Internet of Things

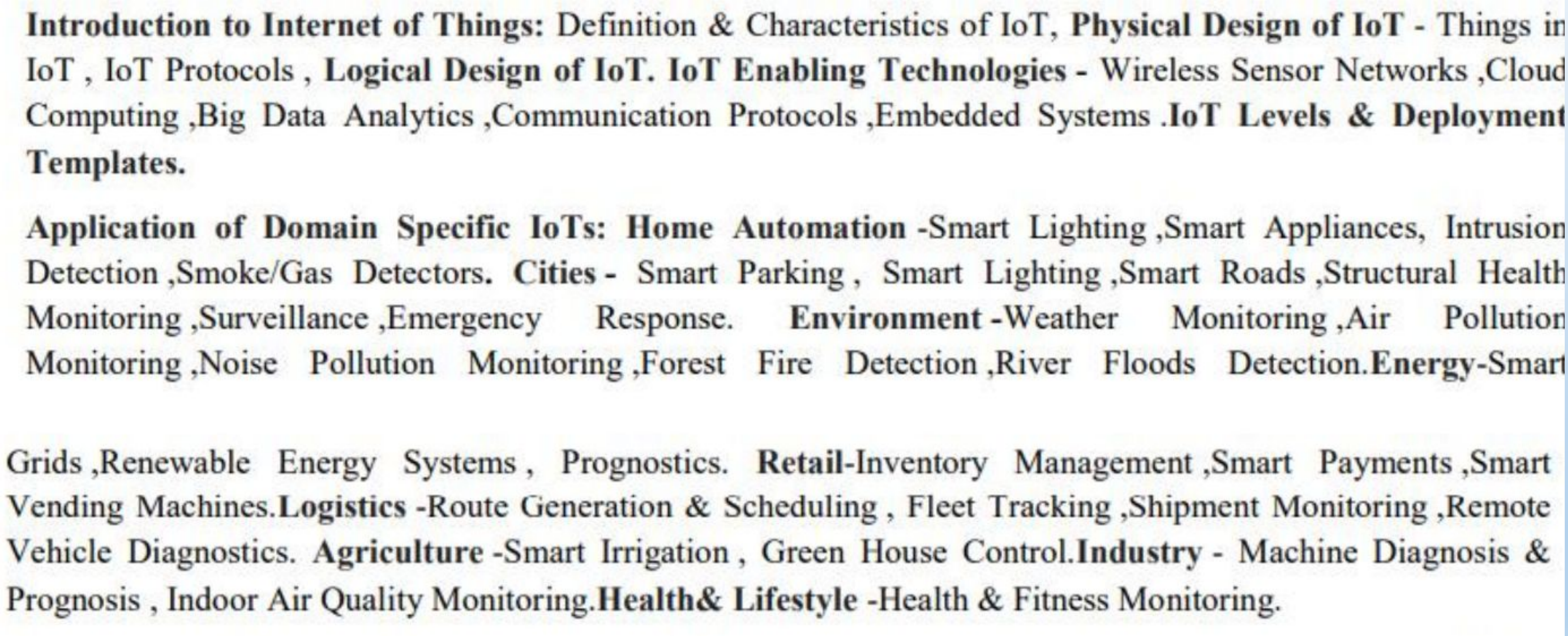
Course Instructor: Akshaya K Pati

Office:
Sensing and Computing Lab
Campus 12, New Building, 3rd Floor
m:8260946353

- Course Outcome:
- □Implement data and knowledge management and use of devices in IoT technology.
- □Understand the application areas of IOT ·
- □Understand building blocks of Internet of Things and characteristics.
- □Understand the State of Art-IoT Architecture.

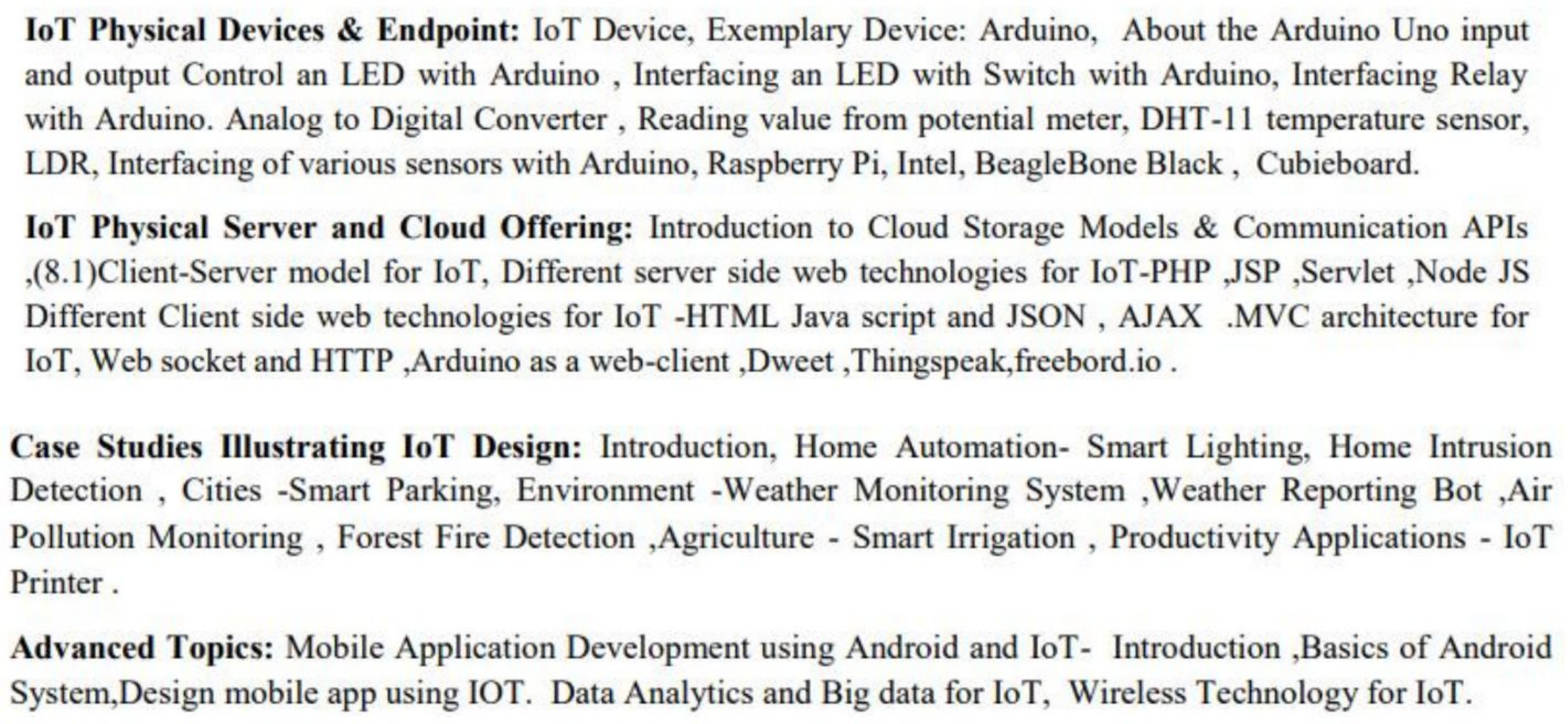
- Text Book:
- T1.Arshadeep Bahga,Vijay Madisetti, "Internet of Things -A Hands-on Approach", Universities Press, 1st Edition, ISBN:9788173719547.
- Reference Books:
- R1. Adrian McEwen, Hakim Cassimally, "Designing the Internet of Things", Wiley Publication, 1st Edition, November 2013,ISBN:9781118430620.
- R2. Harry Fairhead ,"Raspberry Pi IOT in C", IO Press Publication, 1st Edition, ISBN:9781871962468.

Syllabus



Introduction to Internet of Things: Definition & Characteristics of IoT, **Physical Design of IoT -** Things in IoT , IoT Protocols , **Logical Design of IoT. IoT Enabling Technologies -** Wireless Sensor Networks ,Cloud Computing ,Big Data Analytics ,Communication Protocols ,Embedded Systems .**IoT Levels & Deployment Templates.**

Application of Domain Specific IoTs: Home Automation -Smart Lighting ,Smart Appliances, Intrusion Detection ,Smoke/Gas Detectors. **Cities -** Smart Parking , Smart Lighting ,Smart Roads ,Structural Health Monitoring ,Surveillance ,Emergency Response. **Environment -**Weather Monitoring ,Air Pollution Monitoring ,Noise Pollution Monitoring ,Forest Fire Detection ,River Floods Detection.**Energy**-Smart Grids ,Renewable Energy Systems , Prognostics. **Retail**-Inventory Management ,Smart Payments ,Smart Vending Machines.**Logistics** -Route Generation & Scheduling , Fleet Tracking ,Shipment Monitoring ,Remote Vehicle Diagnostics. **Agriculture** -Smart Irrigation , Green House Control.**Industry -** Machine Diagnosis & Prognosis , Indoor Air Quality Monitoring.**Health& Lifestyle** -Health & Fitness Monitoring.

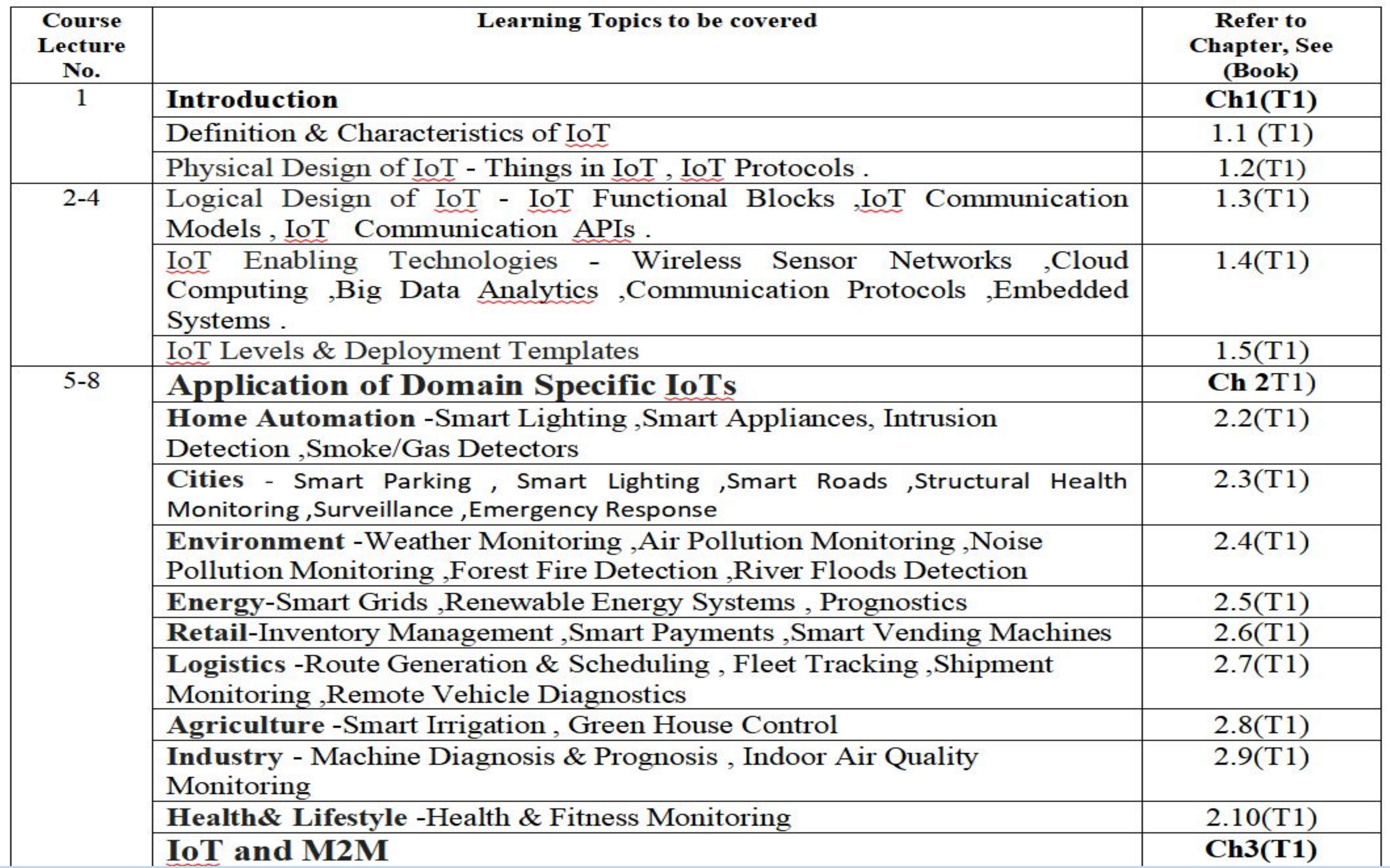


IoT Physical Devices & Endpoint: IoT Device, Exemplary Device: Arduino, About the Arduino Uno input and output Control an LED with Arduino , Interfacing an LED with Switch with Arduino, Interfacing Relay with Arduino. Analog to Digital Converter , Reading value from potential meter, DHT-11 temperature sensor, LDR, Interfacing of various sensors with Arduino, Raspberry Pi, Intel, BeagleBone Black , Cubieboard.

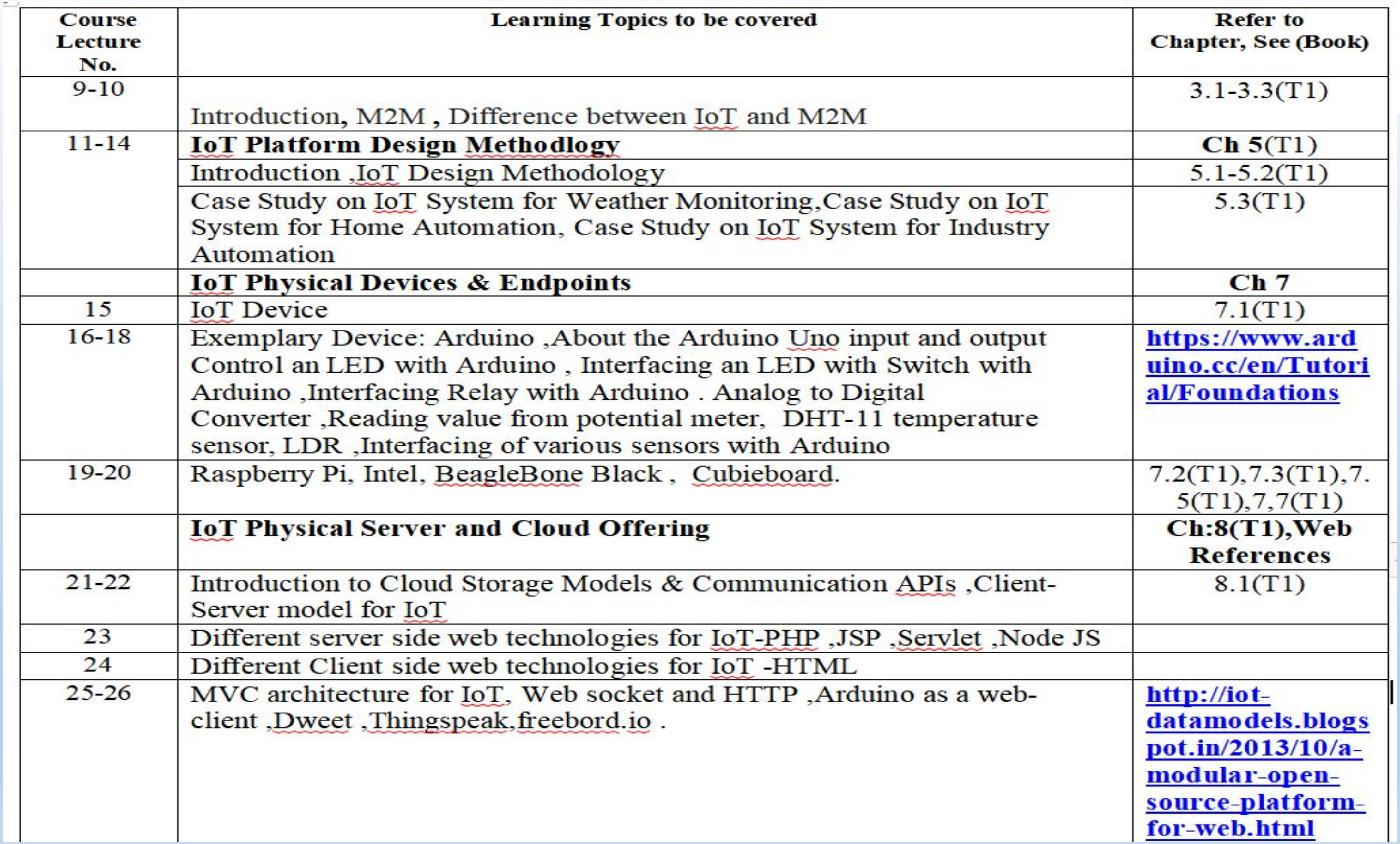
IoT Physical Server and Cloud Offering: Introduction to Cloud Storage Models & Communication APIs ,(8.1)Client-Server model for IoT, Different server side web technologies for IoT-PHP ,JSP ,Servlet ,Node JS Different Client side web technologies for IoT -HTML Java script and JSON , AJAX .MVC architecture for IoT, Web socket and HTTP ,Arduino as a web-client ,Dweet ,Thingspeak,freebord.io .

Case Studies Illustrating IoT Design: Introduction, Home Automation- Smart Lighting, Home Intrusion Detection , Cities -Smart Parking, Environment -Weather Monitoring System ,Weather Reporting Bot ,Air Pollution Monitoring , Forest Fire Detection ,Agriculture - Smart Irrigation , Productivity Applications - IoT Printer .

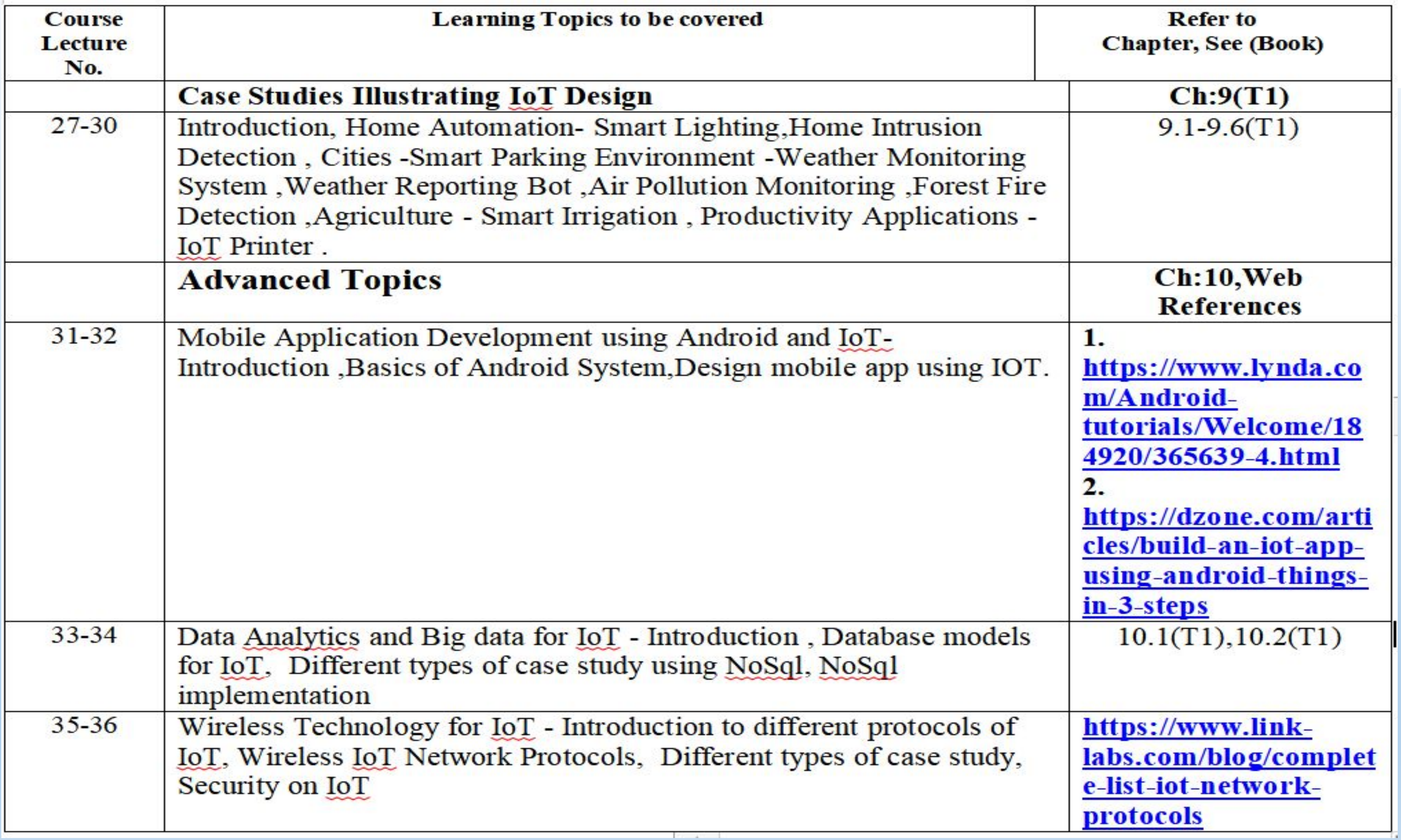
Advanced Topics: Mobile Application Development using Android and IoT- Introduction ,Basics of Android System,Design mobile app using IOT. Data Analytics and Big data for IoT, Wireless Technology for IoT.



