layer
  - 802.3 – Ethernet
  - 802.11 – WiFi
  - 802.16 – WiMax
  - 802.15.4 – LR-WPAN
  - 2G/3G/4G
- Network/Internet Layer
  - IPv4
  - IPv6
  - 6LoWPAN
- Transport Layer
  - TCP
  - UDP
- Application Layer
  - HTTP
  - CoAP
  - WebSocket
  - MQTT
  - XMPP
  - DDS
  - AMQP



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

IoT Protocols…Link Layer…Ethernet

Sr.No	Standard	Shared medium
1	802.3	Coaxial Cable…10BASE5
2	802.3.i	Copper Twisted pair …..10BASE-T
3	802.3.j	Fiber Optic……10BASE-F
4	802.3.ae	Fiber…..10Gbits/s

Data Rates are provided from 10Gbit/s to 40Gb/s and higher

IoT Protocols…Link Layer…WiFi (Wireless Fidelity)

Sr.No	Standard	Operates in
1	802.11a	5 GHz band
2	802.11b and 802.11g	2.4GHz band
3	802.11.n	2.4/5 GHz bands
4	802.11.ac	5GHz band
5	802.11.ad	60Hz band

- Collection of Wireless LAN
- Data Rates from 1Mb/s to 6.75 Gb/s

IoT Protocols…Link Layer…WiMax (Wireless Inter-operability for Microwave Access)

Sr.No	Standard	Data Rate
1	802.16m	100Mb/s for mobile stations 1Gb/s for fixed stations

- Collection of Wireless Broadband standards
- Data Rates from 1.5Mb/s to 1 Gb/s

IoT Protocols…Link Layer…LR-WPAN

- Collection of standards for Low Rate - Wireless Personal Area Networks
- Basis for high level communication protocols such as Zigbee
- Data Rates from 40Kb/s to 250Kb/s
- Provide low-cost and low-speed communication for power constrained devices

IoT Protocols…Link Layer…2G/3G/4G – Mobile Communication

Sr.No	Standard	Operates in
1	2G	GSM-CDMA
2	3G	UMTS and CDMA 2000
3	4G	LTE

- Data Rates from 9.6Kb/s (for 2G) to up to 100Mb/s (for 4G)

IoT Protocols…Network/Internet Layer

- Responsible for sending of IP datagrams from source to destination network
- Performs the host addressing and packet routing
- Host identification is done using hierarchical IP addressing schemes such as IPV4 or IPV6

IoT Protocols…Network Layer

- IPV4
  - Used to identify the devices on a network using hierarchical addressing scheme
  - Uses 32-bit address scheme
- IPV6
  - Uses 128-bit address scheme
- 6LoWPAN (IPV6 over Low power Wireless Personal Area Network)
  - Used for devices with limited processing capacity
  - Operates in 2.4 Ghz
  - Data Rates of 250Kb/s

IoT Protocols…Transport Layer

- Provide end-to-end message transfer capability independent of the underlying network
- It provides functions such as error control, segmentation, flow-control and congestion control

IoT Protocols…TCP

TCP
ARE YOU GETTING ALL OF THIS?
NO, SLOW DOWN SO I CAN SAVE ACCURATELY!
SKULLBOX.NET

- Transmission Control Protocol
- Connection Oriented
- Ensures Reliable transmission
- Provides Error Detection Capability to ensure no duplicate of packets and retransmit lost packets
- Flow Control capability to ensure the sending data rate is not too high for the receiver process
- Congestion control capability helps in avoiding congestion which leads to degradation of n/w performance

IoT Protocols…UDP

- User Datagram Protocol
- Connectionless
- Does not ensures Reliable transmission
- Does not do connection before transmitting
- Does not provide proper ordering of messages
- Transaction oriented and stateless

IoT Protocols…Application Layer…Hyper Transfer Protocol

- Forms foundation of World Wide Web(WWW)
- Includes commands such as GET,PUT, POST, HEAD, OPTIONS, TRACE..etc
- Follows a request-response model
- Uses Universal Resource Identifiers(URIs) to identify HTTP resources

IoT Protocols…Application Layer…CoAP

- Constrained Application Protocol
- Used for Machine to machine (M2M) applications meant for constrained devices and n/w's
- Web transfer protocol for IoT and uses request-response model
- Uses client –server architecture
- Supports methods such as GET,POST, PUT and DELETE

INTERNET of THINGS

CoAP

Constrained Application Protocol
(Web Protocol for IoT)

Anirudha Chakrabarti
Associate Vice President and Chief Architect, Digital Practice, Mphasis
[illegible]

IoT Protocols…Application Layer…WebSocket

- Allows full-duplex communication over single socket
- Based on TCP
- Client can be a browser, IoT device or mobile application

IoT Protocols…Application Layer…MQTT

- Message Queue Telemetry Transport , light-weight messaging protocol
- Based on publish-subscribe model
- Well suited for constrained environments where devices have limited processing, low memory and n/w bandwidth requirement

IoT Protocols…Application Layer…XMPP

- Extensible messaging and presence protocol
- For Real time communication and streaming XML data between n/w entities
- Used for Applications such as Multi-party chat and voice/video calls.
- Decentralized protocol and uses client server architecture.

IoT Protocols…Application Layer…DDS

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

IoT Protocols…Application Layer…AMQP

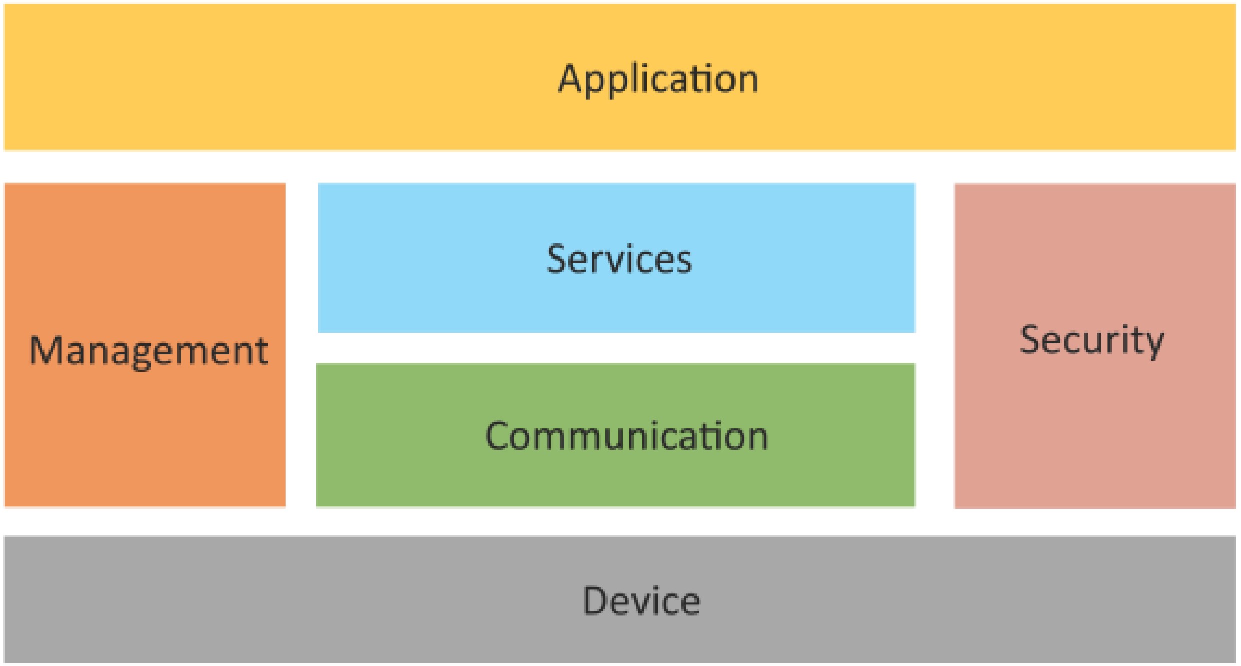
- Advanced Messaging Queuing Protocol used for business messaging.
- Supports both point-to-point and publisher/subscriber models, routing and queuing
- Broker here receives messages from publishers and route them over connections to consumers through messaging queues.