Course Lecture No.	Learning Topics to be covered	Refer to Chapter, See (Book)
1	**Introduction**	**Ch1(T1)**
	Definition & Characteristics of IoT	1.1 (T1)
	Physical Design of IoT - Things in IoT , IoT Protocols .	1.2(T1)
2-4	Logical Design of IoT - IoT Functional Blocks ,IoT Communication Models , IoT Communication APIs .	1.3(T1)
	IoT Enabling Technologies - Wireless Sensor Networks ,Cloud Computing ,Big Data Analytics ,Communication Protocols ,Embedded Systems .	1.4(T1)
	IoT Levels & Deployment Templates	1.5(T1)
5-8	**Application of Domain Specific IoTs**	**Ch 2**T1)
	Home Automation -Smart Lighting ,Smart Appliances, Intrusion Detection ,Smoke/Gas Detectors	2.2(T1)
	Cities - Smart Parking , Smart Lighting ,Smart Roads ,Structural Health Monitoring ,Surveillance ,Emergency Response	2.3(T1)
	Environment -Weather Monitoring ,Air Pollution Monitoring ,Noise Pollution Monitoring ,Forest Fire Detection ,River Floods Detection	2.4(T1)
	Energy-Smart Grids ,Renewable Energy Systems , Prognostics	2.5(T1)
	Retail-Inventory Management ,Smart Payments ,Smart Vending Machines	2.6(T1)
	Logistics -Route Generation & Scheduling , Fleet Tracking ,Shipment Monitoring ,Remote Vehicle Diagnostics	2.7(T1)
	Agriculture -Smart Irrigation , Green House Control	2.8(T1)
	Industry - Machine Diagnosis & Prognosis , Indoor Air Quality Monitoring	2.9(T1)
	Health& Lifestyle -Health & Fitness Monitoring	2.10(T1)
	IoT and M2M	**Ch3(T1)**



Course Lecture No.	Learning Topics to be covered	Refer to Chapter, See (Book)
9-10	Introduction, M2M , Difference between IoT and M2M	3.1-3.3(T1)
11-14	**IoT Platform Design Methodlogy**	**Ch 5**(T1)
	Introduction ,IoT Design Methodology	5.1-5.2(T1)
	Case Study on IoT System for Weather Monitoring,Case Study on IoT System for Home Automation, Case Study on IoT System for Industry Automation	5.3(T1)
	IoT Physical Devices & Endpoints	**Ch 7**
15	IoT Device	7.1(T1)
16-18	Exemplary Device: Arduino ,About the Arduino Uno input and output Control an LED with Arduino , Interfacing an LED with Switch with Arduino ,Interfacing Relay with Arduino . Analog to Digital Converter ,Reading value from potential meter, DHT-11 temperature sensor, LDR ,Interfacing of various sensors with Arduino	**https://www.arduino.cc/en/Tutorial/Foundations**
19-20	Raspberry Pi, Intel, BeagleBone Black , Cubieboard.	7.2(T1),7.3(T1),7.5(T1),7,7(T1)
	IoT Physical Server and Cloud Offering	**Ch:8(T1),Web References**
21-22	Introduction to Cloud Storage Models & Communication APIs ,Client-Server model for IoT	8.1(T1)
23	Different server side web technologies for IoT-PHP ,JSP ,Servlet ,Node JS	
24	Different Client side web technologies for IoT -HTML	
25-26	MVC architecture for IoT, Web socket and HTTP ,Arduino as a web-client ,Dweet ,Thingspeak,freebord.io .	**http://iot-datamodels.blogspot.in/2013/10/a-modular-open-source-platform-for-web.html**



Course Lecture No.	Learning Topics to be covered	Refer to Chapter, See (Book)
	Case Studies Illustrating IoT Design	**Ch:9(T1)**
27-30	Introduction, Home Automation- Smart Lighting,Home Intrusion Detection , Cities -Smart Parking Environment -Weather Monitoring System ,Weather Reporting Bot ,Air Pollution Monitoring ,Forest Fire Detection ,Agriculture - Smart Irrigation , Productivity Applications - IoT Printer .	9.1-9.6(T1)
	Advanced Topics	**Ch:10,Web References**
31-32	Mobile Application Development using Android and IoT- Introduction ,Basics of Android System,Design mobile app using IOT.	**1. https://www.lynda.com/Android-tutorials/Welcome/184920/365639-4.html** **2. https://dzone.com/articles/build-an-iot-app-using-android-things-in-3-steps**
33-34	Data Analytics and Big data for IoT - Introduction , Database models for IoT, Different types of case study using NoSql, NoSql implementation	10.1(T1),10.2(T1)
35-36	Wireless Technology for IoT - Introduction to different protocols of IoT, Wireless IoT Network Protocols, Different types of case study, Security on IoT	**https://www.link-labs.com/blog/complete-list-iot-network-protocols**

Evaluation Process

1. Assignment: 9 Marks(2 Nos.)
2. Quiz: 9 Marks (2 Nos.)
3. Presentation with Simulation: 5 Marks
4. Prototype: 3 Marks
5. Class Performance: 1 Marks
6. Note Copy:2
7. Attendnance(>=90%): 1
8. MidSem: 20 Marks
9. EndSem: 50 Marks

Smart Refrigerator

- Inventory management
- Voice control
- Recipe suggestions
- Entertainment options
- Energy efficiency

Definition of IoT:

Don't forget to buy milk!

A smart Fridge can do it

- You are leaving the home (sense user)
- There’s no milk in fridge (sense object)
- Use this information to make a decision (process)
- Inform user of decision (communicate)

Definition of IoT:

Physical object (“thing”)

+

Controller (“brain”)

+

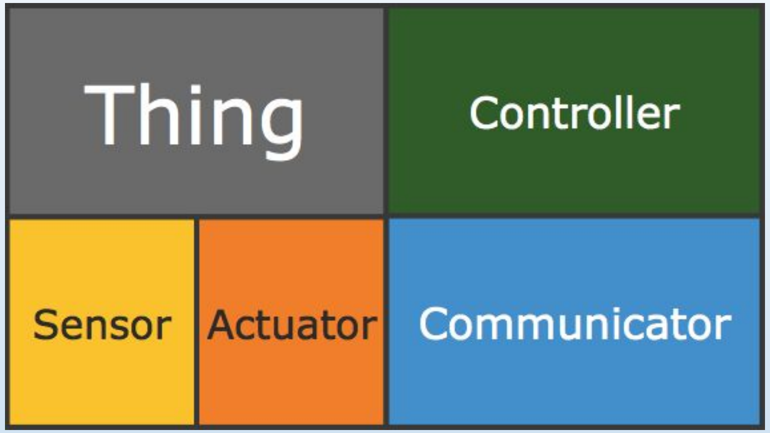
Sensors

+

Actuators

+

Networks (Internet)



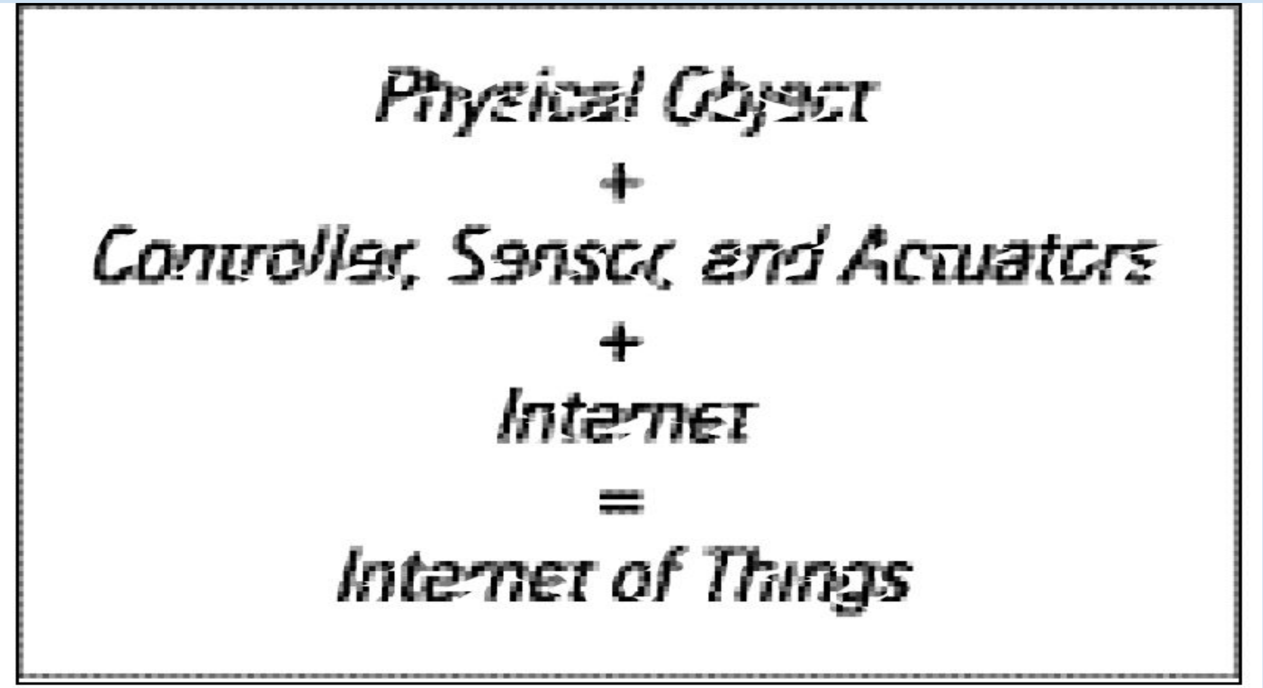
Thing

Controller

Sensor

Actuator

Communicator



Physical Object

+

Controller, Sensor and Actuators

+

Internet

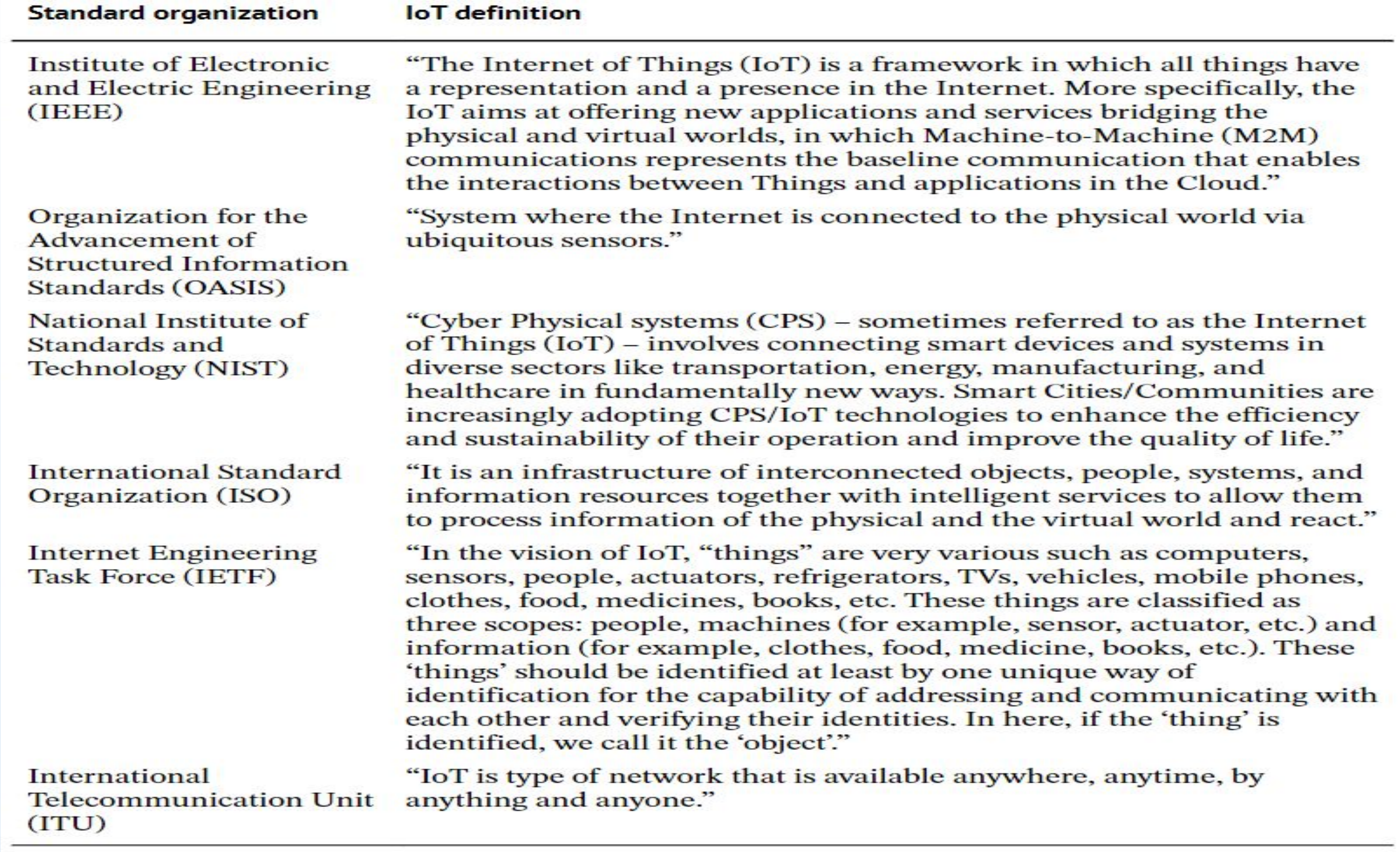
=

Internet of Things

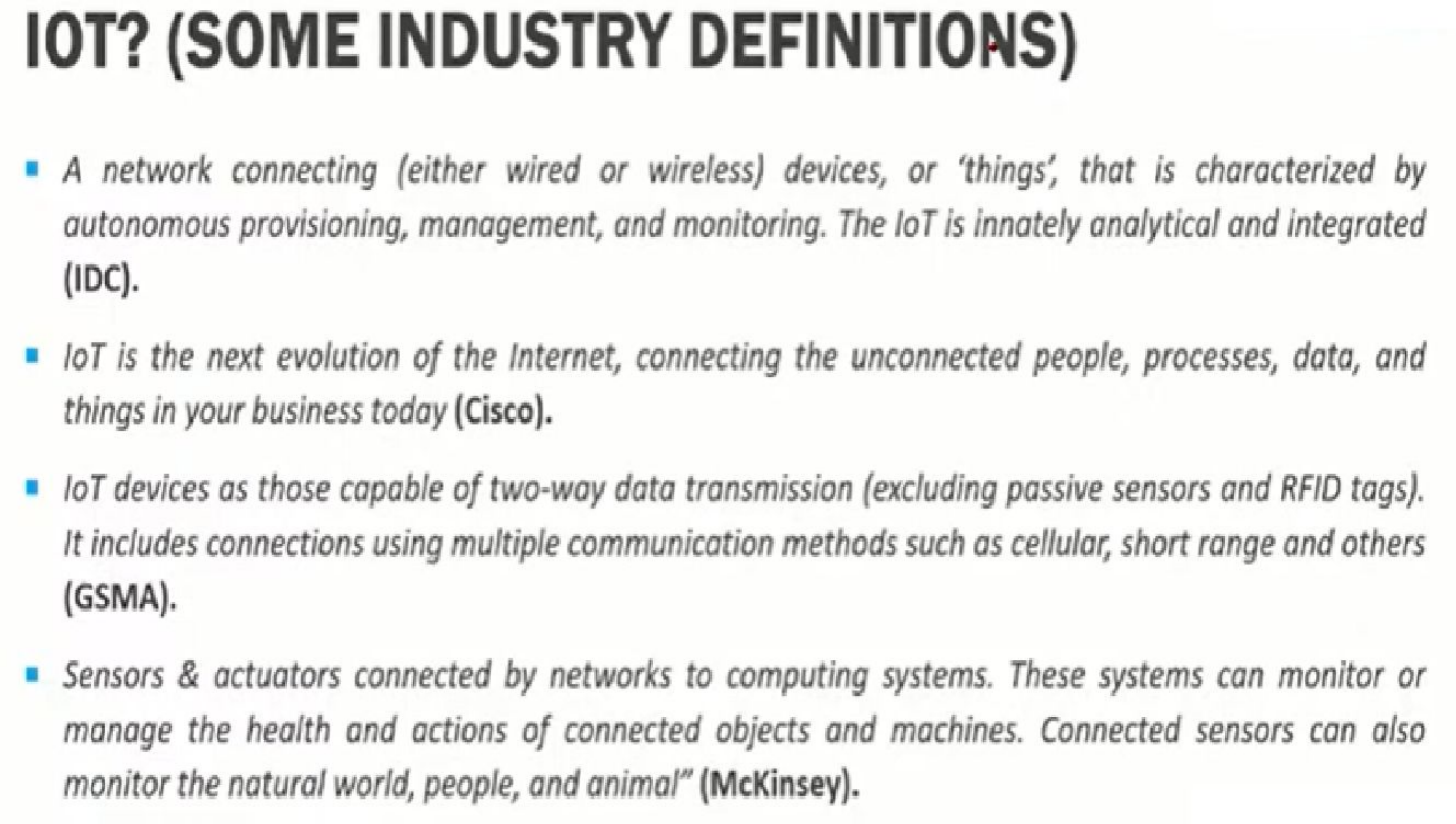
An equation for the Internet of Things.

- The Internet of things (IoT) is the network of physical objects accessed through the Internet.
- IoT has key attributes that distinguish it from the “regular” Internet, as captured by Goldman Sachs’s S-E-N-S-E framework: sensing, efficient, networked, specialized, everywhere
- “Internet of Things (IoT) encapsulates a vision of a world in which objects that have embedded intelligence, communication systems, sensing and actuation capabilities will connect over IP (Internet Protocol) networks”

- Formally IoT was introduced by the International Telecommunication Union (ITU) in the ITU Internet report in 2005).
- **ITU-T Definition**: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.
- **IEEE definition**: A network of items – each embedded with sensors – which are connected to the Internet.
- **European Commission**: The IoT is the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment



Standard organization	IoT definition
Institute of Electronic and Electric Engineering (IEEE)	"The Internet of Things (IoT) is a framework in which all things have a representation and a presence in the Internet. More specifically, the IoT aims at offering new applications and services bridging the physical and virtual worlds, in which Machine-to-Machine (M2M) communications represents the baseline communication that enables the interactions between Things and applications in the Cloud."
Organization for the Advancement of Structured Information Standards (OASIS)	"System where the Internet is connected to the physical world via ubiquitous sensors."
National Institute of Standards and Technology (NIST)	"Cyber Physical systems (CPS) – sometimes referred to as the Internet of Things (IoT) – involves connecting smart devices and systems in diverse sectors like transportation, energy, manufacturing, and healthcare in fundamentally new ways. Smart Cities/Communities are increasingly adopting CPS/IoT technologies to enhance the efficiency and sustainability of their operation and improve the quality of life."
International Standard Organization (ISO)	"It is an infrastructure of interconnected objects, people, systems, and information resources together with intelligent services to allow them to process information of the physical and the virtual world and react."
Internet Engineering Task Force (IETF)	"In the vision of IoT, "things" are very various such as computers, sensors, people, actuators, refrigerators, TVs, vehicles, mobile phones, clothes, food, medicines, books, etc. These things are classified as three scopes: people, machines (for example, sensor, actuator, etc.) and information (for example, clothes, food, medicine, books, etc.). These 'things' should be identified at least by one unique way of identification for the capability of addressing and communicating with each other and verifying their identities. In here, if the 'thing' is identified, we call it the 'object'."
International Telecommunication Unit (ITU)	"IoT is type of network that is available anywhere, anytime, by anything and anyone."



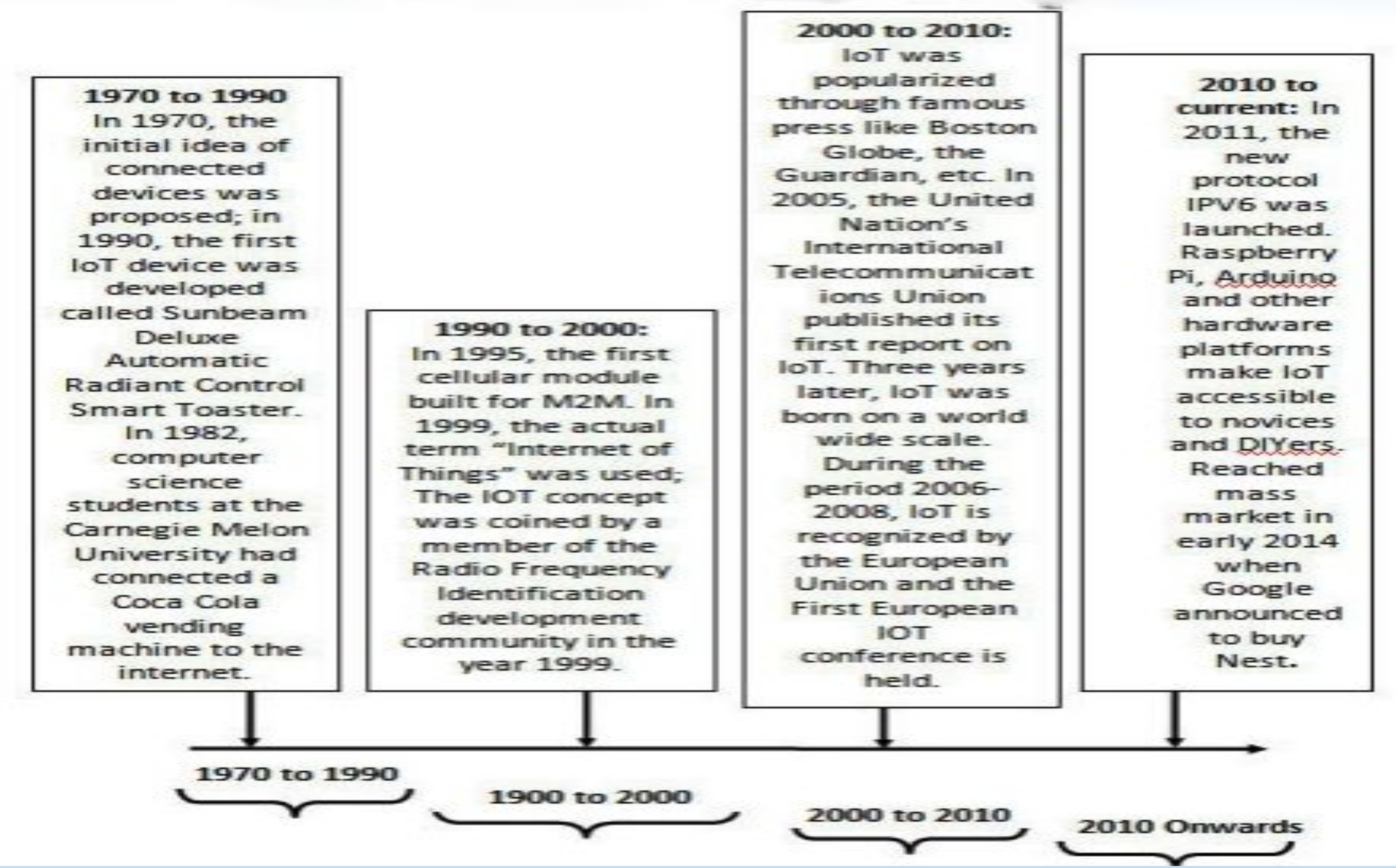
IOT? (SOME INDUSTRY DEFINITIONS)

- *A network connecting (either wired or wireless) devices, or 'things', that is characterized by autonomous provisioning, management, and monitoring. The IoT is innately analytical and integrated* **(IDC).**
- *IoT is the next evolution of the Internet, connecting the unconnected people, processes, data, and things in your business today* **(Cisco).**
- *IoT devices as those capable of two-way data transmission (excluding passive sensors and RFID tags). It includes connections using multiple communication methods such as cellular, short range and others* **(GSMA).**
- *Sensors & actuators connected by networks to computing systems. These systems can monitor or manage the health and actions of connected objects and machines. Connected sensors can also monitor the natural world, people, and animal"* **(McKinsey).**

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.

History of IoT



1970 to 1990
In 1970, the initial idea of connected devices was proposed; in 1990, the first IoT device was developed called Sunbeam Deluxe Automatic Radiant Control Smart Toaster. In 1982, computer science students at the Carnegie Melon University had connected a Coca Cola vending machine to the internet.

1990 to 2000:
In 1995, the first cellular module built for M2M. In 1999, the actual term “Internet of Things” was used; The IOT concept was coined by a member of the Radio Frequency Identification development community in the year 1999.

2000 to 2010:
IoT was popularized through famous press like Boston Globe, the Guardian, etc. In 2005, the United Nation’s International Telecommunications Union published its first report on IoT. Three years later, IoT was born on a world wide scale. During the period 2006-2008, IoT is recognized by the European Union and the First European IOT conference is held.

2010 to current: In 2011, the new protocol IPV6 was launched. Raspberry Pi, Arduino and other hardware platforms make IoT accessible to novices and DIYers. Reached mass market in early 2014 when Google announced to buy Nest.

Scenario #1: IoT in your home

- Imagine you wake up at 7am every day to go to work. Your alarm clock does the job of waking you just fine. That is, until something goes wrong. Your train’s cancelled and you have to drive to work instead. The only problem is that it takes longer to drive, and you would have needed to get up at 6.45am to avoid being late. Oh, and it’s pouring with rain, so you’ll need to drive slower than usual. A connected or IoT-enabled alarm clock would reset itself based on all these factors, to ensure you got to work on time. It could recognize that your usual train is cancelled, calculate the driving distance and travel time for your alternative route to work, check the weather and factor in slower travelling speed because of heavy rain, and calculate when it needs to wake you up so you’re not late. If it’s super-smart, if might even sync with your IoT-enabled coffee maker, to ensure your morning caffeine’s ready to go when you get up.

Scenario #2: IoT in transport

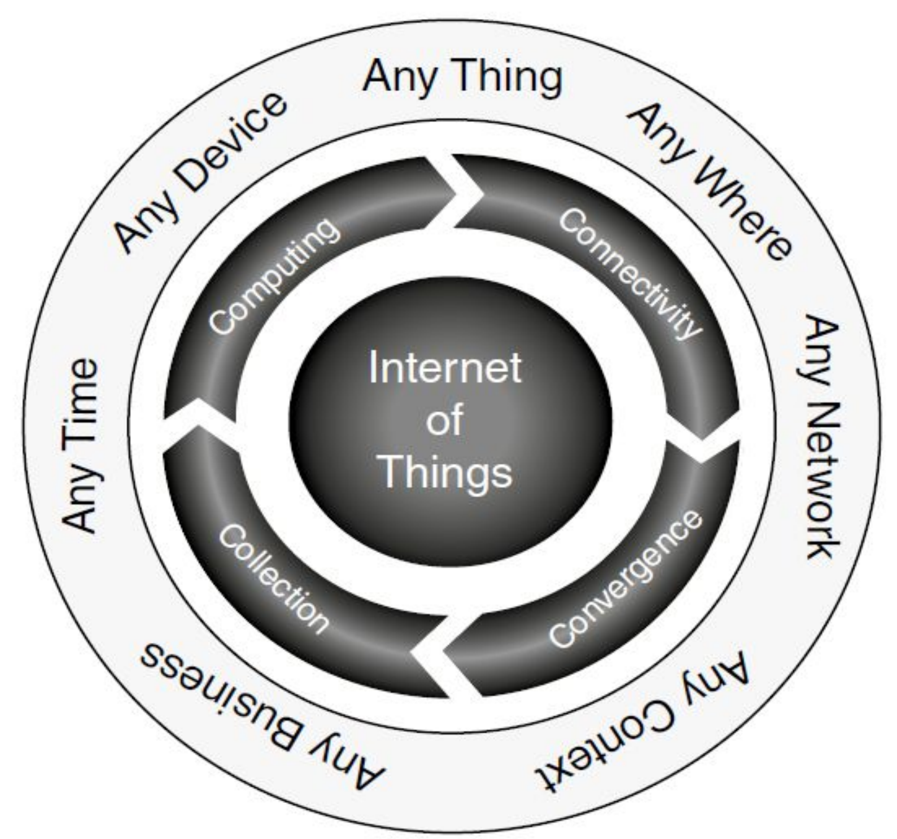
Having been woken by your smart alarm, you're now driving to work. On comes the engine light. You'd rather not head straight to the garage, but what if it's something urgent? In a connected car, the sensor that triggered the check engine light would communicate with others in the car. A component called the diagnostic bus collects data from these sensors and passes it to a gateway in the car, which sends the most relevant information to the manufacturer's platform. The manufacturer can use data from the car to offer you an appointment to get the part fixed, send you directions to the nearest dealer, and make sure the correct replacement part is ordered so it's ready for you when you show up.

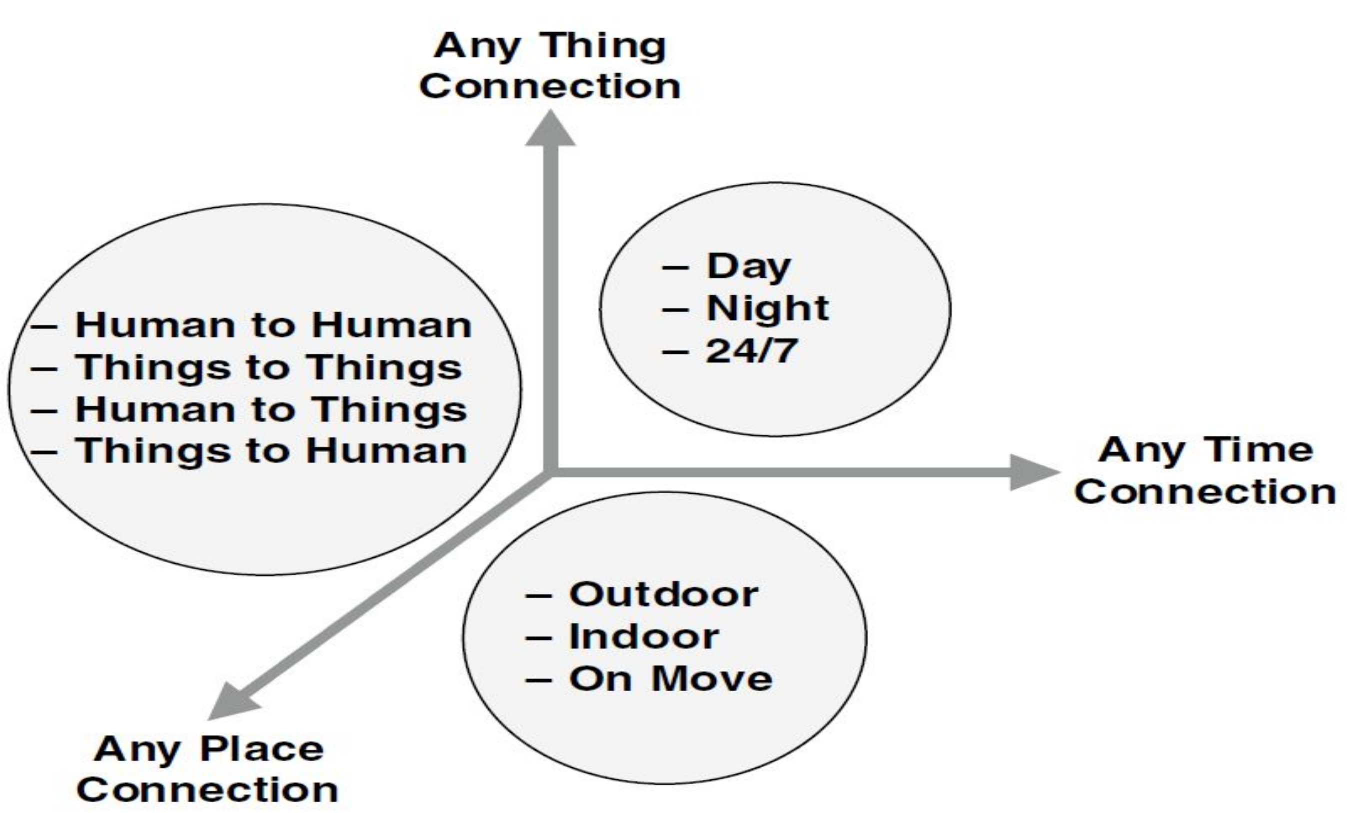
A number of significant technology changes have come together to enable the rise of the IoT.

- **Cheap sensors**: Sensor prices have dropped to an average 60 cents from $1.30 in the past 10 years.
- **Cheap bandwidth**: Thecost of bandwidth hasalso declined precipitously, byafactor of nearly 40 times over the past 10 years.
- **Cheap processing**: Similarly, processing costs have declined by nearly 60 times over the past 10 years, enabling more devices to be not just connected, but smart enough to know what to do with all the new data they are generating or receiving.
- **Smartphones**: Smartphones are now becoming the personal gateway to the IoT, serving as a remote control or hub for the connected home, connected car or the health and fitness devices that consumers have increasingly started to wear.

- **Ubiquitous wireless coverage**: With Wi-Fi coverage now ubiquitous, wireless connectivity is available for free or at a very low cost, given Wi-Fi utilizes an unlicensed spectrum and thus does not require monthly access fees to a carrier.
- **Big data**: AstheIoT will, by definition, generate voluminous amounts of unstructured data, the availability of big data analytics is a key enabler.
- **IPv6**: Most networking equipment now supports IPv6, the newest version of the Internet Protocol (IP) standard that is intended to replace IPv4. IPv4 supports 32-bit addresses, IPv6 can support 128-bit addresses

- The basic idea of IoT can be conceived as a representation of various As and Cs, the As reflect the concept of ubiquity or globalization (i.e. any device, anywhere, anytime, any network etc.) and the Cs mirror the main characteristics of IoT (i.e. connectivity, computing, convergence, etc.). IoT, in essence, can be seen as an addition of the third dimension named "Thing" to the plane of ICT world, which is fundamentally based on two dimensions of Place and Time





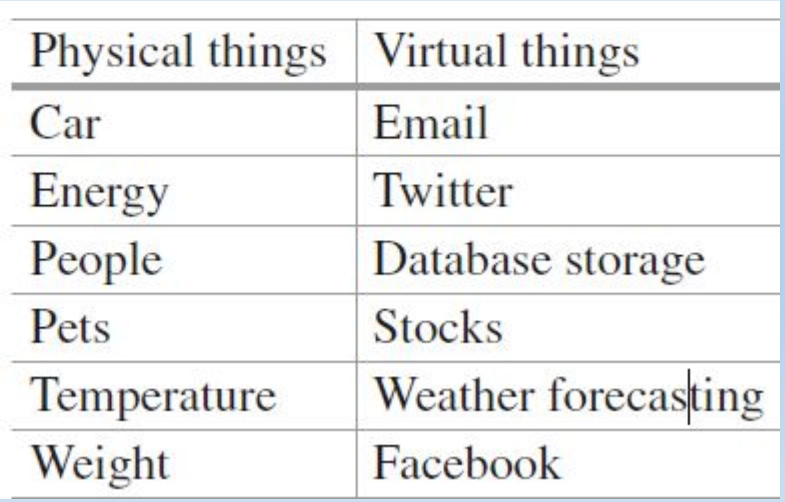
Things: A real/physical or digital/virtual entity that exists and moves in space and time and is capable of being identified. Things are commonly identified either by assigned identification numbers, names and location addresses.

A thing in the internet of things can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low or any other natural or man-made object that can be assigned an IP address and is able to transfer data over a network.

Physical things exist in the physical world and are capable of being sensed, actuated and connected. Examples of physical things include the surrounding environment, industrial robots, goods and electrical equipment.

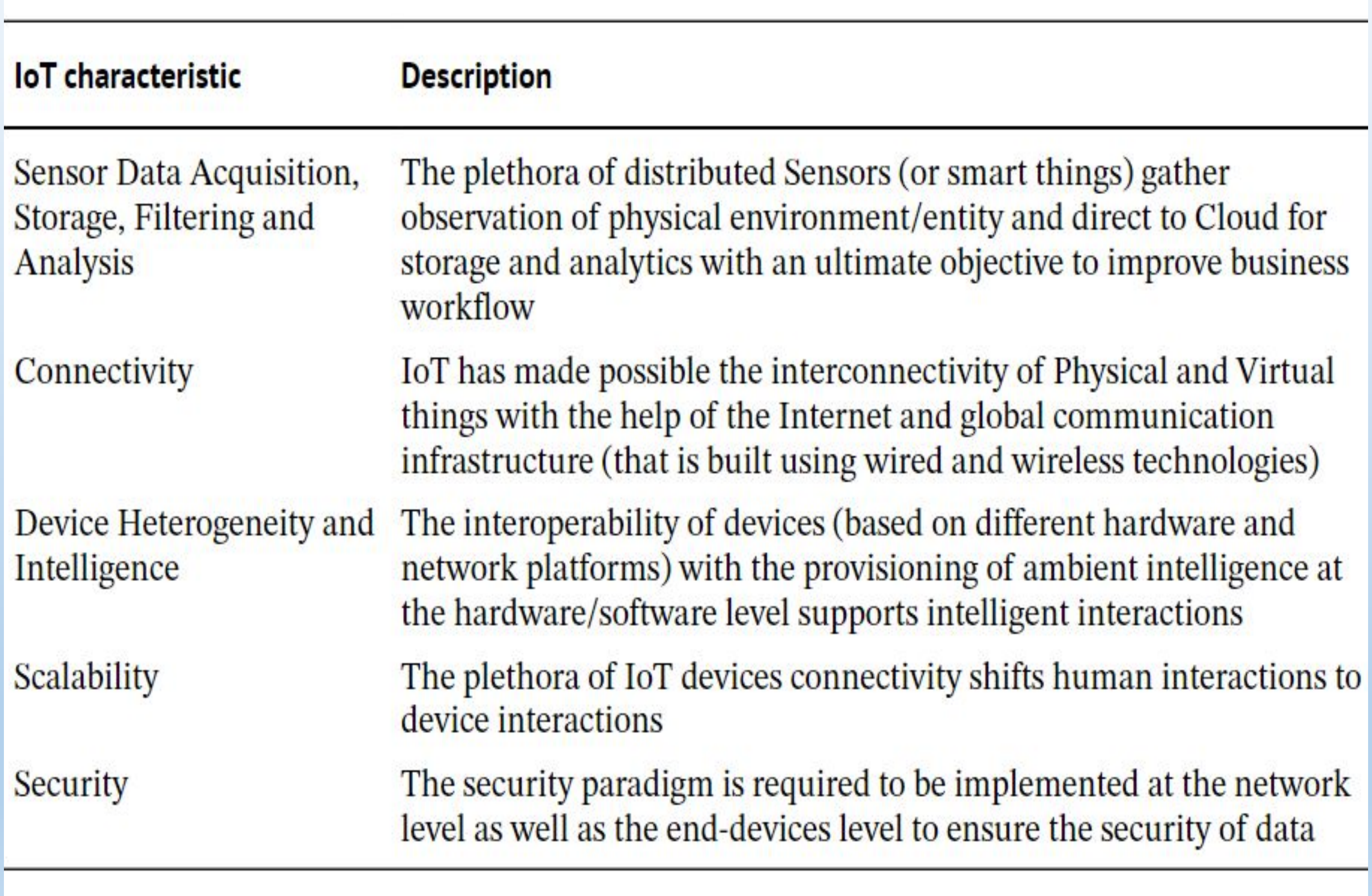
Virtual things exist in the information world and are capable of being stored, processed and accessed. Examples of virtual things include multimedia content and application software

- Those object capable of connecting to the Internet will fall into the "Things" category

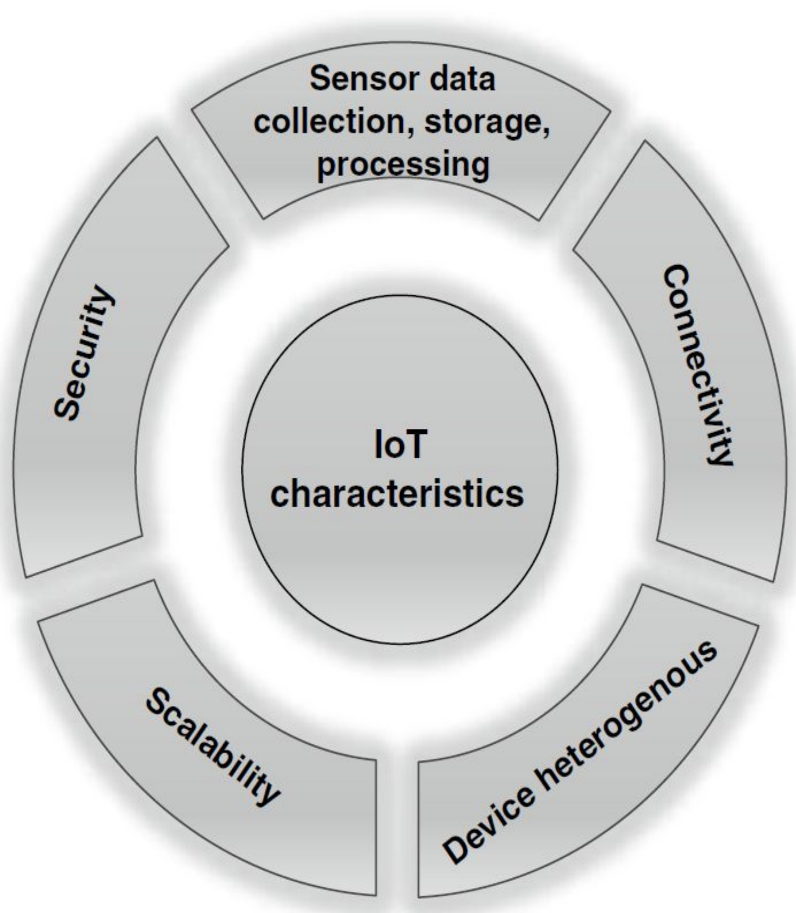


Physical things	Virtual things
Car	Email
Energy	Twitter
People	Database storage
Pets	Stocks
Temperature	Weather forecasting
Weight	Facebook