Logical Design of IoT

- IoT Functional Blocks
- IoT Communication Models
- IoT Communication API

IoT Functional Blocks

- 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.



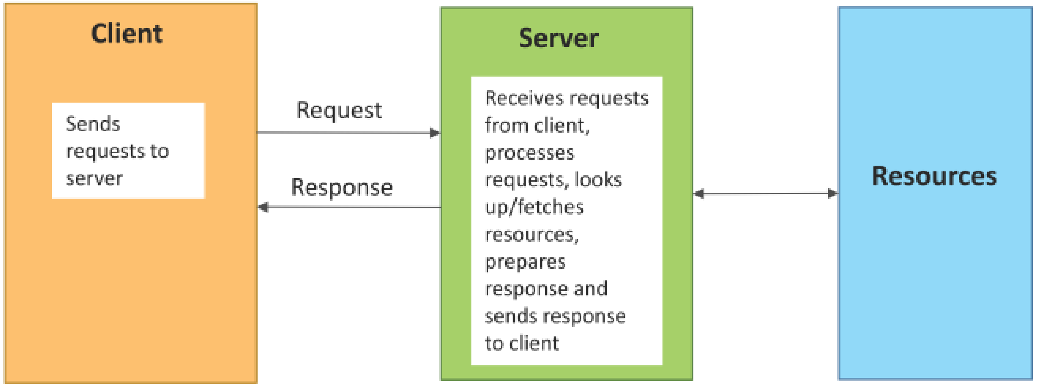
Functional Blocks of IoT

IoT Functional Blocks

- Device : Devices such as sensing, actuation, monitoring and control functions.
- Communication : IoT Protocols
- Services like device monitoring, device control services, data publishing services and device discovery
- Management : Functions to govern the system
- Security : Functions as authentication, authorization, message and content integrity, and data security
- Applications