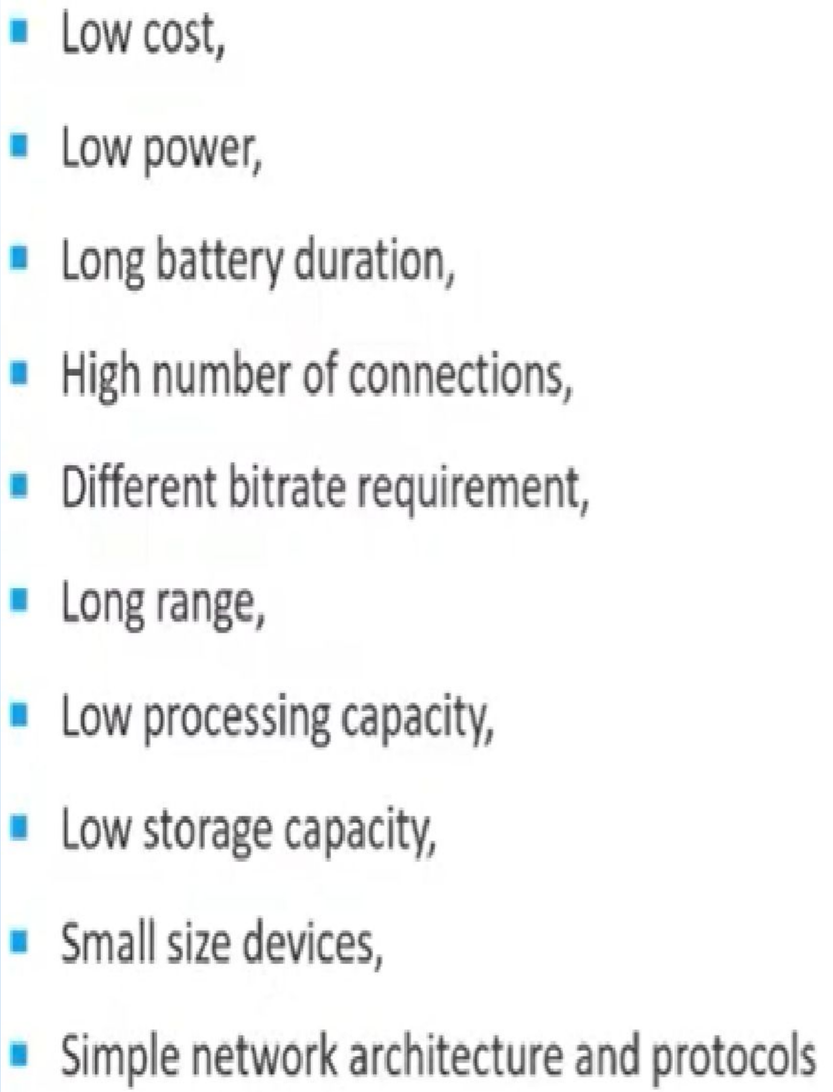
Characteristic of IoT



IoT characteristic	Description
Sensor Data Acquisition, Storage, Filtering and Analysis	The plethora of distributed Sensors (or smart things) gather observation of physical environment/entity and direct to Cloud for storage and analytics with an ultimate objective to improve business workflow
Connectivity	IoT has made possible the interconnectivity of Physical and Virtual things with the help of the Internet and global communication infrastructure (that is built using wired and wireless technologies)
Device Heterogeneity and Intelligence	The interoperability of devices (based on different hardware and network platforms) with the provisioning of ambient intelligence at the hardware/software level supports intelligent interactions
Scalability	The plethora of IoT devices connectivity shifts human interactions to device interactions
Security	The security paradigm is required to be implemented at the network level as well as the end-devices level to ensure the security of data

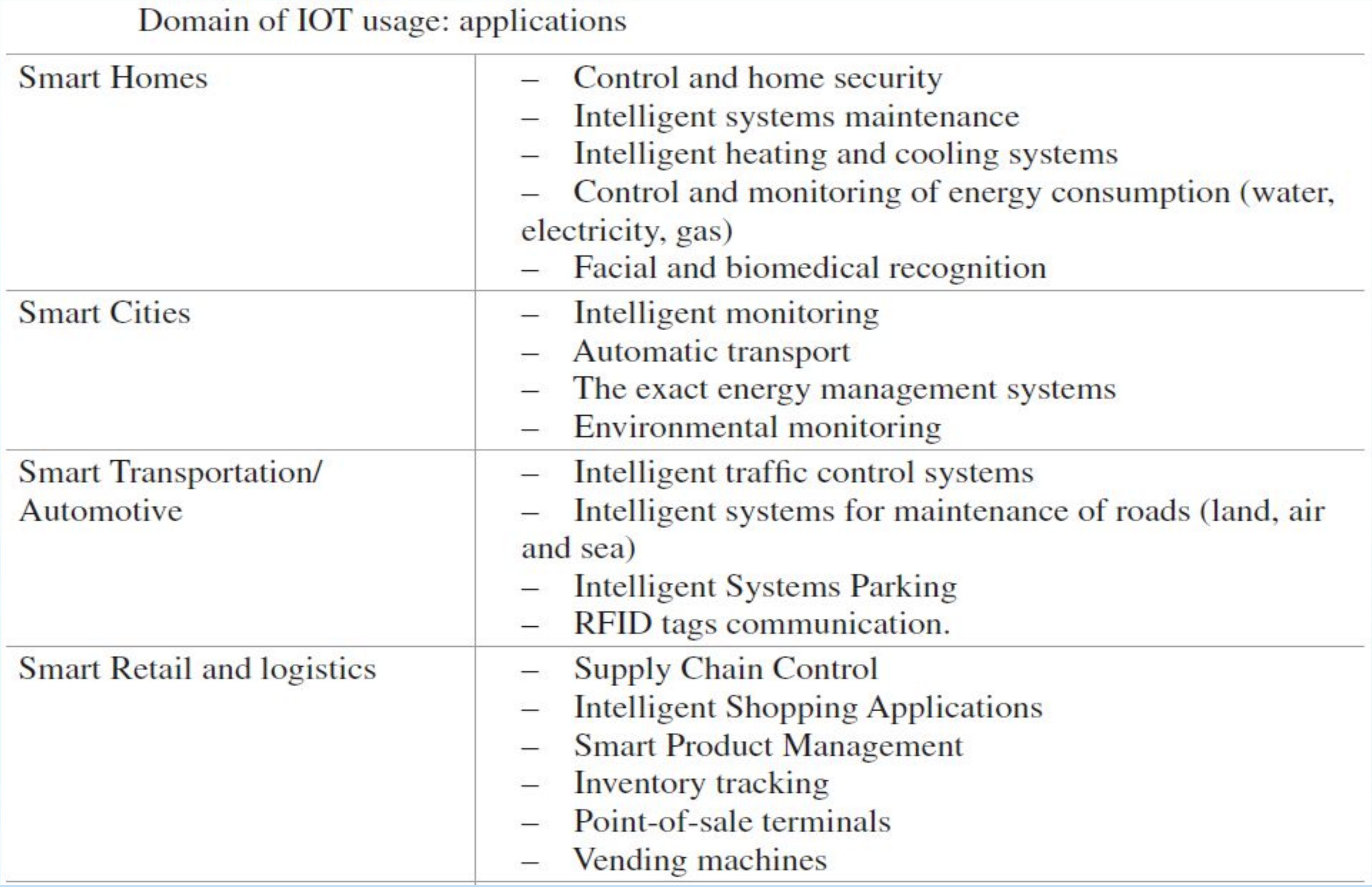


Features of IOT Device

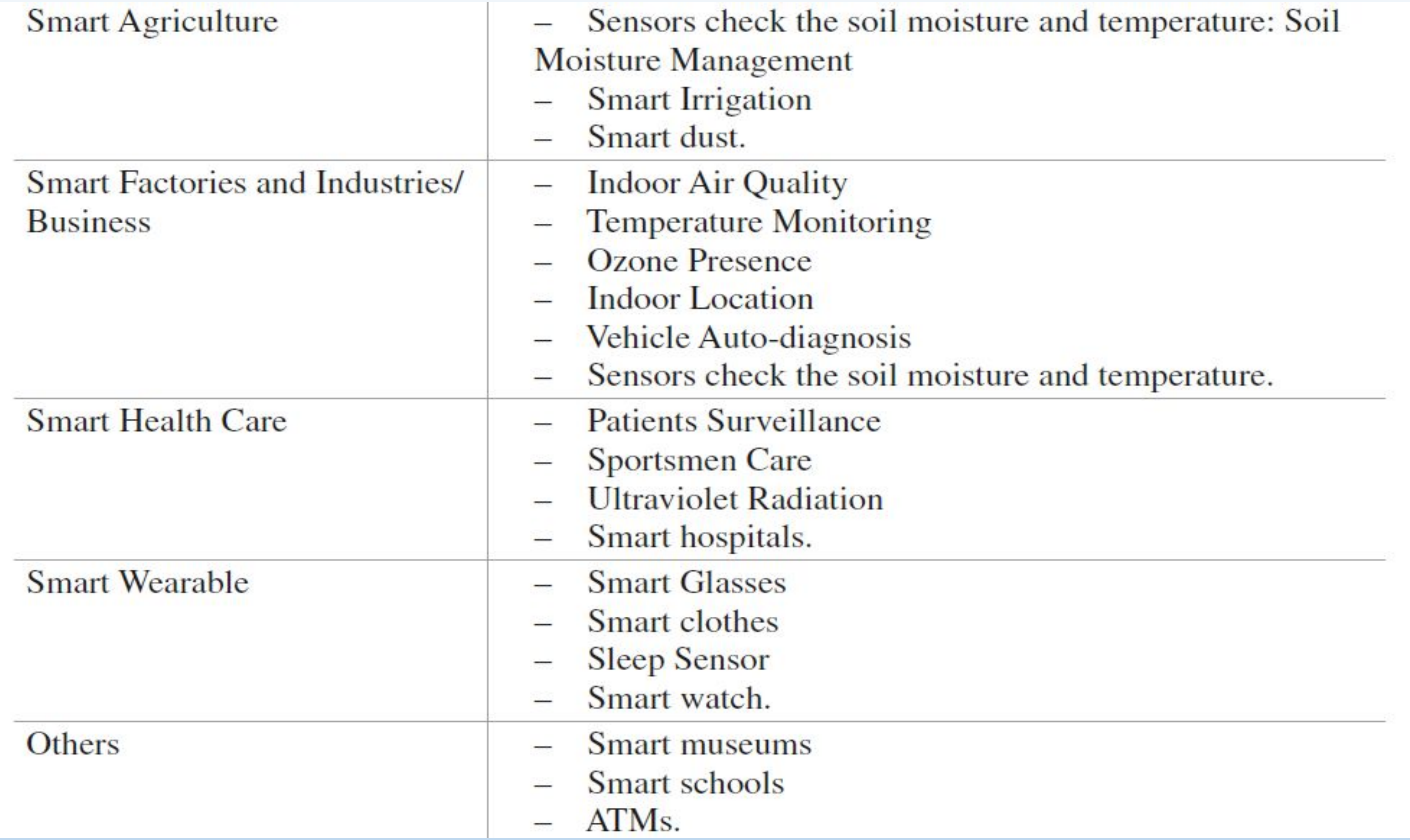


- Low cost,
- Low power,
- Long battery duration,
- High number of connections,
- Different bitrate requirement,
- Long range,
- Low processing capacity,
- Low storage capacity,
- Small size devices,
- Simple network architecture and protocols

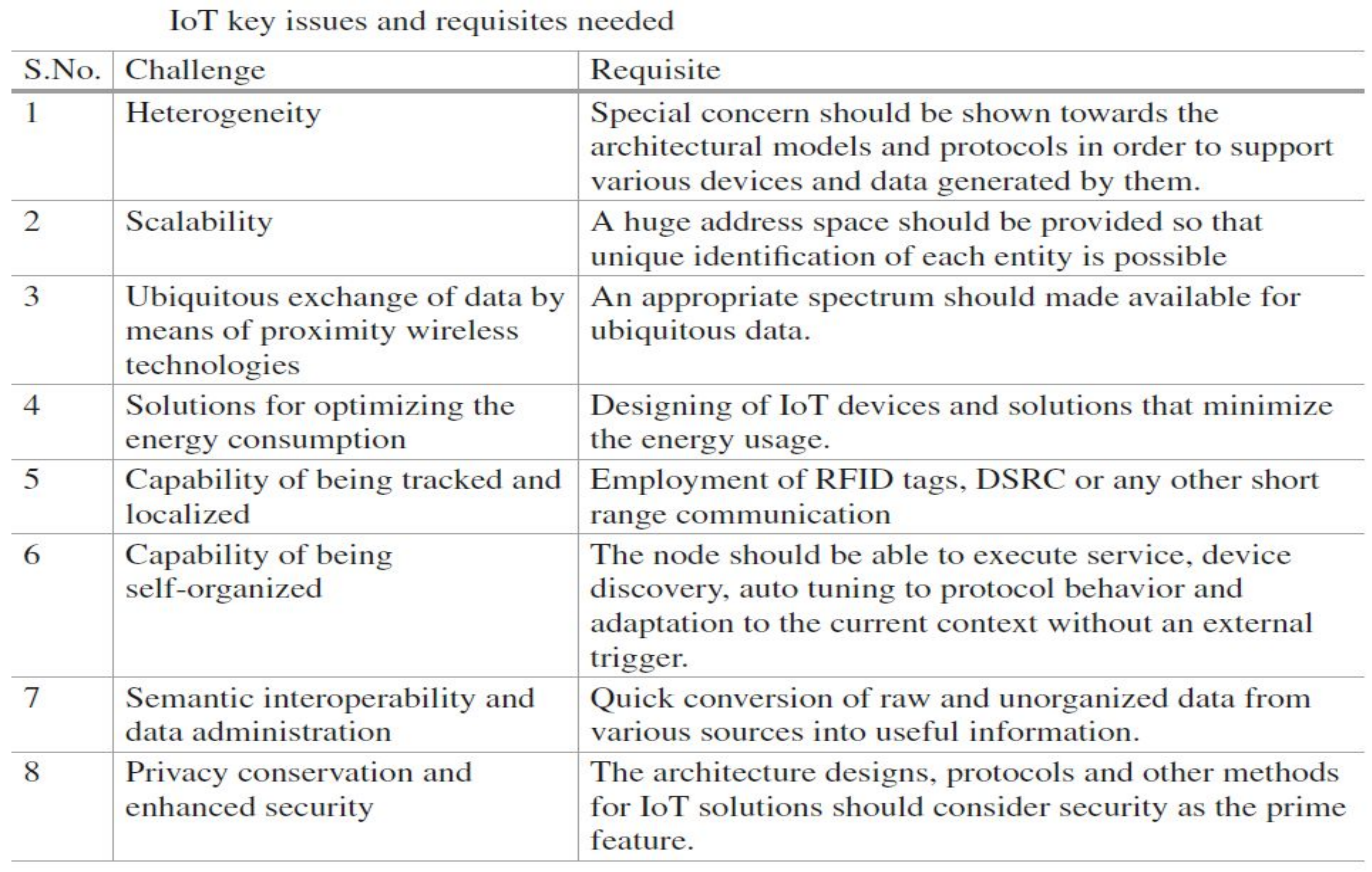
Applications of IoT



Domain of IOT usage: applications	
Smart Homes	– Control and home security – Intelligent systems maintenance – Intelligent heating and cooling systems – Control and monitoring of energy consumption (water, electricity, gas) – Facial and biomedical recognition
Smart Cities	– Intelligent monitoring – Automatic transport – The exact energy management systems – Environmental monitoring
Smart Transportation/ Automotive	– Intelligent traffic control systems – Intelligent systems for maintenance of roads (land, air and sea) – Intelligent Systems Parking – RFID tags communication.
Smart Retail and logistics	– Supply Chain Control – Intelligent Shopping Applications – Smart Product Management – Inventory tracking – Point-of-sale terminals – Vending machines

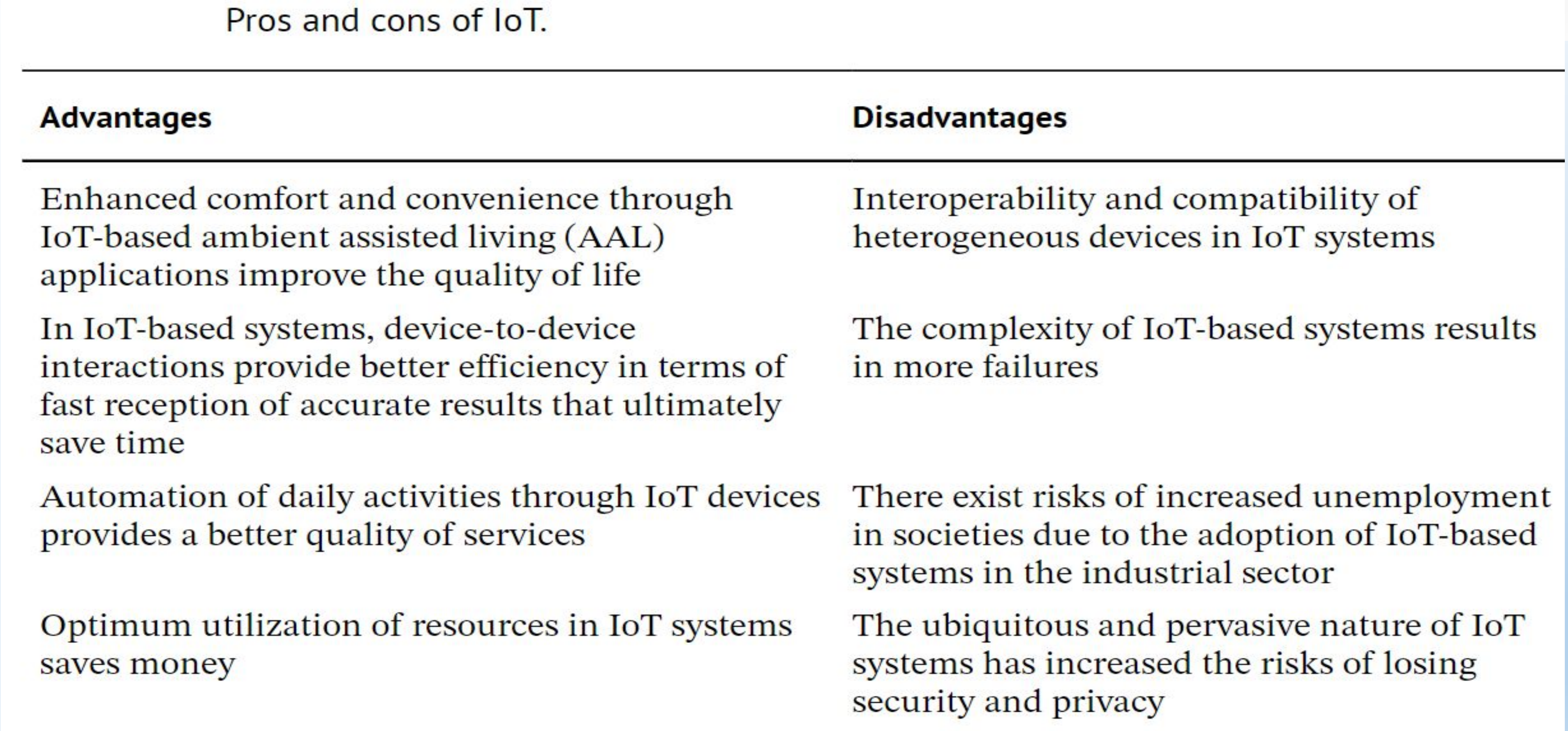


Smart Agriculture	– Sensors check the soil moisture and temperature: Soil Moisture Management – Smart Irrigation – Smart dust.
Smart Factories and Industries/ Business	– Indoor Air Quality – Temperature Monitoring – Ozone Presence – Indoor Location – Vehicle Auto-diagnosis – Sensors check the soil moisture and temperature.
Smart Health Care	– Patients Surveillance – Sportsmen Care – Ultraviolet Radiation – Smart hospitals.
Smart Wearable	– Smart Glasses – Smart clothes – Sleep Sensor – Smart watch.
Others	– Smart museums – Smart schools – ATMs.



IoT key issues and requisites needed

S.No.	Challenge	Requisite
1	Heterogeneity	Special concern should be shown towards the architectural models and protocols in order to support various devices and data generated by them.
2	Scalability	A huge address space should be provided so that unique identification of each entity is possible
3	Ubiquitous exchange of data by means of proximity wireless technologies	An appropriate spectrum should made available for ubiquitous data.
4	Solutions for optimizing the energy consumption	Designing of IoT devices and solutions that minimize the energy usage.
5	Capability of being tracked and localized	Employment of RFID tags, DSRC or any other short range communication
6	Capability of being self-organized	The node should be able to execute service, device discovery, auto tuning to protocol behavior and adaptation to the current context without an external trigger.
7	Semantic interoperability and data administration	Quick conversion of raw and unorganized data from various sources into useful information.
8	Privacy conservation and enhanced security	The architecture designs, protocols and other methods for IoT solutions should consider security as the prime feature.



Pros and cons of IoT.

Advantages	Disadvantages
Enhanced comfort and convenience through IoT-based ambient assisted living (AAL) applications improve the quality of life	Interoperability and compatibility of heterogeneous devices in IoT systems
In IoT-based systems, device-to-device interactions provide better efficiency in terms of fast reception of accurate results that ultimately save time	The complexity of IoT-based systems results in more failures
Automation of daily activities through IoT devices provides a better quality of services	There exist risks of increased unemployment in societies due to the adoption of IoT-based systems in the industrial sector
Optimum utilization of resources in IoT systems saves money	The ubiquitous and pervasive nature of IoT systems has increased the risks of losing security and privacy

SWOT Analysis of IoT

Strength of IoT:

- Portability
- Scalability: real-time connectivity of billions of devices.
- Unlimited functionality.

Weakness of IoT

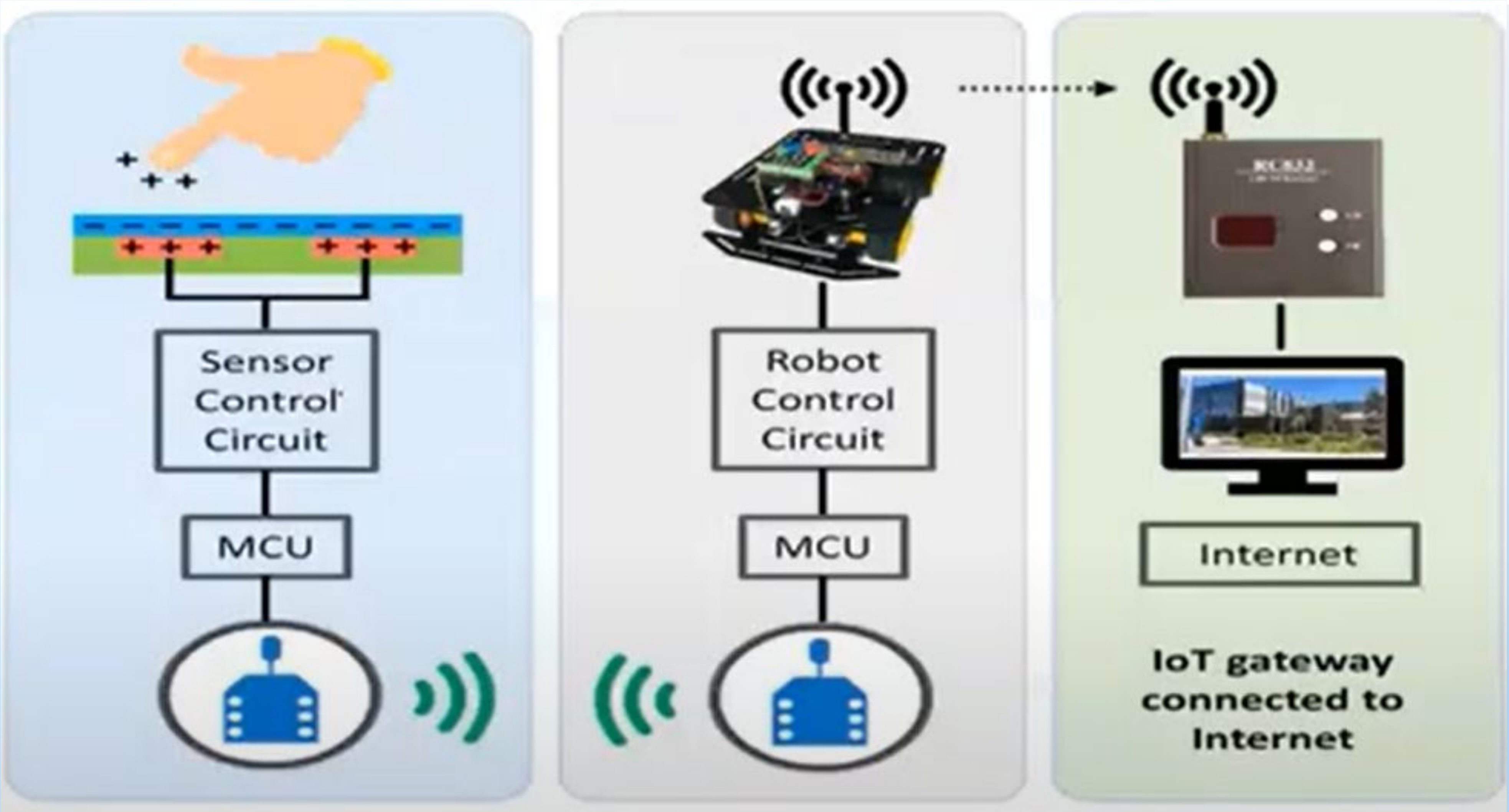
- Dependency on Internet (network outage).
- Dependency on electricity (electricity outage).
- Security.

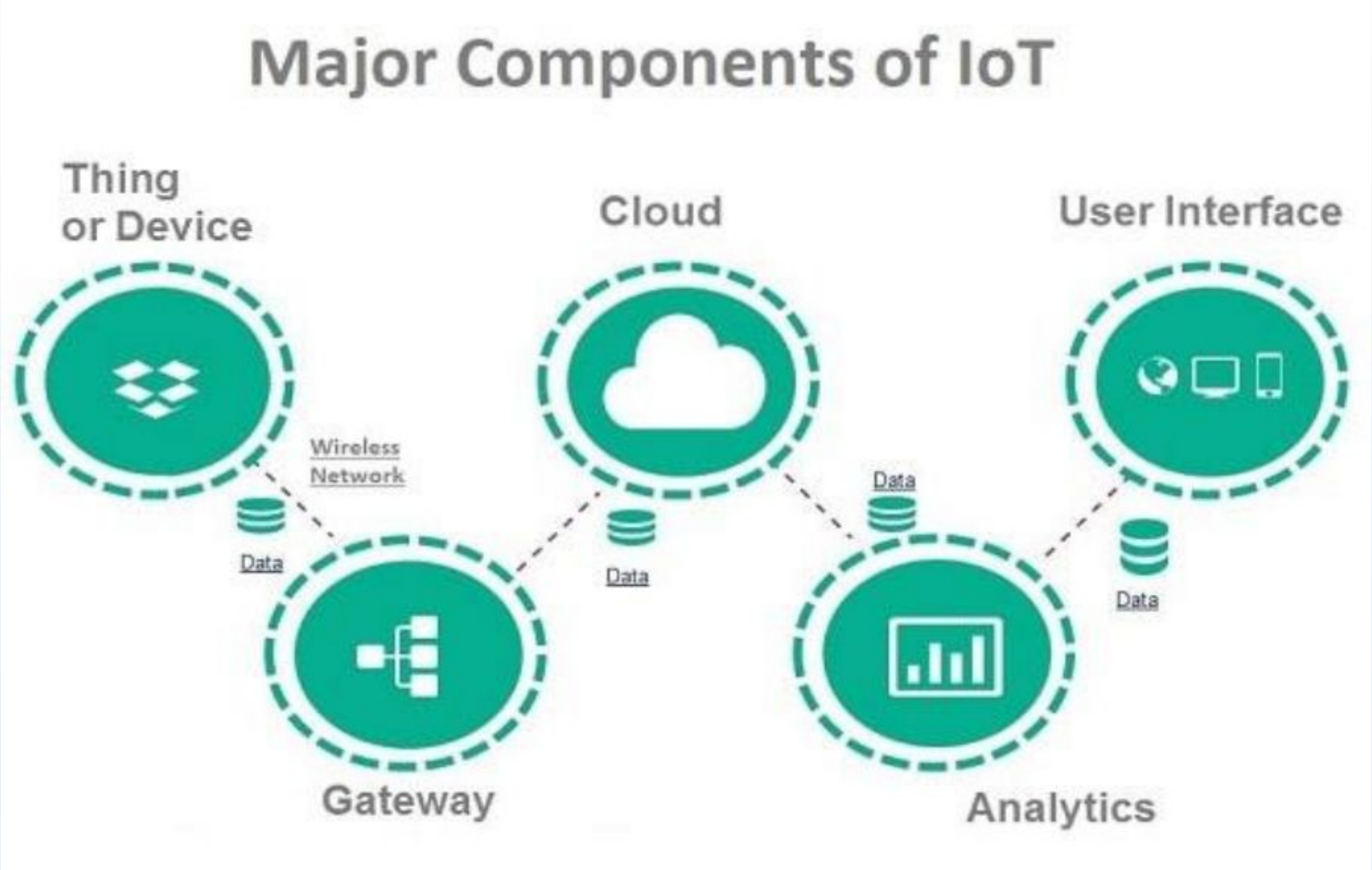
Opportunities of IoT

- A new field for a startup company.
- Business consulting.

Threats of IoT

- Social isolation.
- Dependency on machines may reduce human abilities
- Regulations.


Sensor Control Circuit
MCU
Robot Control Circuit
MCU
Internet
IoT gateway connected to Internet

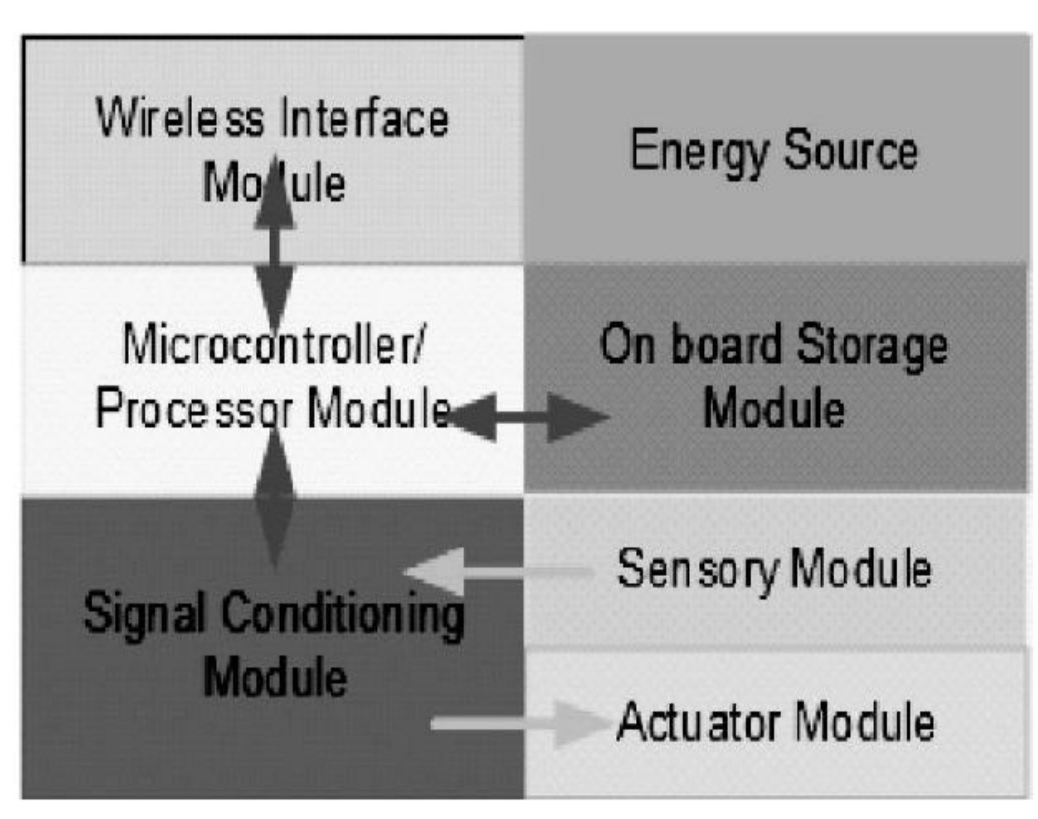


Major Components of IoT

Things:

- These form the front end of the IoT devices. Their main purpose is to collect data from its surroundings (sensors) or give out data to its surrounding (actuators).
- These have to be uniquely identifiable devices with a unique IP address so that they can be easily identifiable over a large network.
- These have to be active in nature which means that they should be able to collect real-time data. These can either work on their own (autonomous in nature) or can be made to work by the user depending on their needs (user-controlled).
- It can perform remote sensing, actuating and monitoring capabilities
- Examples of sensors are gas sensor, water quality sensor, moisture sensor, etc.

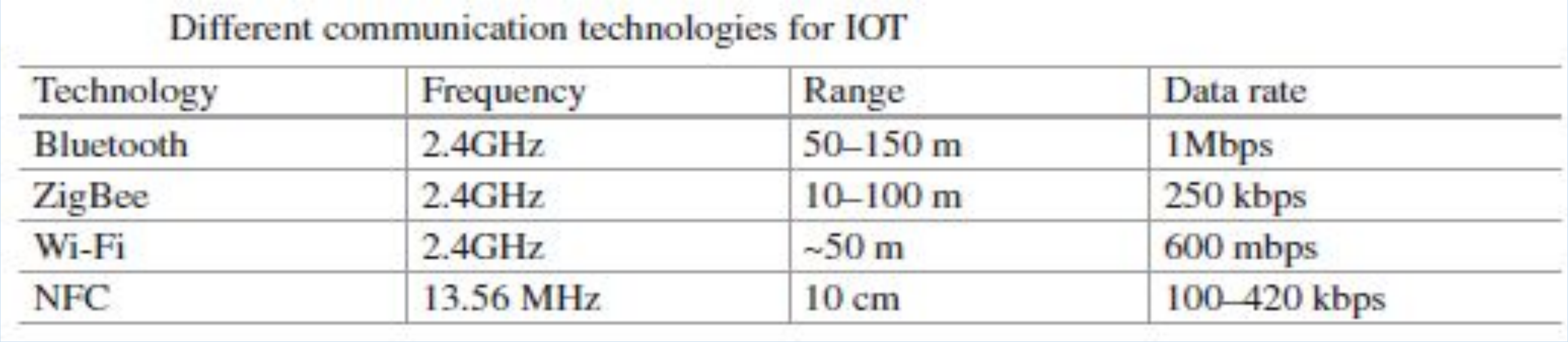
- A typical node consists of seven basic modules
- The number of modules used by a node will depend on the configuration of a node which depends on the application requirements.
- The input module consists of a set of sensors used to gather data from different physical environments.
- The electrical signal output of sensor module are passed on to the signal conditioning module which may contain different submodules such as the amplifier, noise remover and signal conditioning circuits.
- The conditioned signals are then fed to a microcontroller through an A/D (analog to digital) converter.


Wireless Interface Module
Energy Source
Microcontroller/ Processor Module
On board Storage Module
Sensory Module
Signal Conditioning Module
Actuator Module

A Generic Architecture of Things

Gateway:

- An IoT Gateway is a solution for enabling IoT communication, usually **device-to-device communications** or **device-to-cloud communications**. The gateway is typically a hardware device housing application software that performs essential tasks. AIoT Gateway manages the bidirectional data traffic between different networks and protocols.
- The gateway is used to translate different network protocols and make sure interoperability of the connected devices and sensors.
- It perform pre-processing of the collected data from thousands of sensors locally before transmitting it to the next stage.
- It acts as a middle layer between devices and cloud to protect the system from malicious attacks and unauthorized access



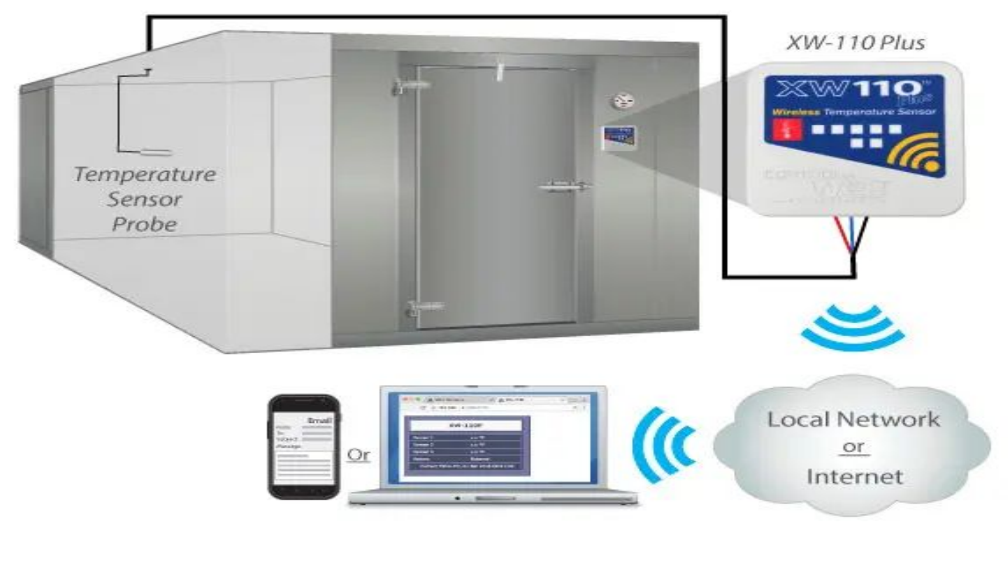
Different communication technologies for IOT

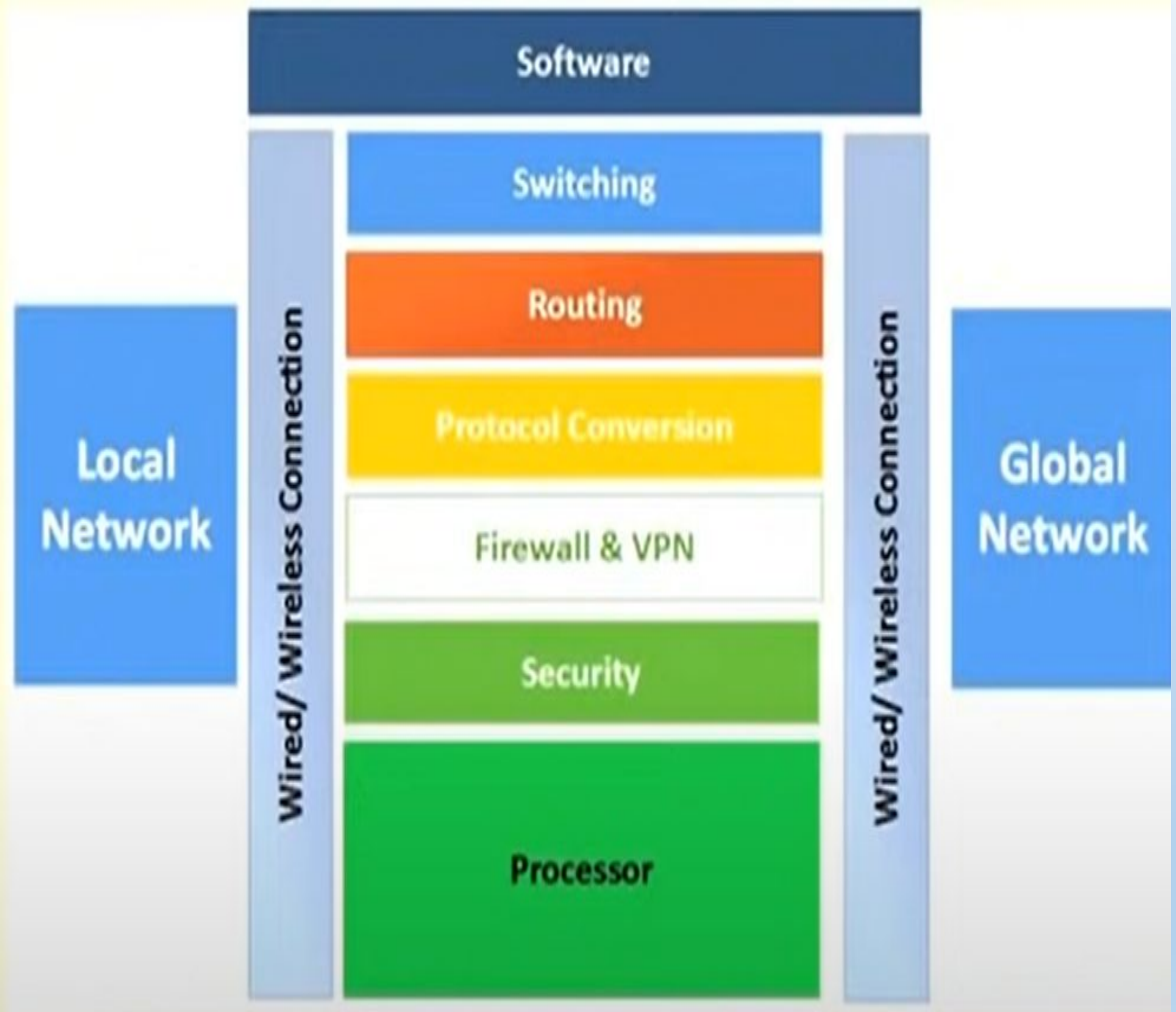
Technology	Frequency	Range	Data rate
Bluetooth	2.4GHz	50–150 m	1Mbps
ZigBee	2.4GHz	10–100 m	250 kbps
Wi-Fi	2.4GHz	~50 m	600 mbps
NFC	13.56 MHz	10 cm	100–420 kbps