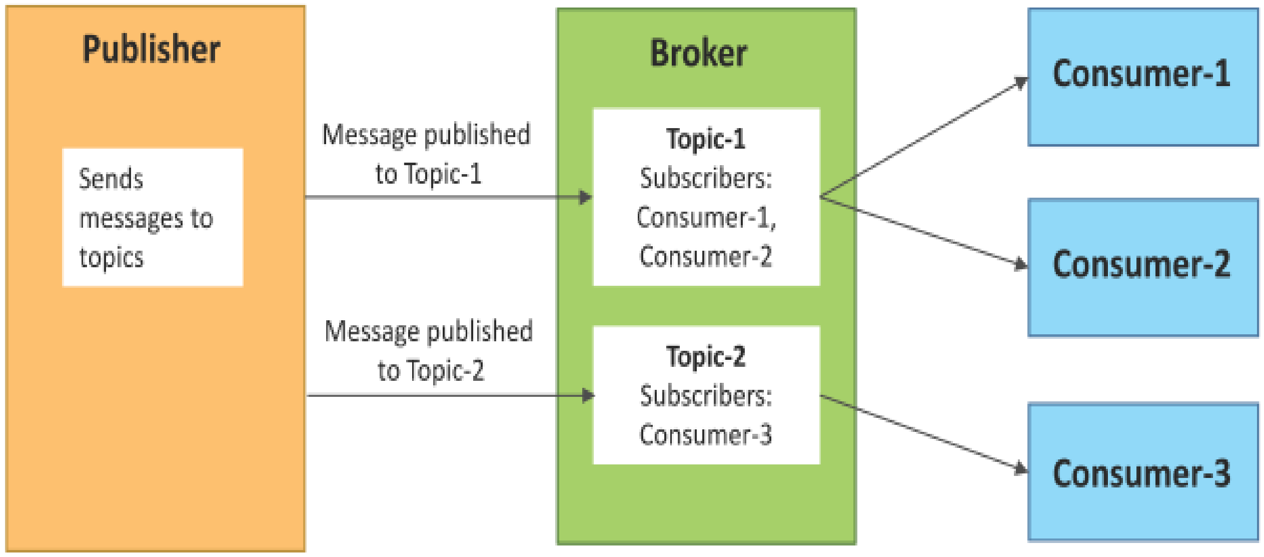
Communication Model

Request–Response Communication Model

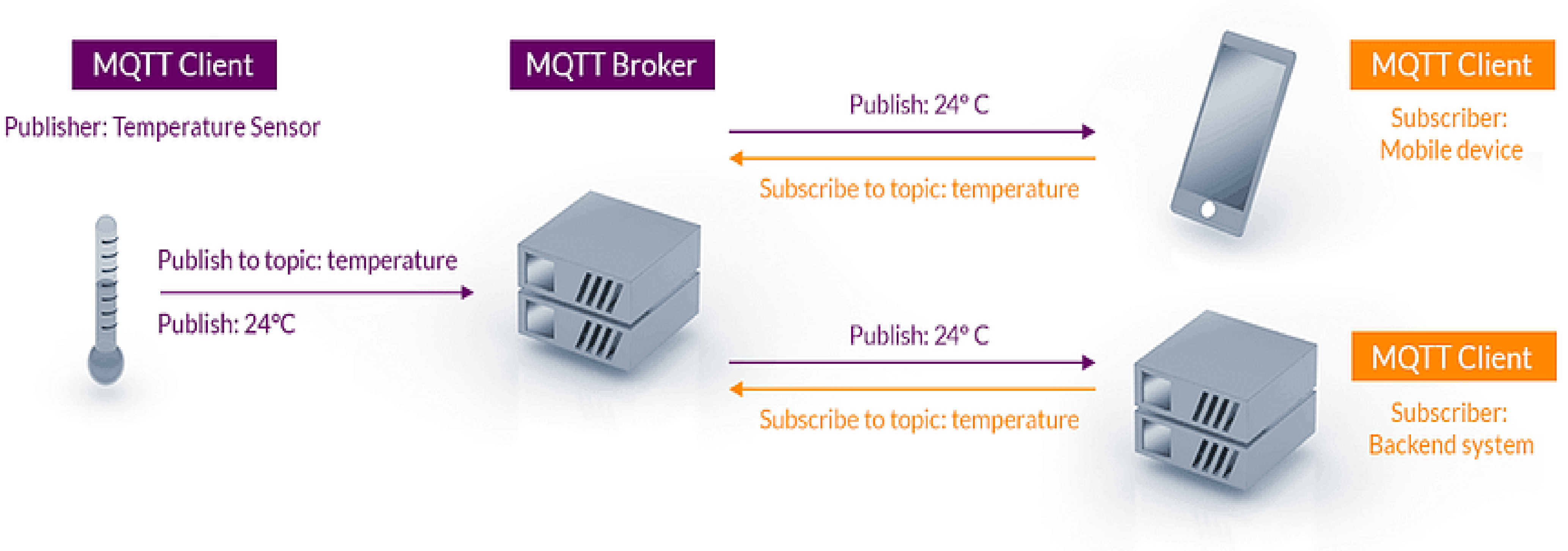

Client
Sends requests to server
Request
Response
Server
Receives requests from client, processes requests, looks up/fetches resources, prepares response and sends response to client
Resources

- 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

Publish–Subscribe Communication Model



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


MQTT Client
Publisher: Temperature Sensor
MQTT Broker
Publish to topic: temperature
Publish: 24°C
Publish: 24° C
Subscribe to topic: temperature
MQTT Client
Subscriber: Mobile device
Publish: 24° C
Subscribe to topic: temperature
MQTT Client
Subscriber: Backend system

Advantages of Pub/Sub Model in IoT:

1.**Scalability:** Easily scales to a large number of devices without direct connections between them.

2.**Flexibility: P**ublishers and subscribers can operate independently, allowing for dynamic changes in the network.

3.**Decoupling:** Decouples the sender (publisher) from the receiver (subscriber), providing flexibility in the system architecture.

4.**Efficiency:** Reduces unnecessary communication, as subscribers only receive messages relevant to their interests.

Push–Pull Communication Model

1.Publisher:

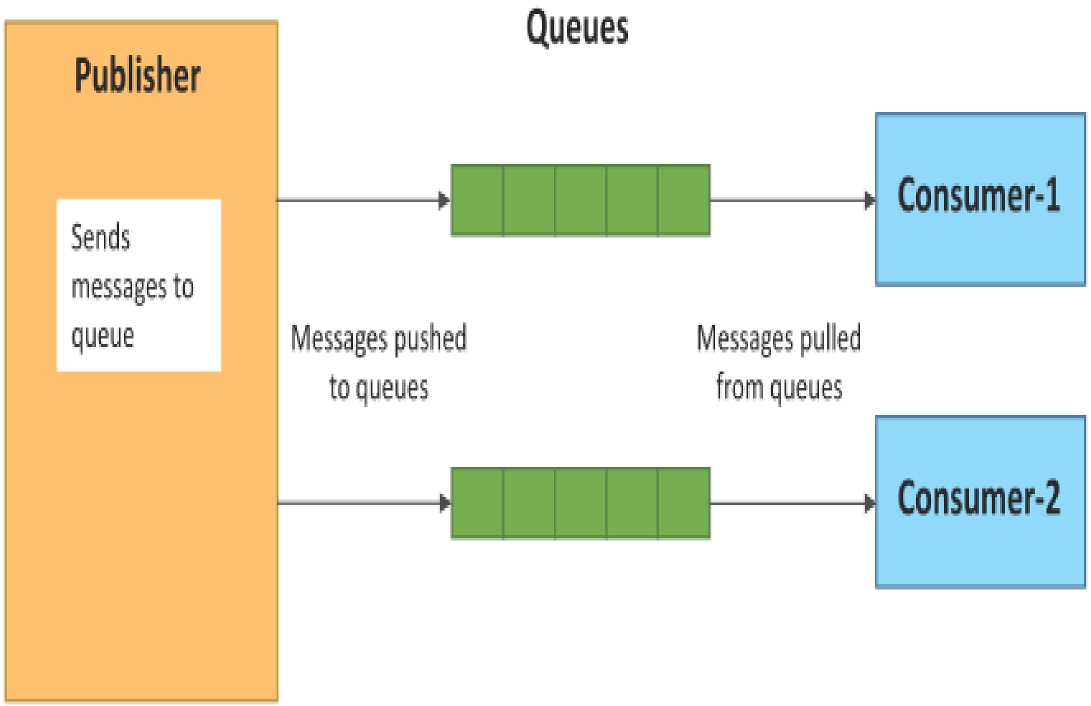
1. Generates and sends messages, similar to the traditional Pub/Sub model.
2. Notifies subscribers actively when there's new information.

2.Subscriber:

1. Expresses interest in specific topics or types of messages.
2. Actively pulls information from the publisher when needed.

3.Broker (Optional):

1. Can still act as an intermediary, but not strictly necessary.
2. Distributes messages from publishers to subscribers or facilitates communication.



1. Publisher Pushes Updates:

1. The publisher generates new information and actively pushes updates to the subscribers who have expressed interest in the corresponding topics.

2. Subscriber Pulls Information:

1. Subscribers actively pull information from the publisher when they are ready to receive updates.
2. Subscribers control the frequency and timing of information retrieval.

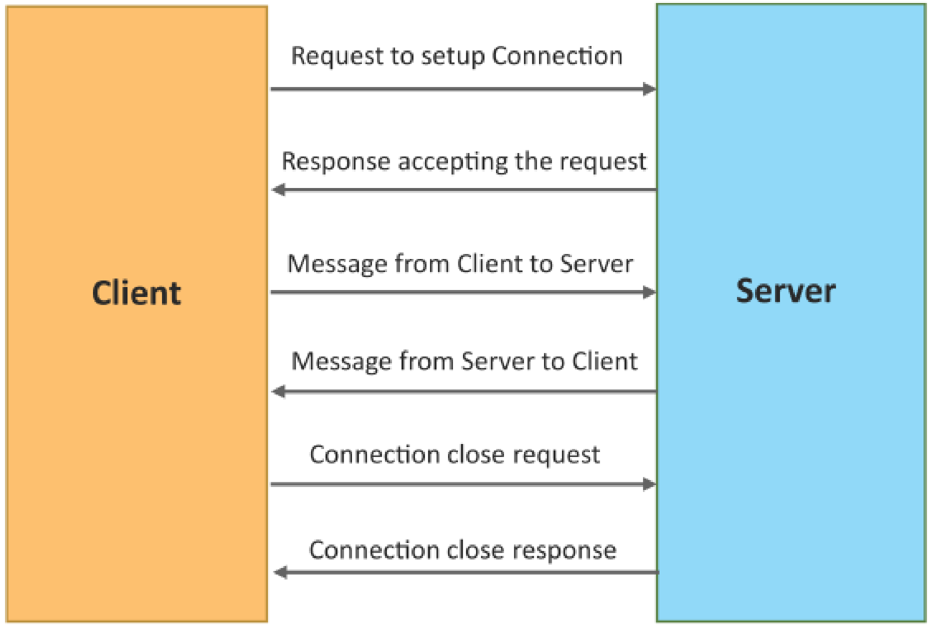
3. Two-Way Communication:

1. Unlike the traditional Pub/Sub model, the Push-Pull model allows for bidirectional communication.
2. Publishers can actively notify subscribers, and subscribers can request information as needed.