IOT Gateway

Functions of Gateway:

1. Communication With The Cloud
2. Route the traffic
3. Support of multiple transfer protocols
4. Isolation of sensor nodes
5. Aggregation and preprocessing of the data
6. Security
7. Local storage of data



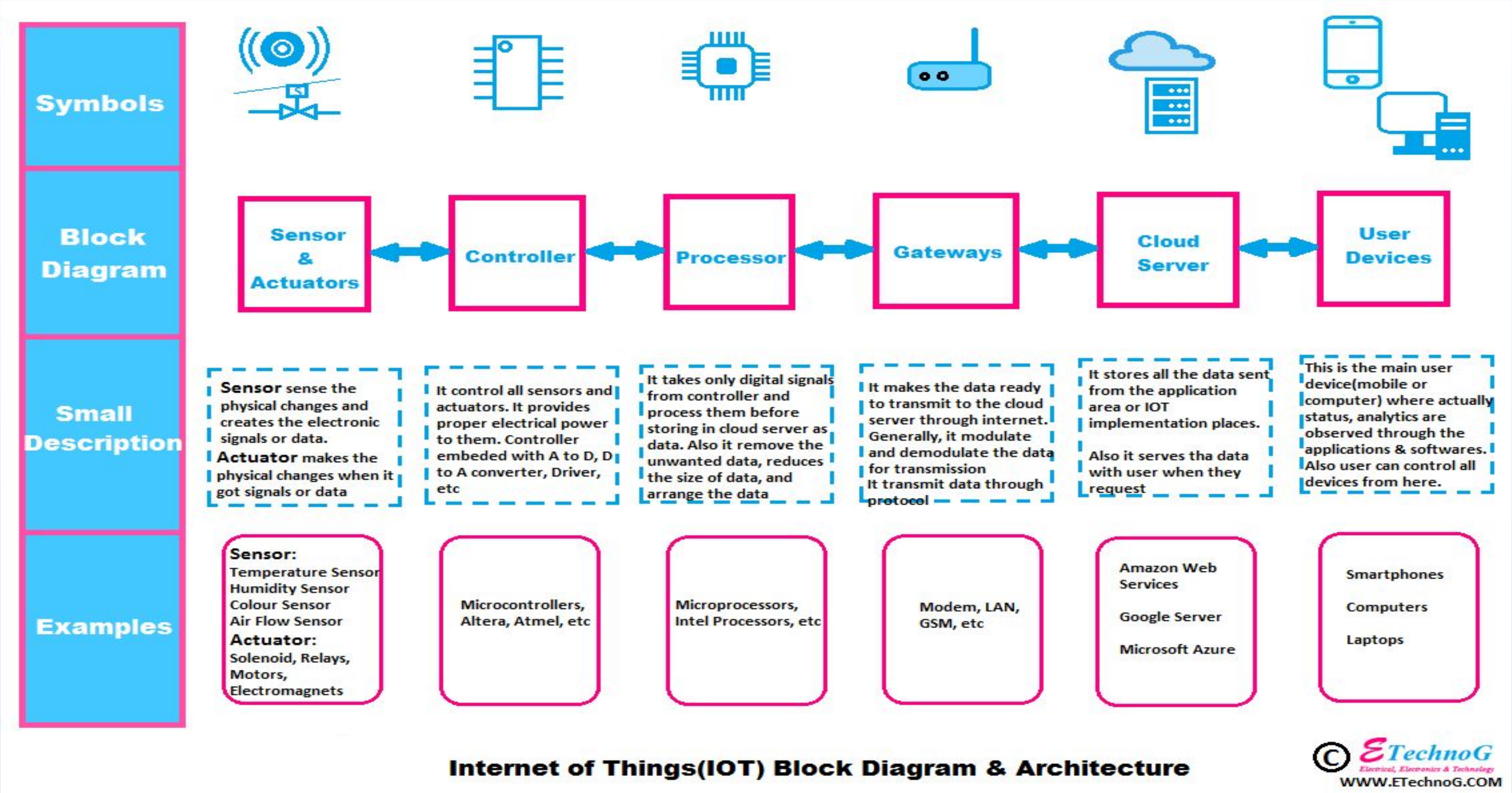


Source: https://sumatosoft.com/blog/what-is-an-internet-of-things-iot-gateway-functions-types-examples

Cloud: It is a sophisticated high performance network of servers optimized to perform high speed data processing of billions of devices, traffic management and deliver accurate analytics. It collect, process, manage and store huge amount of data in real time.

Analytic: Analytics is the process of converting analog data from billions of smart devices and sensors into useful insights which can be interpreted and used for detailed analysis.

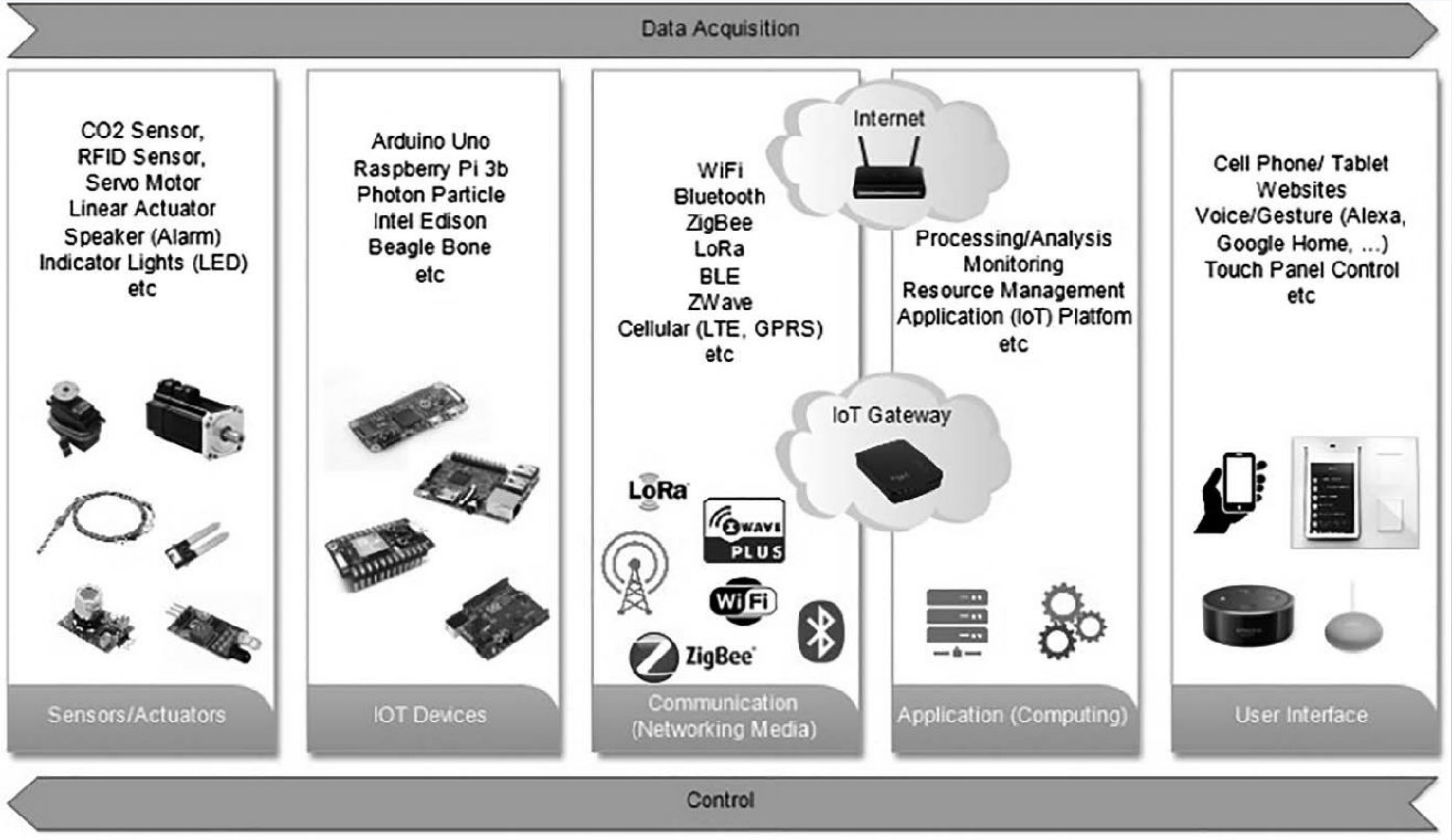
User interface: User interfaces are the visible, tangible part of the IoT system which can be accessible by users. Designers will have to make sure a well designed user interface for minimum effort for users and encourage more interactions.



Symbols						
Block Diagram	Sensor & Actuators	Controller	Processor	Gateways	Cloud Server	User Devices
Small Description	**Sensor** sense the physical changes and creates the electronic signals or data. **Actuator** makes the physical changes when it got signals or data	It control all sensors and actuators. It provides proper electrical power to them. Controller embeded with A to D, D to A converter, Driver, etc	It takes only digital signals from controller and process them before storing in cloud server as data. Also it remove the unwanted data, reduces the size of data, and arrange the data	It makes the data ready to transmit to the cloud server through internet. Generally, it modulate and demodulate the data for transmission It transmit data through protocol	It stores all the data sent from the application area or IOT implementation places. Also it serves tha data with user when they request	This is the main user device(mobile or computer) where actually status, analytics are observed through the applications & softwares. Also user can control all devices from here.
Examples	**Sensor:** Temperature Sensor Humidity Sensor Colour Sensor Air Flow Sensor **Actuator:** Solenoid, Relays, Motors, Electromagnets	Microcontrollers, Altera, Atmel, etc	Microprocessors, Intel Processors, etc	Modem, LAN, GSM, etc	Amazon Web Services Google Server Microsoft Azure	Smartphones Computers Laptops

Internet of Things(IOT) Block Diagram & Architecture

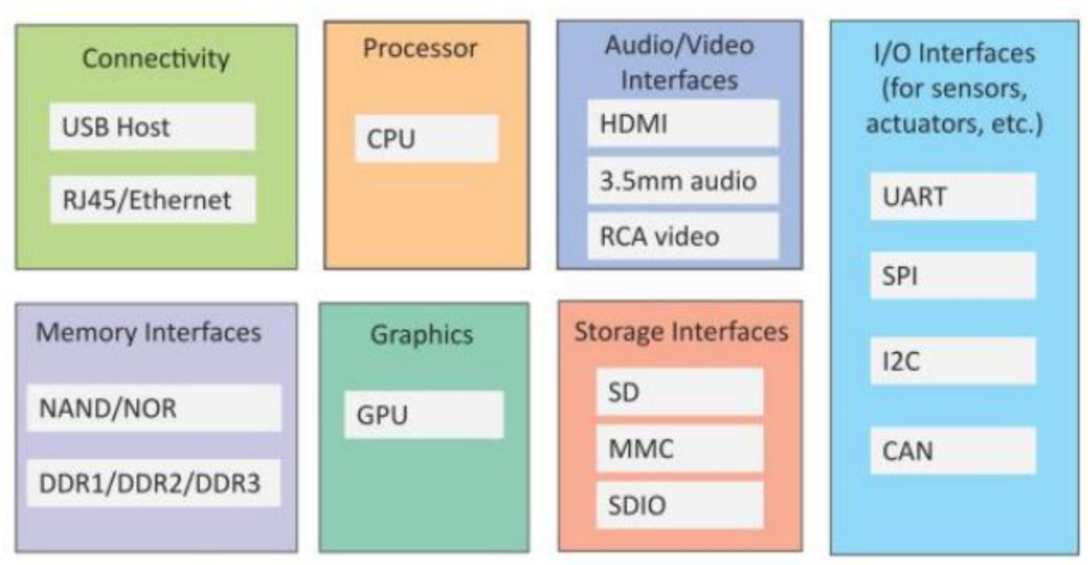
© ETechnoG Electrical, Electronics & Technology
WWW.ETechnoG.COM


Data Acquisition
CO2 Sensor,
RFID Sensor,
Servo Motor
Linear Actuator
Speaker (Alarm)
Indicator Lights (LED)
etc
Arduino Uno
Raspberry Pi 3b
Photon Particle
Intel Edison
Beagle Bone
etc
Internet
WiFi
Bluetooth
ZigBee
LoRa
BLE
ZWave
Cellular (LTE, GPRS)
etc
Processing/Analysis
Monitoring
Resource Management
Application (IoT) Platfom
etc
Cell Phone/ Tablet
Websites
Voice/Gesture (Alexa,
Google Home, ...)
Touch Panel Control
etc
IoT Gateway
LoRa
Z-WAVE PLUS
WiFi
ZigBee
Sensors/Actuators
IOT Devices
Communication
(Networking Media)
Application (Computing)
User Interface
Control

Physical Design of IOT

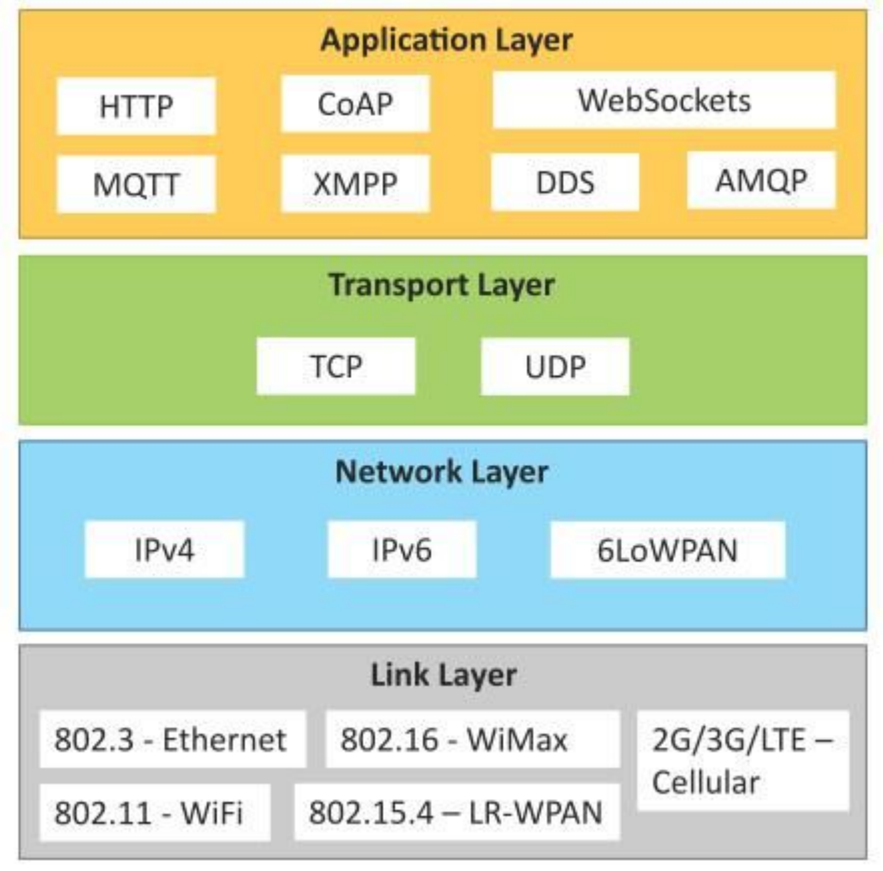
- The "Things" in IoT usually refers to IoT devices which have unique identities and can perform remote sensing, actuating and monitoring capabilities.
- IoT devices can:
- Exchange data with other connected devices and applications (directly or indirectly), or
- Collect data from other devices and process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

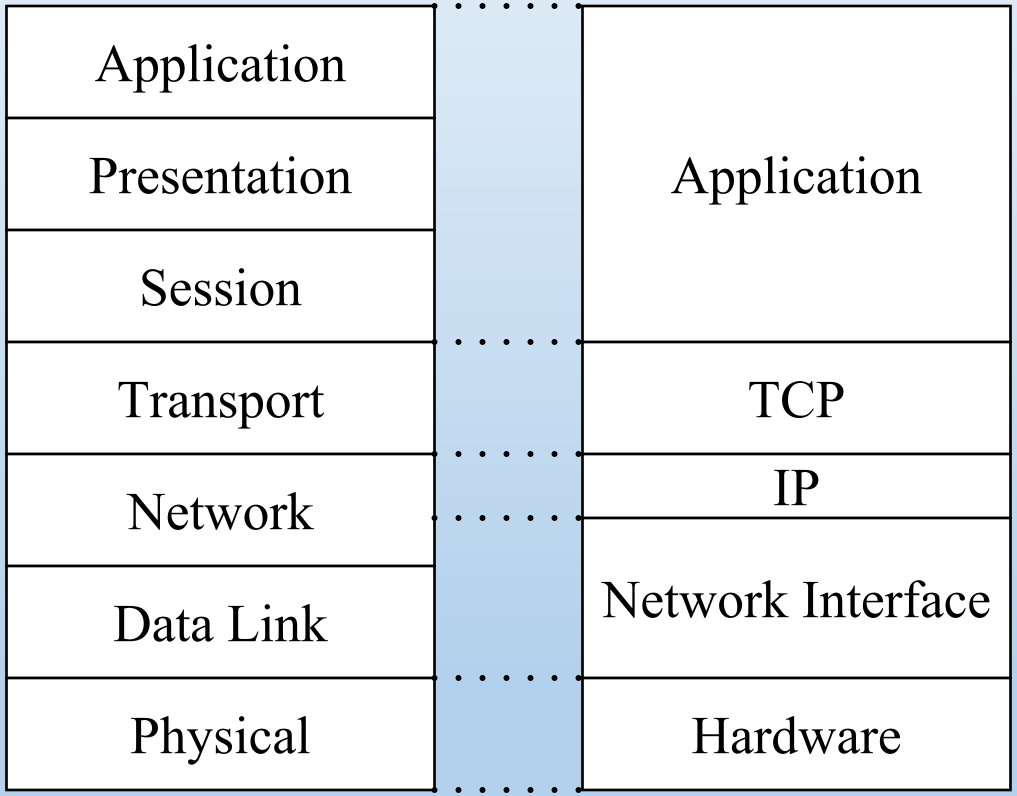
- An IoT device may consist of several interfaces for connections to other devices, both wired and wireless.
- I/O interfaces for sensors
- Interfaces for Internet connectivity
- Memory and storage interfaces
- Audio/video interfaces.



IoT Protocols

- A protocol is a set of rules that governs the communication between two or more devices. A protocol defines the rules, syntax, semantics and synchronization of communication and possible error recovery methods.
- Protocols are used to establish communication between a node device and a server over the internet.
- it helps to send commands to an IoT device and receive data from an IoT device over the internet.
- Protocols are managed by network layers like application, transport, network, and link layer.





Application

Presentation

Session

Transport

Network

Data Link

Physical

Application

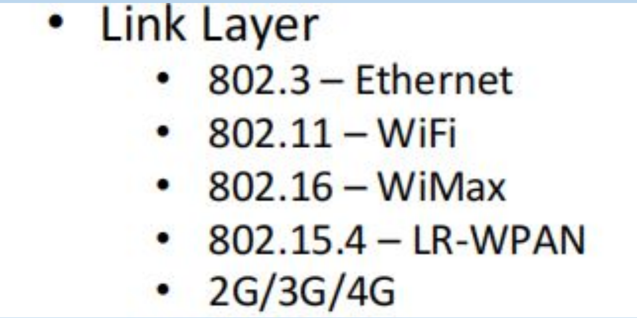
TCP

IP

Network Interface

Hardware

- **Link Layer**:
- Responsible for establishing and terminating links between the nodes
- Link-layer protocols are used to send data over the network's physical layer.
- It also determines how the packets are coded and signaled by the devices.
- Link Layer Protocols are:



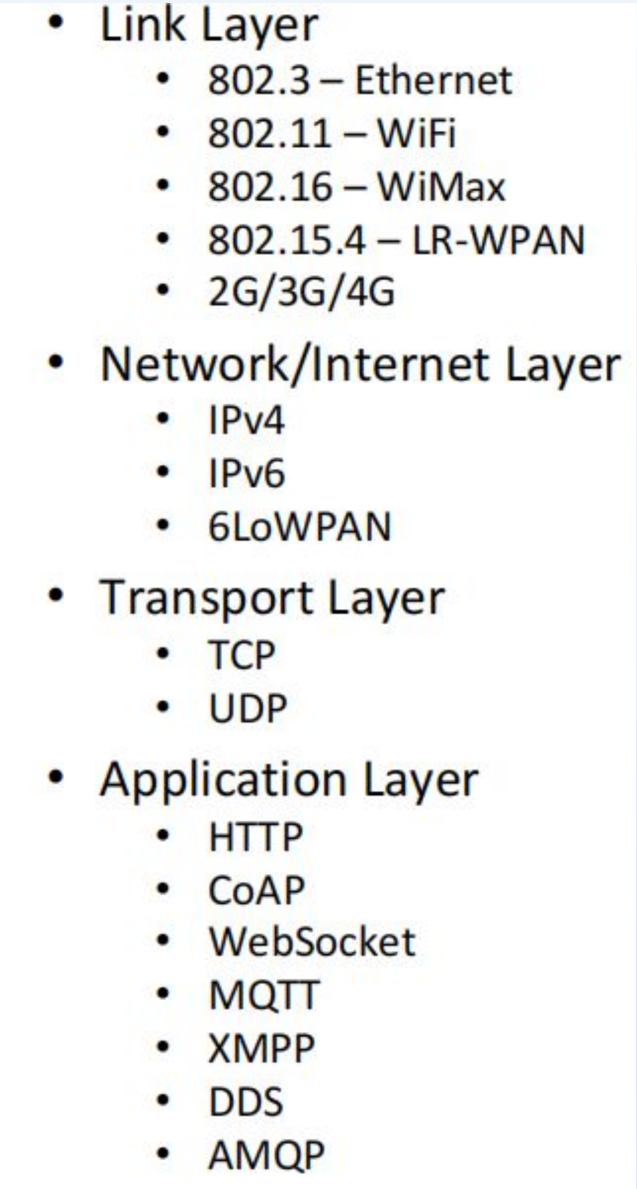
- Link Layer
  - 802.3 – Ethernet
  - 802.11 – WiFi
  - 802.16 – WiMax
  - 802.15.4 – LR-WPAN
  - 2G/3G/4G

- **Network Layer**:
- Transfer the data packet from sender to receiving host.
- Handles routing, which involves selecting the next node and forwarding the packets across the communication path
- Responsible for logical addressing (like IP address) and for congestion control which prevents the network from being overloaded with traffic.
- Network Layer Protocols are:
- IPv4 (32-bit addresses)
- IPv6 (128-bit addresses)
- 6LoWPAN (IPv6 over Low power Wireless Personal Area Network)

- **Transport Layer:**
- The transport layer is responsible for the reliable delivery of the message across the end node, flow control and multiplexing and demultiplexing of the channels at end nodes
- used to control the flow of data segments and handle error control.
- provides a logical communication channel through which the end applications can communicate with each other.
- In this layer message transfer after the set up on connections as either using handshakes (as in TCP) or without handshake/acknowledgements (as in UDP).
- Different protocols at transport layer are:

- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)

- **Application Layer :**
- The users of an IoT application interact with the IoT application/system.
- It allows the users to interact with the IoT sensors and access other services provided by the communication network.
- It provides services like authentication, naming, message formatting, email, etc, to the users.
- HTTP (HyperText Transfer Protocol)
- CoAP (Constrained Application Protocol)
- MQTT (Message Queue Telemetry Transport)
- XMPP (Extensible Messaging and Presence Protocol)
- AMQP (Advanced Message Queuing Protocol)
- WebSocket
- DDS (Data Distribution Service)



- 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

802.3-Ethernet:

- It it a collection of wired ethernet standards for the link layer
- These standards provide data rates from 10 Mb/s to 40 Gb/s and higher.
- o IEEE 802.3 (10BASE5 Ethernet):- coaxial cable.
- o IEEE 802.3.i (10BASE-T Ethernet):- copper twisted-pair connections.
- o IEEE 802.3.j (10BASE-F Ethernet):- fibre optic connections.
- o IEEE 802.3ae (10 Gbit/s Ethernet):- fibre, and so on.

802.11-WiFi:

- wireless local area network (WLAN) communication (operates in the 2.4/5 GHz band to 60 GHz bands (data rates 1 Mb/s to upto 6.75 Gb/s))
- 802.11a: 5GHz band
- 802.11b and 802.11g: 2.4GHz band
- 802.11n: 2.4/5 GHz band
- 802.11ac: 5GHz band
- 802.11ad: 60GHz band

802.16-WiMax:

WiMax standards provide data rates from 1.5 Mb/s to 1 Gb/s ((802.16m) provides data rates of 100 Mbit/s for mobile stations and I Gbit/s for fixed stations).

802.15.4-LR-WPAN:

- (low-rate wireless personal area networks) use for high level communication protocols provide data rates from 40 Kb/s 250 Kb/s such as ZigBee. Provide low-cost and low-speed communication for power constrained devices.

2G/3G/4G - Mobile Communication:

- 2G including GSM and CDMA, 3G-including UMTS and CDMA2000 and 4G-including LTE.
- IoT devices based on these standards can communicate over cellular networks.

IPv4: (Internet Protocol version 4)

- This Internet protocol is used to identify the devices on a network by a hierarchical addressing scheme.
- IPv4 uses a 32-bit address scheme (total of 2^{32} or 4,294,967,296 addresses).
- In this protocol IP protocols establish connections on packet networks, without guarantee of packets delivery.
- Guaranteed delivery and data integrity are handled by the upper layer protocols (such TCP).

IPv6 : (Internet Protocol version 6)

- It is the newest version of Internet protocol and successor of IPv4.
- It is use 128-bit address scheme(total 2^{128} or $3.4*10^{38}$ addresses).

6LoWPAN (IPv6 over Low power Wireless Personal Area Networks)

- It is IP protocol to the low-power devices (limited processing capability). Operates in the 2.4 ghz frequency range and provides data transfer rates 250 Kb/s.

TCP: (Transmission Control Protocol):

- It is used by web browsers, email programs (SMTP application layer protocol) and file transfer (FTP).
- TCP is a connection oriented and stateful protocol, ensures reliable transmission of packets and also, provides error detection capability so that duplicate packets can be discarded and lost packets are retransmitted.
- The flow control capability ensures that rate at the sender is not too high for the receiver to process.
- The congestion control capability of TCP helps in avoiding network congestion and congestion collapse which can lead to degradation of network performance.

UDP (User Datagram Protocol):-

- UDP is connectionless, transaction oriented and stateless protocol.
- useful for time-sensitive applications (for very small data units to exchange and do not want the overhead of connection setup).
- Not provide guaranteed to delivery, ordering of messages and duplicate elimination.

- **HTTP: (Hypertext Transfer Protocol)**
- Application layer protocol that forms the foundation of the World Wide Web (WWW).
- Includes commands such as GET, PUT, POST DELETE, HEAD, TRACE, OPTIONS, etc.
- It follows a request-response model where a client sends requests to a server using the HTTP commands.
- It is a stateless protocol and each HTTP request is independent to other requests.
- An HTTP client can be a browser or an application running on the client (e.g., an application running on an loT device a mobile application.pr other software).
- HTTP protocol uses Universal Resource Identifiers (URIS) to identify HTTP resources HTTP.

COAP: (Constrained Application Protocol)

- It is an application layer protocol for machine-to-machine (M2M) applications.
- It is a web transfer protocol and uses a request-response model.
- it runs on top of UDP instead of TCP.
- It uses a client-server architecture where clients communicate with servers using connectionless datagrams.
- It is designed to easy interface with HTTP.
- Like HTTP, COAP supports methods such as GET, PUT, POST, and DELETE.

WebSocket:

- WebSocket protocol allows full-duplex communication over a single socket connection for sending messages between client and server.
- It is based on TCP and allows streams of messages to be sent back and forth between the client and server while keeping the TCP connection open.
- The client can be a browser a mobile application or an IoT device.

MQTT : (Message Queue Telemetry Transport)

- It is a light weight messaging protocol based on the publish-subscribe model.
- It uses a client-server architecture where the client (such as an IoT device) connects to the server (also called MQTT Broker) and publishes messages to topics on the server.
- The broker forwards the messages to the clients subscribed to topics MOTT is well suited for constrained environments where the devices have limited processing and memory resources and the network bandwidth is low.

XMPP : (Extensible Messaging and Presence Protocol)

- It is a protocol for real-time communication and streaming XML data between network entities in IOT devices.
- It is use in wide range of applications including messaging, presence, data syndication, gaming, multi-party chat and voice/video calls.
- It allows sending small chunks" of XML data from one network entity to another in near real-time.
- It is a decentralized protocol and uses a client-server architecture.
- supports both client-to-server and server-to-server communication paths.

Data Distribution Service (DDS):-

- It is a data-centric middleware standard for device-to-device or machine-to-machine communication.
- It uses a publish-subscribe model where publishers (e.g. devices that generate data) create topics to which subscribers (e.g.. devices that want to consume data) can subscribe.
- Publisher is an object responsible for data distribution and the subscriber is responsible for receiving published data.
- DDS provides quality-of-service (QoS) control and configurable reliability.

- **Advanced Message Queuing Protocol (AMQP):-**
- It is an open application layer protocol for business messaging.
- it supports both point-to-point and publisher/subscriber models, routing and queuing.
- It’s brokers receive messages from publishers (e.g.. devices or applications that generate data) and route them over connections to consumers (applications 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.

Some open source software for IOT

Several software tools and solutions are used to deploy low-level IoT applications.

1. **Ardublock** is an open source visual block programming tool for Arduino systems
2. **Scratch** is a visual programming tool, developed at MIT, for IoT code generation and communication with Arduino-based IoT products
3. **NetLab** is an open source environment for developing embedded systems

Source: IoT-Enabled Smart Cities: A Review of Concepts, Frameworks and Key Technologies

- These devices are available in different forms:
- a. Microcontroller units (MCUs)
- b. Microcontroller development boards
- c. Single board computer (SBC)

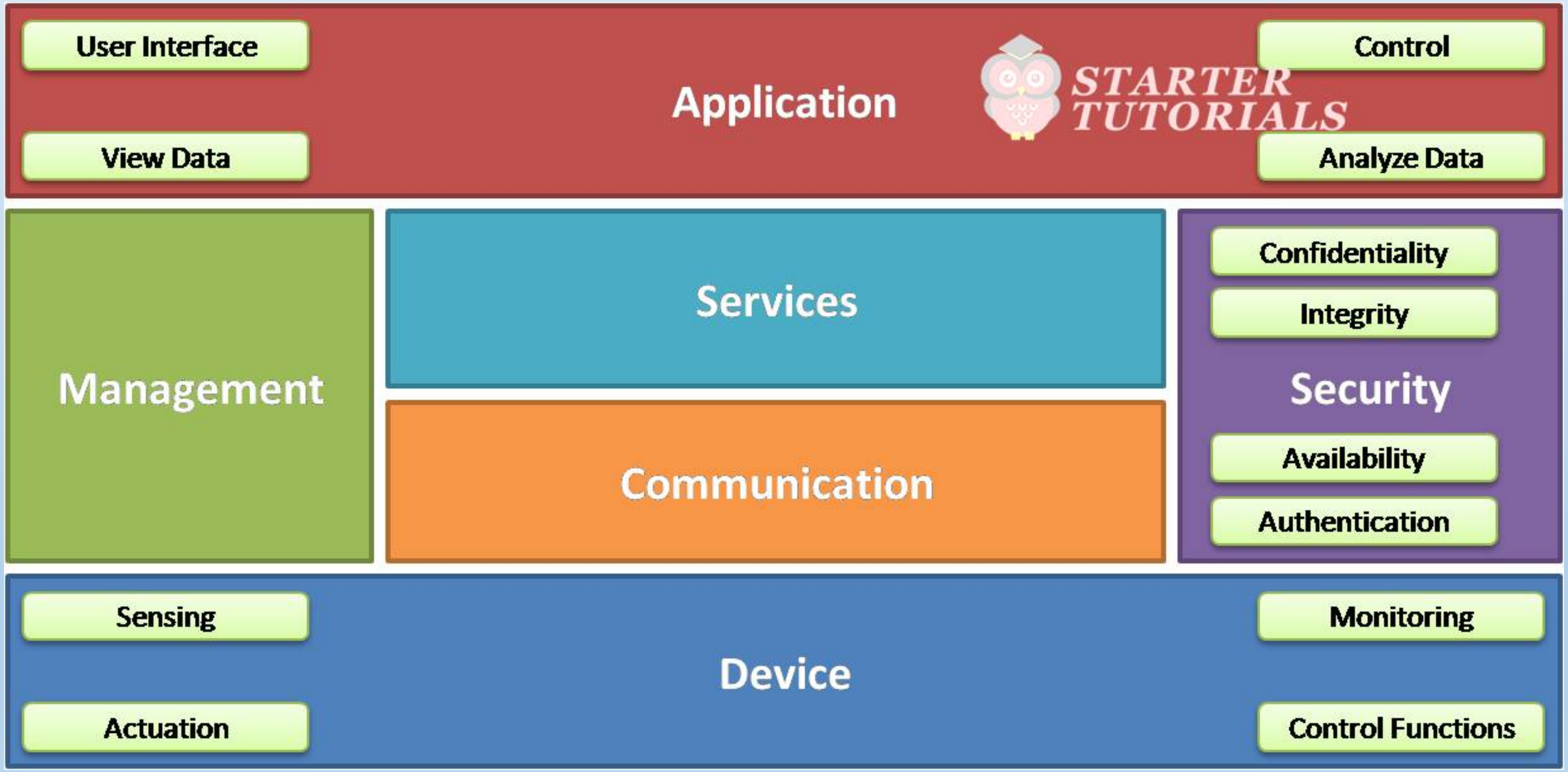
Microcontroller units	Microcontroller development boards	Single board computer (SBC)
PIC16F886-I/SO from Microchip	Arduino Uno	Raspberry Pi 3b
ATMEGA328P-MU from Atmel	Particle Photon	Asus tinker board
STM32F103C8T6 from ST Microelectronics	Adrafruit Flora	Qualcomm Dragonboard
MSP430G2402IN20 from Texas Instruments	LightBlue Bean	Raspberry Pi Zero W
	pcDuino	Rock64 Media board
	Beagle Bone	ODROID-XU4
	Intel Galileo	Orange Pi PC 2

Logical Design of IoT

- The “Logical Design” of IoT is the framework or the imaginary ideal design in which the components including software and the hardware components will be laid out.
- It doesn’t go into the depth of describing how each component will be built with low-level programming specifics.
- IoT Functional Blocks
- IoT communications models
- IoT Communication API

- Functional blocks: Functional blocks are parts of the system that interact with each other in order to let the services run properly. Following are the functional blocks of the IoT ecosystem:

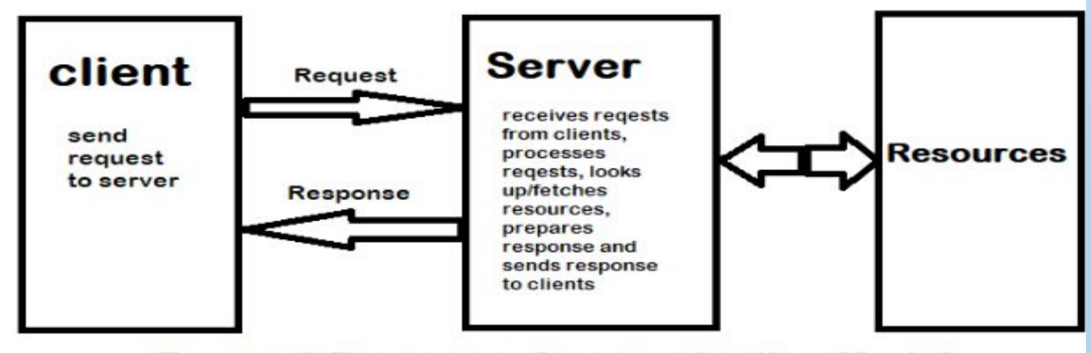
- Device
- Communication
- Services
- Management
- Security
- Application



- Communication models: Communication models are used for making the services of IoT reach up to the end users.
- Request-Response Communication model
- Publish Subscribe Communication Model
- Push Pull Communication model
- Exclusive pair Communication model

Request-Response Communication model:

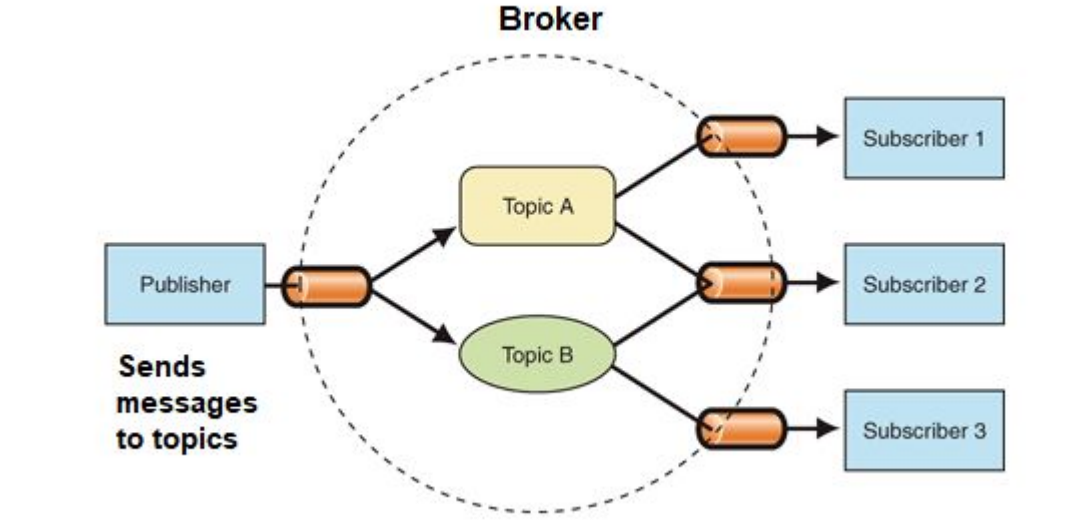
- The two main entities in request-response model are client and server.
- The client can be a web application, mobile application, browser requesting web pages or accessing an email. Each access of a resource will be treated as request.
- The server accepts the requests from the clients, processes them and sends back responses to the clients. While processing the requests, the server might access additional resources like file, databases, etc.
- The request-response model is stateless, i.e., the requests in a session are not related to each other with respect to a server.



Request-Response Communication Model

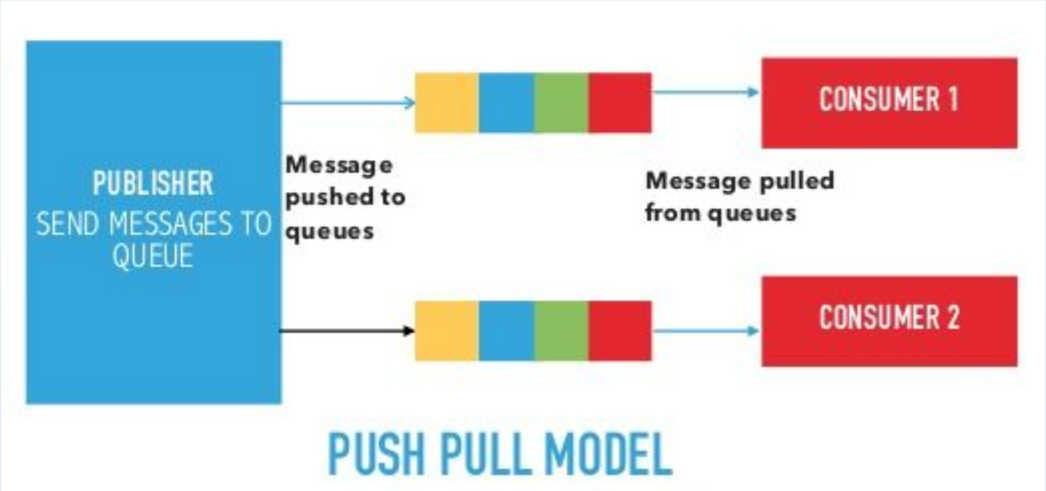
Publish-Subscribe communication model:

- It 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 receive data for a topic from the publisher, it sends the data to all the subscribed consumers.