1.Reduced Latency: Subscribers can receive updates in near real-time as publishers actively push information.
2.Efficient Resource Usage: Subscribers have more control over when and how frequently they retrieve information, reducing unnecessary communication.
3.Flexibility: Subscribers can pull information based on their own requirements, making the model more flexible and adaptive.

Exclusive Pair Communication Model

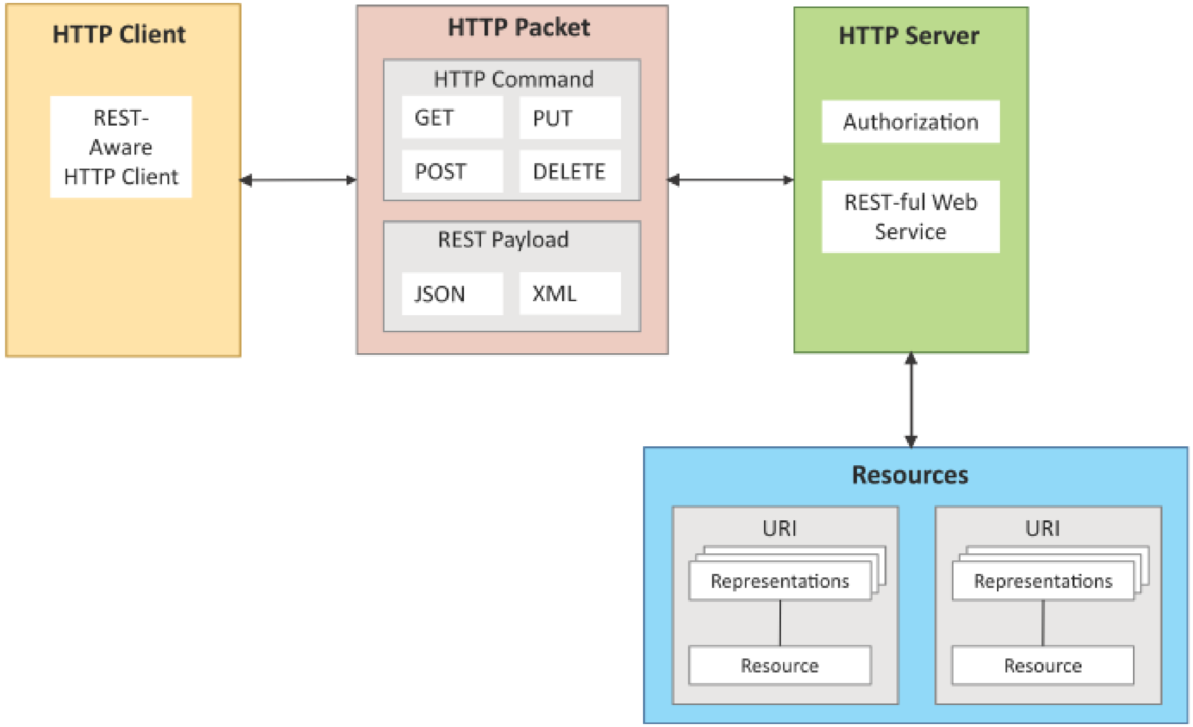
- Exclusive Pair is a bidirectional, fully duplex communication model that uses a persistent connection between the client and the server.
- Once the connection is set up it, remains open until the client sends a request to close the connection.
- Client and server can send messages to each other after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


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

IoT Communication APIs

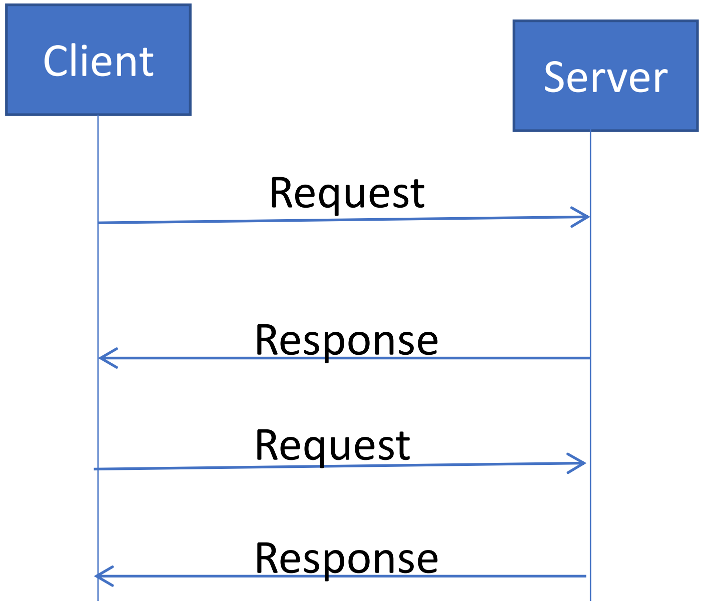
REST-based Communication APIs

- 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**.
- There are several constraints that define the principles and characteristics of a RESTful system.
- REST architectural constraints apply to the components, connectors and data elements within a distributed hypermedia system.



REST-based Communication APIs Constraints

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



Client

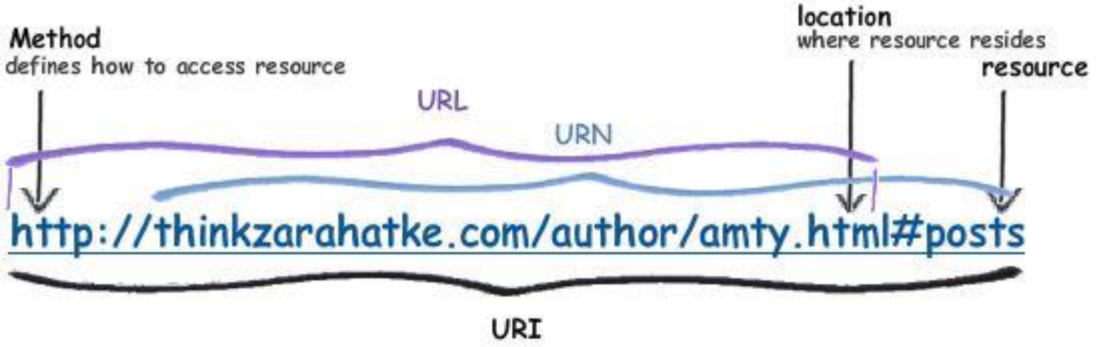
Server

Request

Response

Request

Response



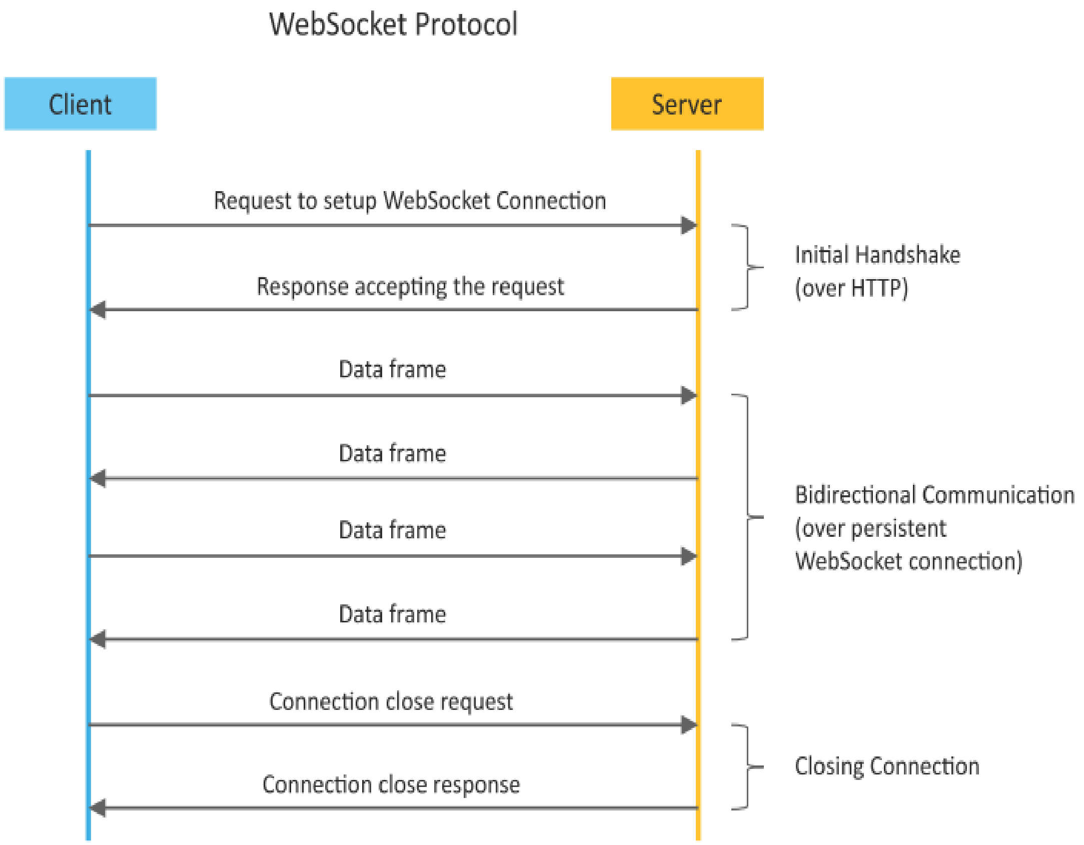
Method
defines how to access resource
location
where resource resides
resource
URL
URN
http://thinkzarahatke.com/author/amty.html#posts
URI

WebSocket-based Communication APIs

- WebSocket APIs **allow bi-directional, real-time communication** between clients and servers.
- WebSocket APIs follow the **exclusive pair communication model**.

Key Components

- WebSocket Protocol, Handshake, WebSocket API Endpoints (URIs), WebSocket Frames, Message Types, WebSocket Events


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 Communication APIs

Comparison Based on	REST	Websocket
State	Stateless	Statefull
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.
Ease of Use	Simple	Complex

IoT Enabling Technologies

- Wireless Sensor Network (WSN)
- Cloud Computing
- Big Data Analytics

WSN

- WSN is a subset of IoT. It specifically refers to networks of sensors that communicate wirelessly to collect and transmit data from the physical environment. WSN focuses on the sensing aspect and is often dedicated to monitoring and data collection within a particular area.
- **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
- WSN ENABLESD BY 802.15.4 Operates at 2.4Ghtz, 250KB/s range 10 to 100 mts
- WSN is self organzied

Example

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

Protocols

- Zigbee

Cloud Computing

Cloud computing plays a crucial role in the Internet of Things (IoT) ecosystem by providing scalable, flexible, and cost-effective **solutions for managing and processing the massive amounts of data generated** by IoT devices

key roles of cloud computing in IoT:

- Data Storage
  - Scalable Storage
  - Data Replication and Backup
- Data Processing and Analytics
  - Big Data Analytics
  - Real-time Data Processing
- Device Management
  - Remote Device Management
  - Firmware Updates
- Scalability and Flexibility
  - Resource Scaling
  - Pay-as-You-Go Model

- Security
  - Secure Data Transmission
  - Identity and Access Management
- Integration with Other Services
- Edge Computing Integration

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.
- Big Data Analytics plays a crucial role in the Internet of Things (IoT) ecosystem by providing the means to **process, analyze, and derive meaningful insights** from the massive volumes of data generated by IoT devices.
- It involves data cleansing, processing and visualization

key roles of Big Data Analytics in IoT:

- Data Processing
  - Data Filtering and Cleansing
  - Normalization and Standardization
- Real-Time Analytics
  - Immediate Insights
  - Predictive Maintenance
- Pattern Recognition
  - Anomaly Detection
  - Behavioral Analysis
- Scalable Data Storage
  - Scalable Storage Solutions
  - Historical Data Analysis
- Advanced Analytics
- Customized User Experiences
  - Personalization
- Optimizing IoT Networks

IoT Levels & Deployment Templates

An IoT system comprises of the following components:

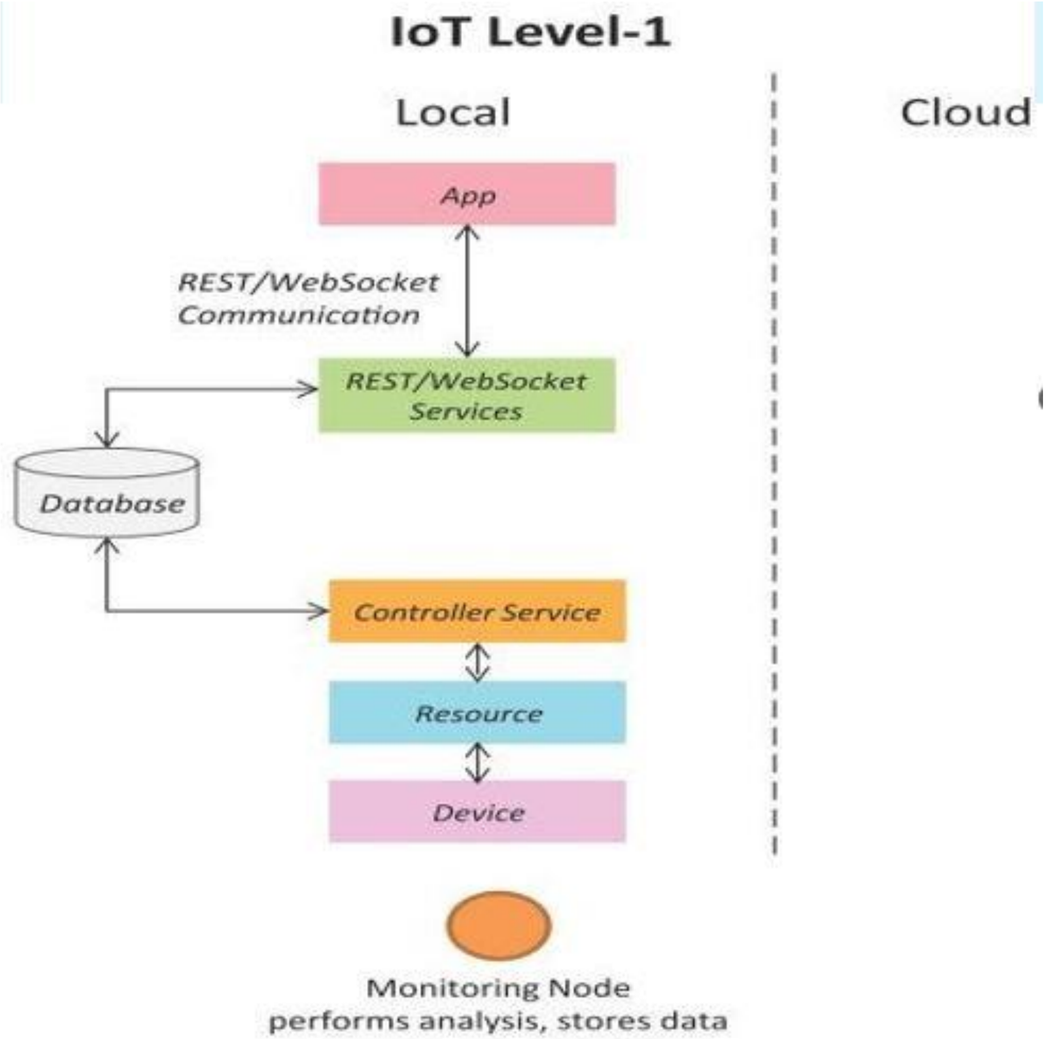
- **Device**: An IoT device allows identification, remote sensing, actuating and remote monitoring capabilities. You learned about various examples of IoT devices in section
- **Resource**: Resources are software components on the IoT device for accessing, processing, and storing sensor information, or controlling actuators connected to the device. Resources also include the software components that enable network access for the device.
- **Controller Service**: Controller service is a native service that runs on the device and interacts with the web services. Controller service sends data from the device to the web service and receives commands from the application (via web services) for controlling the device.

IoT Levels & Deployment Templates

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

IoT Level-1

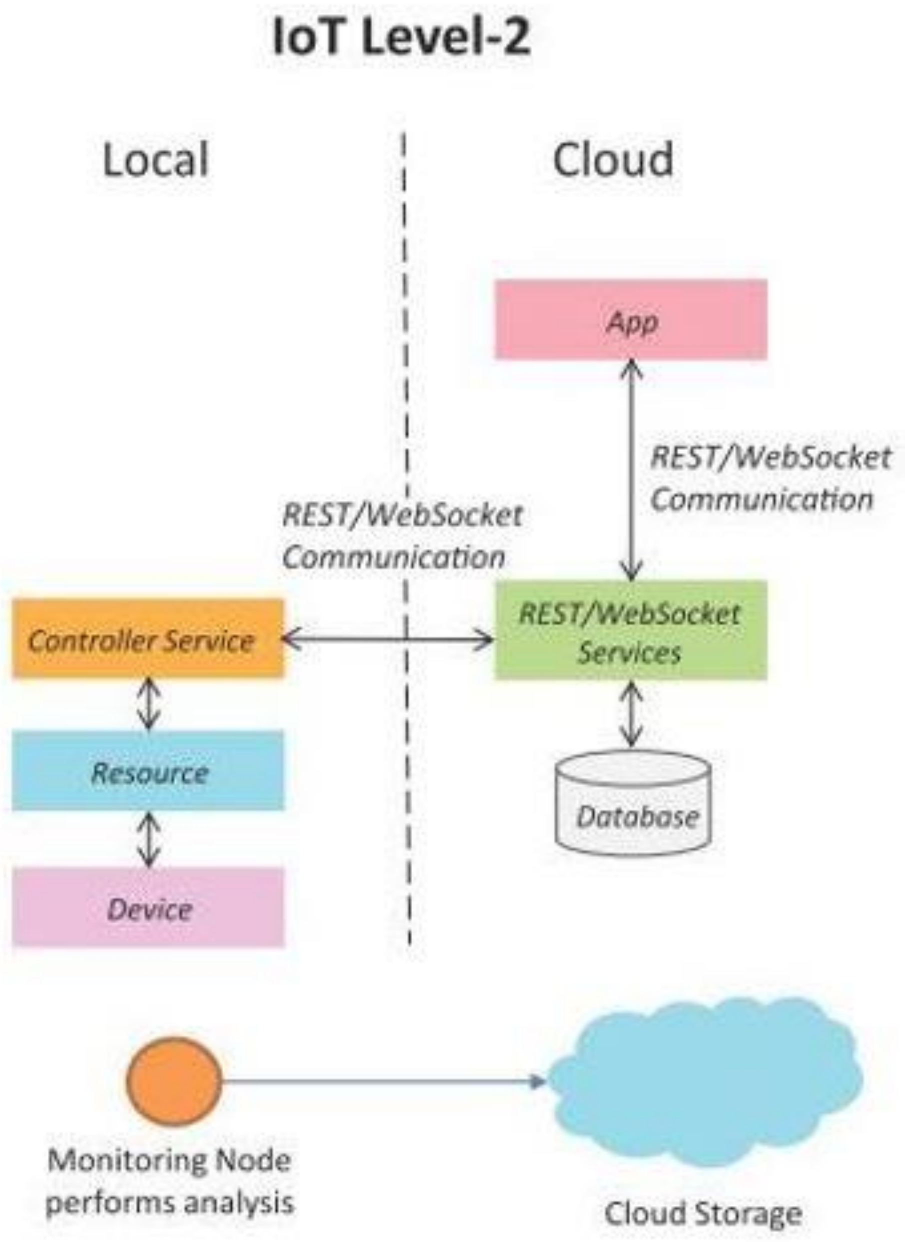
- A level-1 IoT system has a single node/device that performs sensing and/or actuation, stores data, performs analysis and hosts the application
- Level-1 IoT systems are suitable for modeling low-cost and low-complexity solutions where the data involved is not big and the analysis requirements are not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



- Ex: **IoT Home automation**

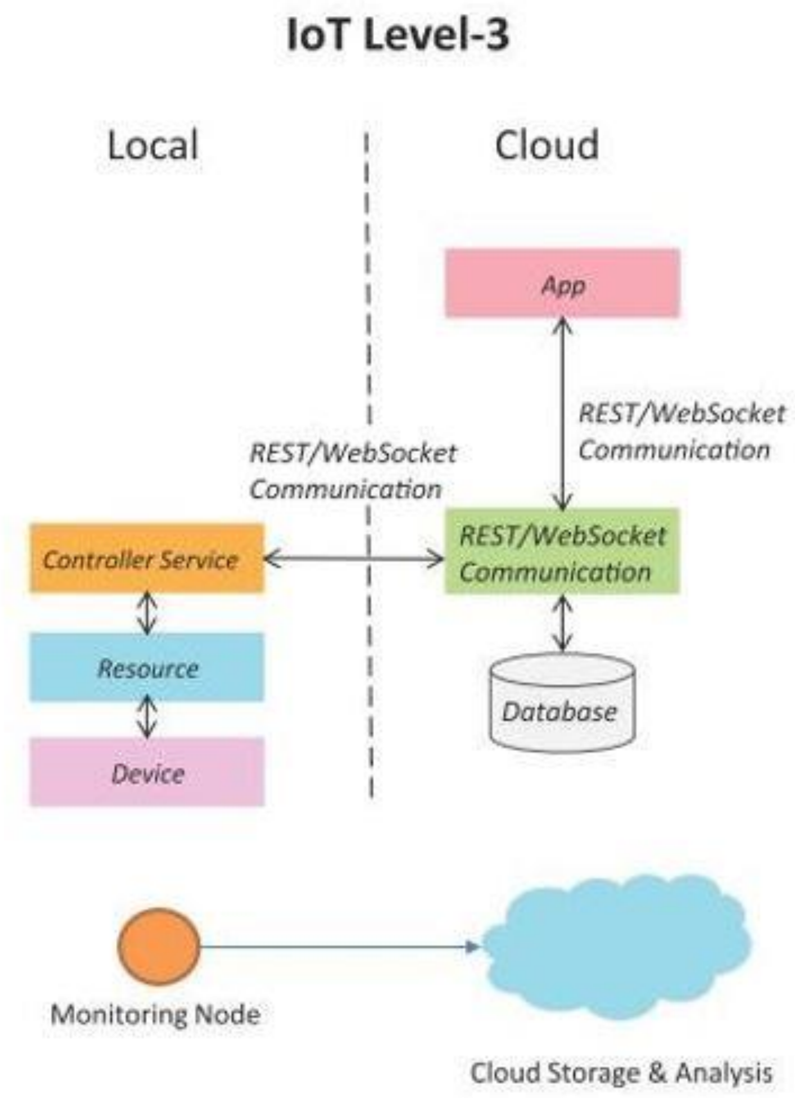
IoT Level-2

- 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, however, the primary analysis requirement is not computationally intensive and can be done locally itself.
- Ex: **IoT system for Smart Irrigation**



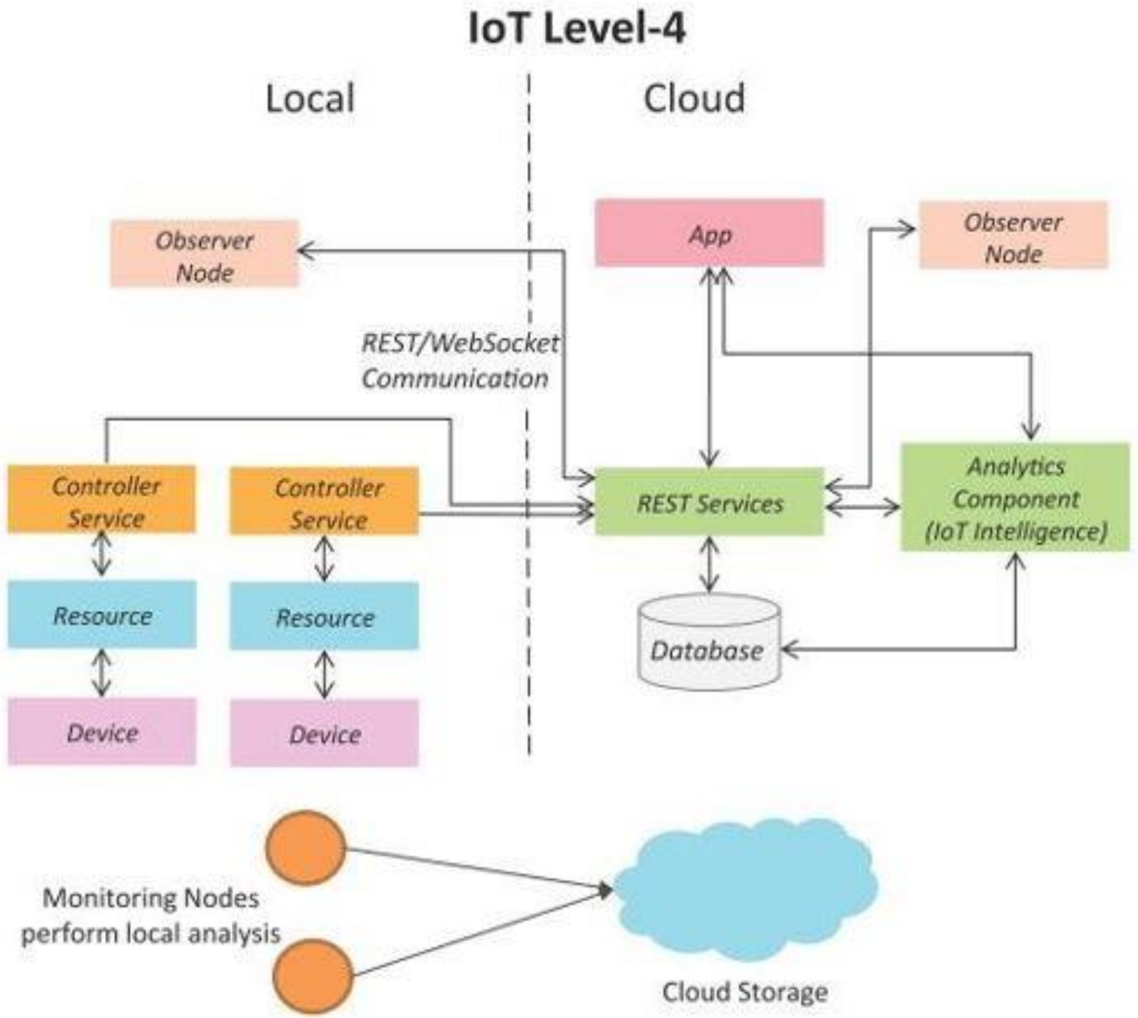
IoT Level-3

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



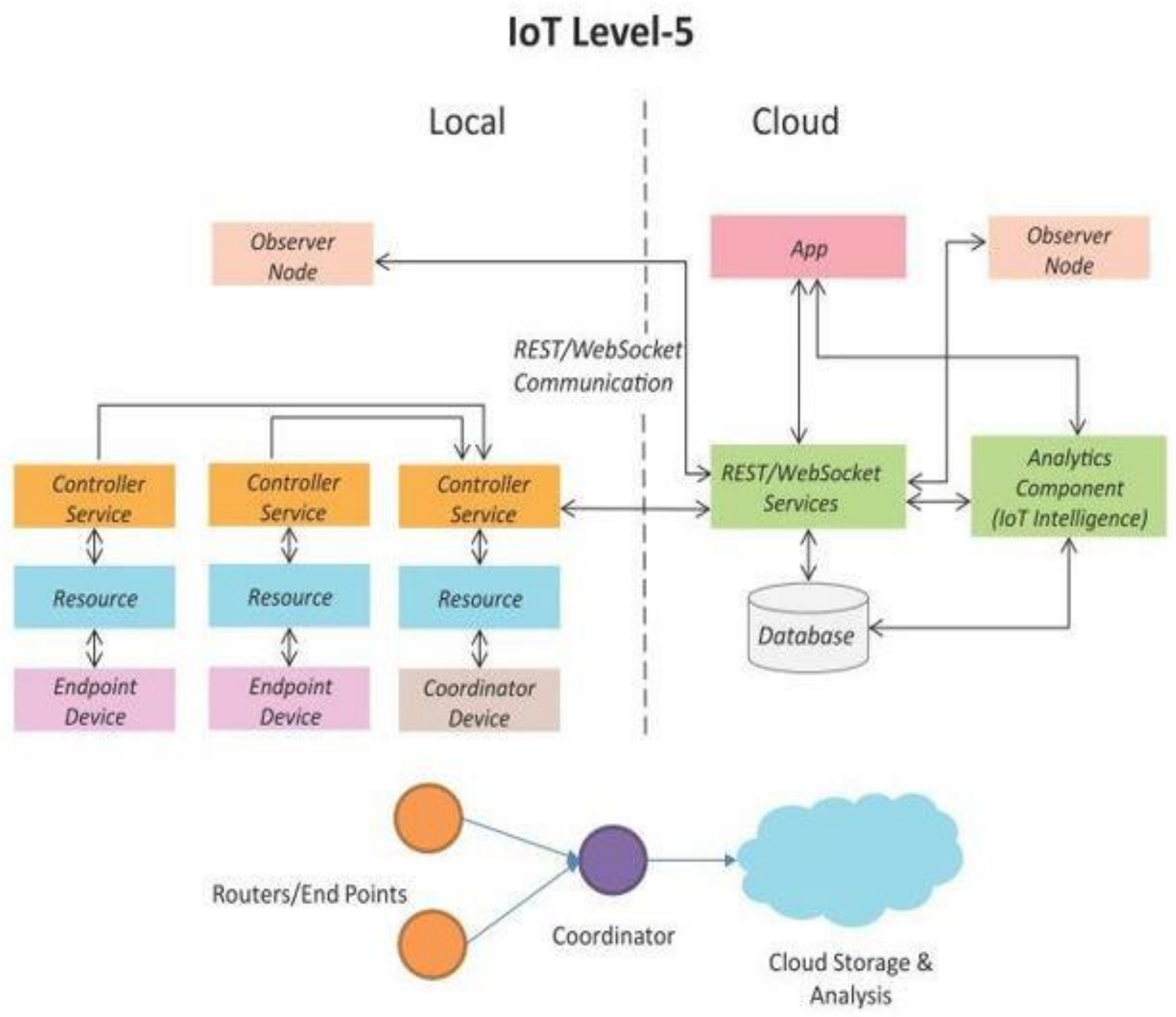
IoT Level-4

- A level-4 IoT system has multiple nodes that perform local analysis. Data is stored in the cloud and application is cloud-based.
- 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.
- **Ex: IoT Noise Monitoring System**



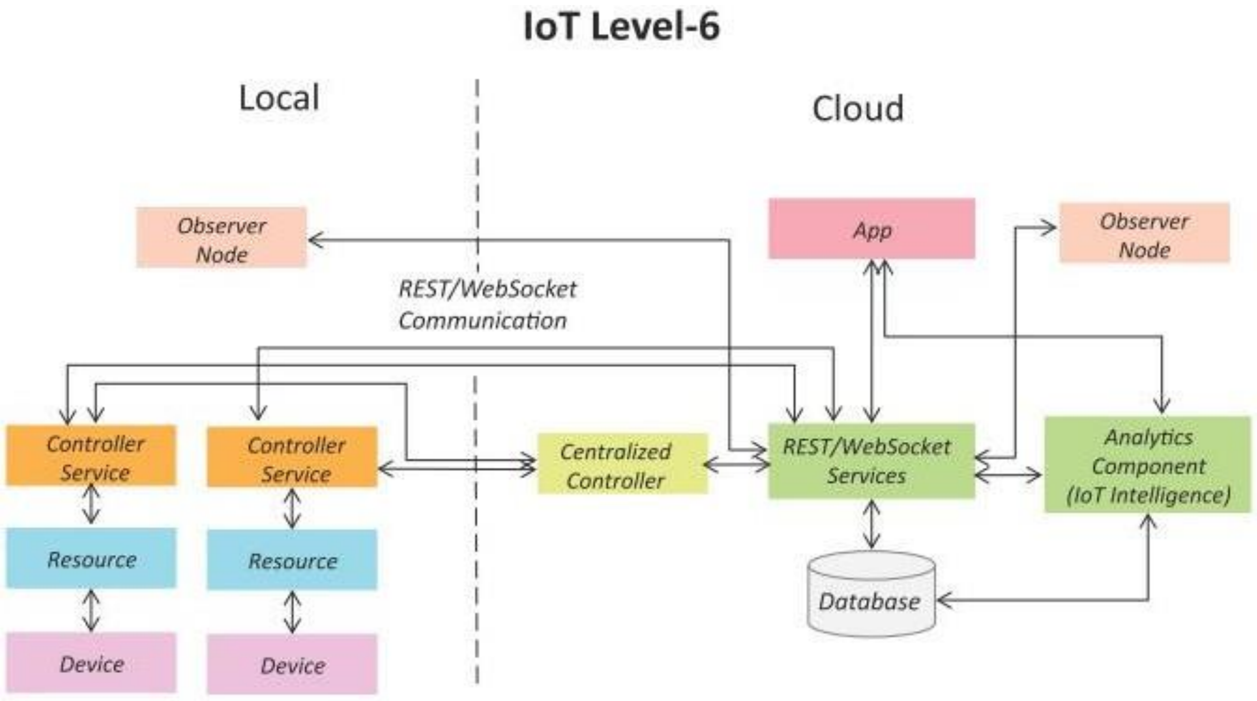
IoT Level-5

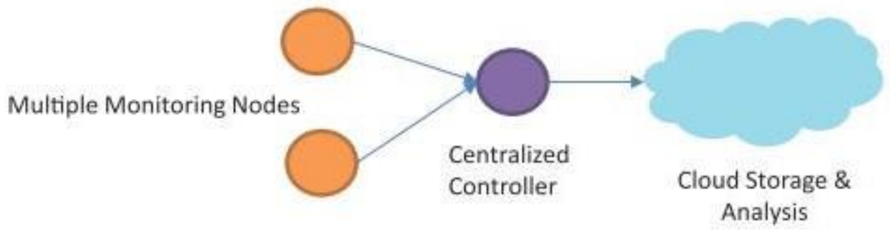
- A level-5 IoT system has multiple end nodes and one coordinator node.
- The end nodes that perform sensing and/or actuation.
- Coordinator node collects data from the end nodes and sends to the cloud.
- Data is stored and analyzed in the cloud and application is cloud-based.
- Level-5 IoT systems are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



IoT Level-6

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


IoT Level-6
Local
Cloud
Observer Node
REST/WebSocket Communication
App
Observer Node
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