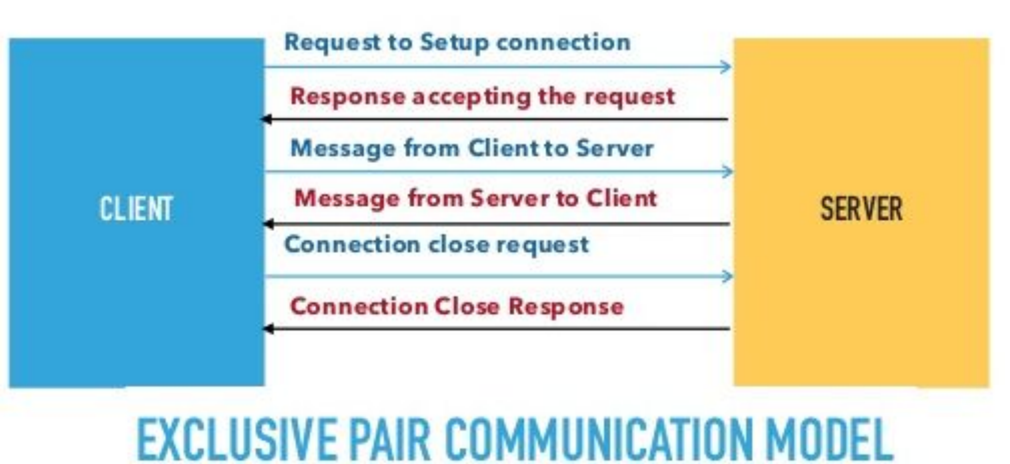
Push Pull Communication model:

- The main entities in a push-pull model are publisher, consumer, and queues.
- Publishers push the data in a queue and consumers are required to pull the data as and when required.



Exclusive pair Communication model:

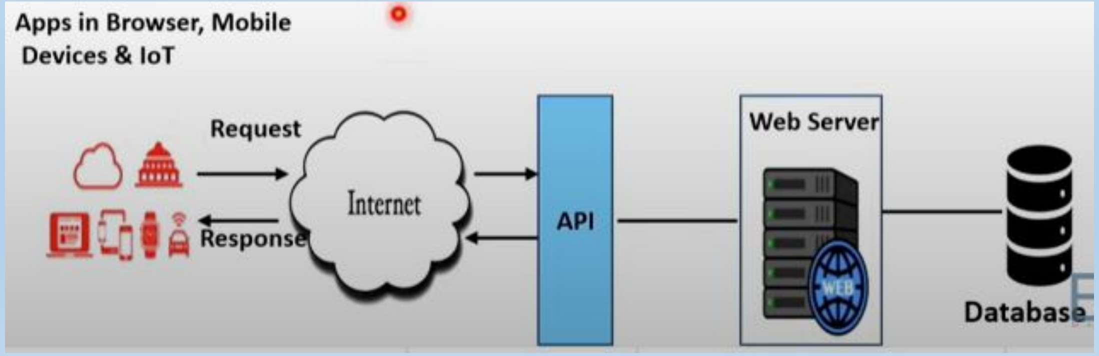
- The main entities in an exclusive pair model are client and server.
- This model is pure Client-Server model. Two devices come in an exclusive pair and work on full duplex CS model.
- Before sending a request, the client establishes a connection with the server and then sends the request.
- The data requests can be one or more. The server receives these requests through the same connection and sends back responses
- Exclusive pair model is stateful


Request to Setup connection
Response accepting the request
Message from Client to Server
Message from Server to Client
Connection close request
Connection Close Response
CLIENT
SERVER

EXCLUSIVE PAIR COMMUNICATION MODEL

IoT Communication APIs

- Application Programming Interface (API) is an interfacing software platform that allows the exchange of any information or data and supports the interaction among different applications or any such intermediaries.
- IoT APIs are the points of interaction between an IoT device and the internet and/or other elements within the network.
- Some popular communication APIS are:
- REST(Representational State Transfer) based Communication API
- WebSocket based Communication API



- **Client**: The person or program using the API. The client makes requests to the API in order to retrieve some information or change something within the application. Your web browser is a client — it interacts with APIs different websites to get page content from them. The requested info is sent back to your browser and displayed on your screen.
- **Resource:** Any piece of information that the API can provide the client. For instance, a resource in Facebook's API could be a user, a page, a photo, or a post. Each resource has a unique name, called the resource identifier.
- **Server:** is used by the application that receives client requests, and contains resources that the client wants. The server has an API to interact with clients without giving them direct access to content stored in its database.

REST-based Communication API

- REST is a set of guidelines that software can use to communicate over the internet in order to make integrations simple and scalable.
- REST design principles - also known as architectural constraints:

❖ **Uniform interface**

- The method of communication between a client and server must be uniform
- All API requests for the same resource should look the same, no matter where the request comes from.
- The REST API should ensure that the same piece of data, such as the name or email address of a user, belongs to only one uniform resource identifier (URI).
- Resources shouldn’t be too large but each message should includes enough information to describe how to process the message.

Client-server :

- In REST API design, client and server applications must be completely independent of each other.
- The only information the client application should know is the URI of the requested resource; it can't interact with the server application in any other ways.
- Similarly, a server application shouldn't modify the client application other than passing it to the requested data via HTTP.

❖ Statelessness

- REST APIs are stateless, meaning that each request needs to include all the information necessary for processing it.
- REST APIs do not require any server-side sessions.
- Server applications aren’t allowed to store any data related to a client request.

❖ **Cacheability**

- When possible, resources should be cacheable on the client or server side. Server responses also need to contain information about whether caching is allowed for the delivered resource. The goal is to improve performance on the client side, while increasing scalability on the server side.

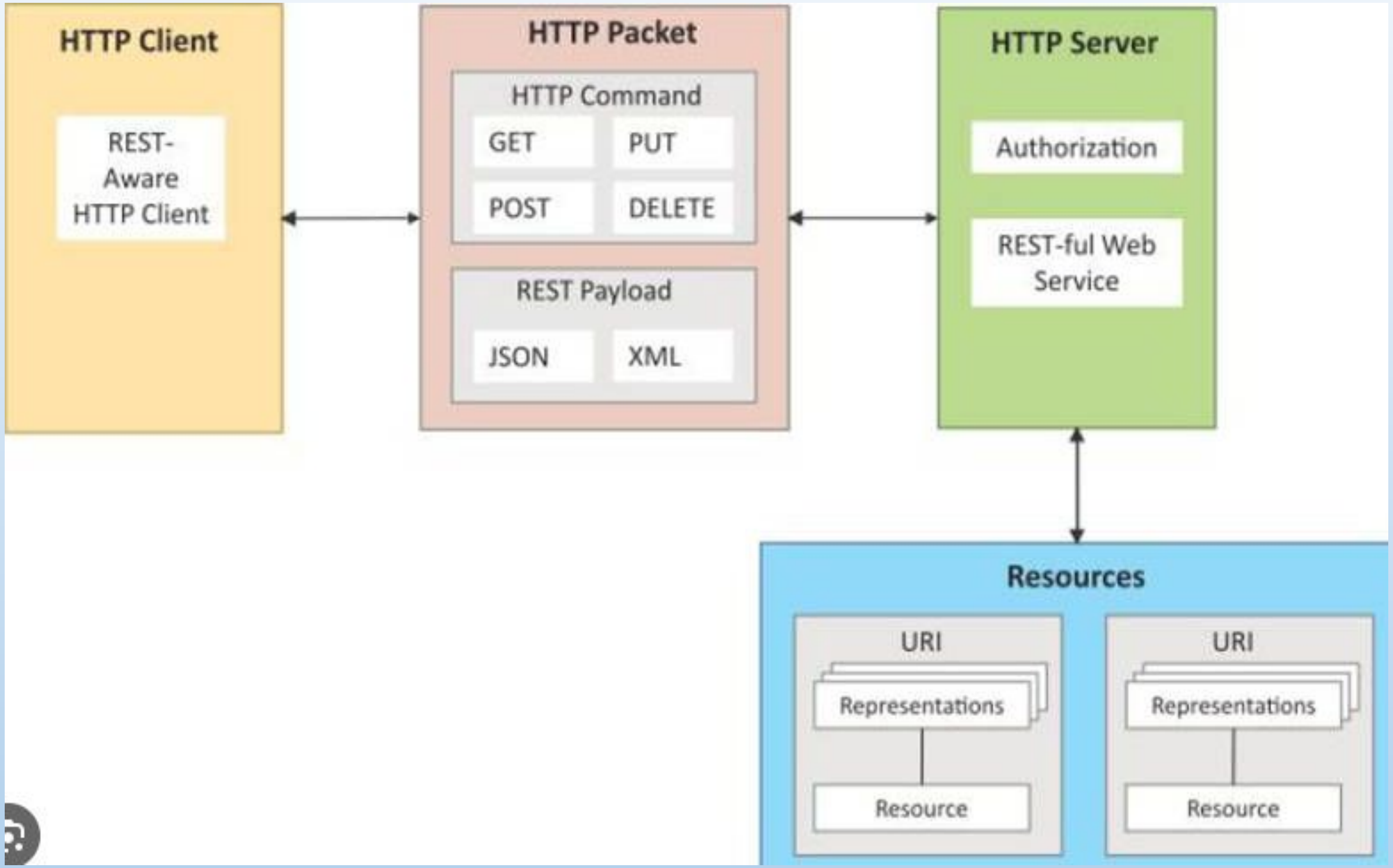
❖ Layered system architecture:

- In REST APIs, the calls and responses go through different layers.
- There may be a number of different intermediaries in the communication loop between client and server.
- Each component cannot see beyond the immidiate layer with which they are interacting.
- REST APIs need to be designed so that neither the client nor the server can tell whether it communicates with the end application or an intermediary.

❖ Code on demand (optional)

- Server can provide executable code or scripts for clent to execute in their context

https://blog.hubspot.com/website/what-is-rest-api



GET: To retrieve a resource.
POST: To create a new resource.
PUT: To edit or update an existing resource.
DELETE: To delete a resource

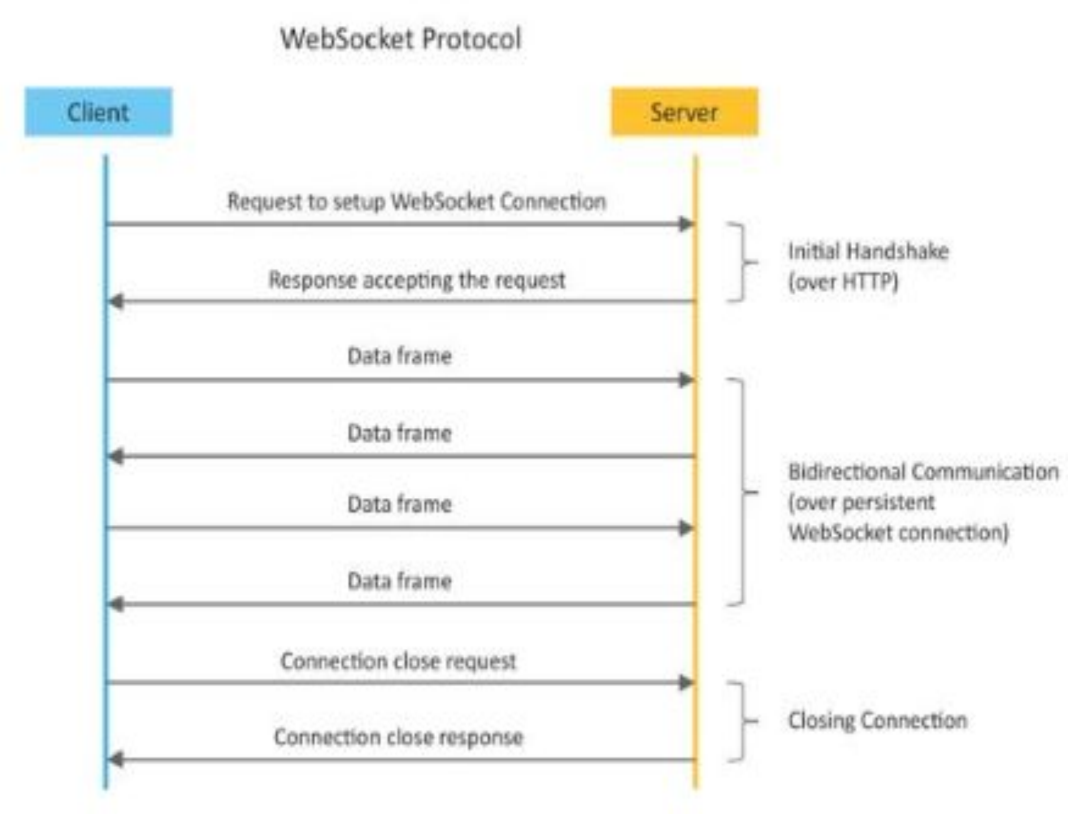
Communication with REST APIs

- Example:
- Let’s say I want to build a program that integrates with YouTube. My program (the client) can ask YouTube’s REST API for information about a specific video (a resource). YouTube’s API will respond to my request with the resource state, which includes attributes like the video name, publishing date, and view count, and video link, all packaged in a format that my program can quickly parse and use. My program could also post a video (i.e., add a new resource) to my personal YouTube channel through the API.

https://blog.hubspot.com/website/what-is-rest-api

WebSocket–Based Communication APIs

- WebSocket–based communication APIs facilitate full–duplex communication between clients and servers, adhering to an exclusive web pair communication model.
- This method of communication is characterized by its stateful nature, enabling ongoing and efficient interactions.
- It allows full duplex communicationand dont require a new connection to be setup for each message to be sent
- It reduces the network traffic and latency as there is no overhead for connection setup and termination request for each message


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

Exclusive pair model used by WebSocket APIs

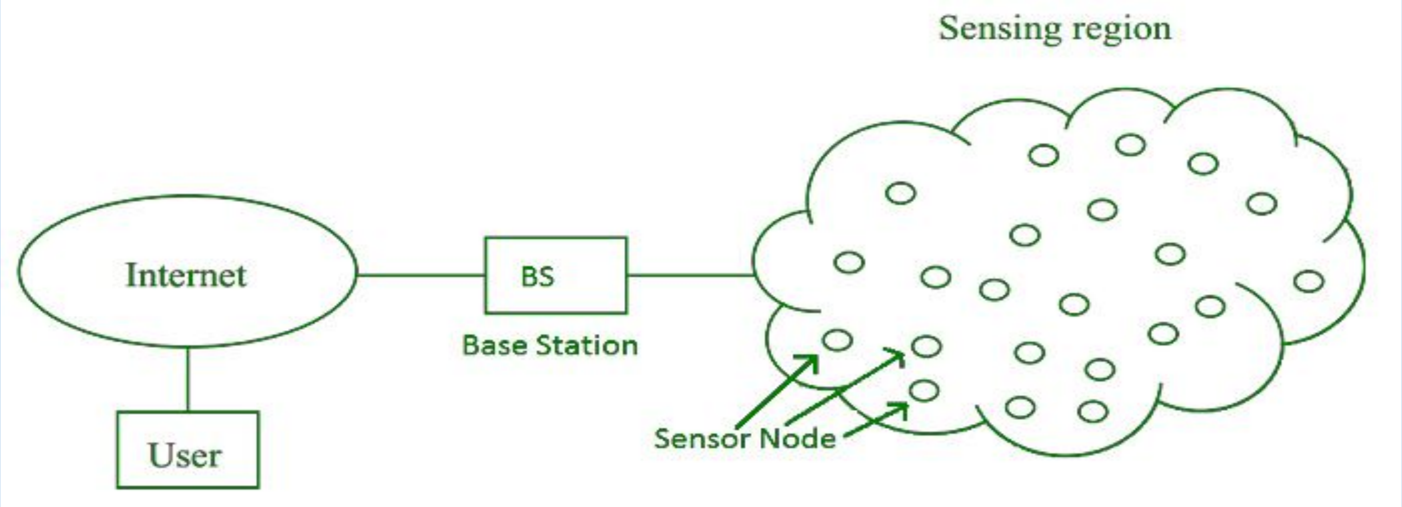
Comparison of REST and WebSocket

REST	WebSocket
Stateless	Stateful
Unidirectional	Bidirectional
Request Response	Full Duplex
Setting up new TCP	Single TCP
Not suitable for real-time application	suitable for real-time application
Scalability is easier	Scalability is combuersome

IoT Enabling Technologies

- IoT(internet of things) enabling technologies are
- Wireless Sensor Network
- Cloud Computing
- Big Data Analytics
- Communications Protocols
- Embedded System

- **Wireless Sensor Network(WSN) :**
- A Wireless Sensor Network (WSN) is a collection of devices which communicate through wireless channels to monitor environmental and physical conditions.
- It consists of end nodes, routers and coordinators.
- End nodes have several sensors attached to them where the data is passed to a coordinator with the help of routers. The coordinator also acts as the gateway that connects WSN to the internet.



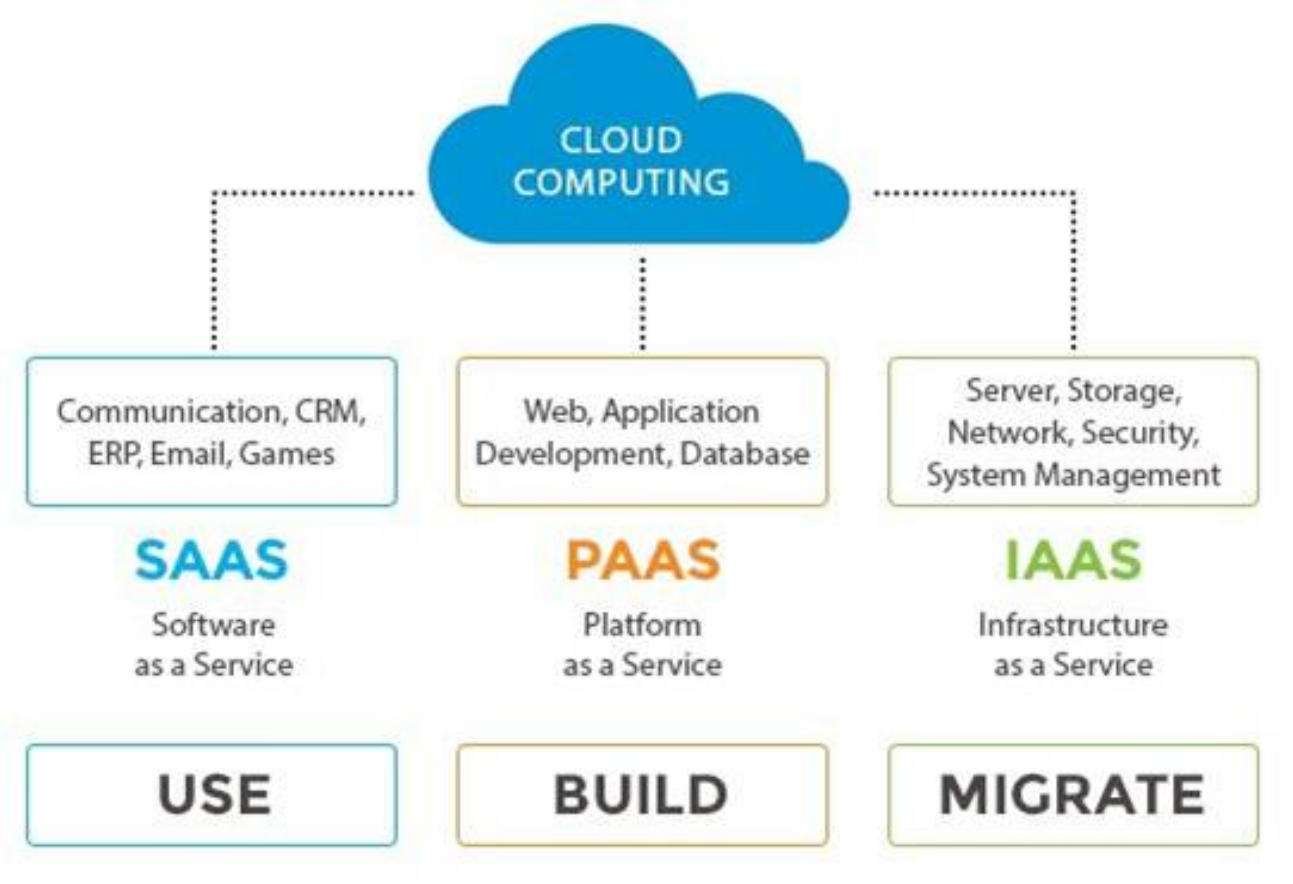
- Example:
- Weather monitoring system
- Indoor air quality monitoring system
- Soil moisture monitoring system
- Surveillance system
- Health monitoring system

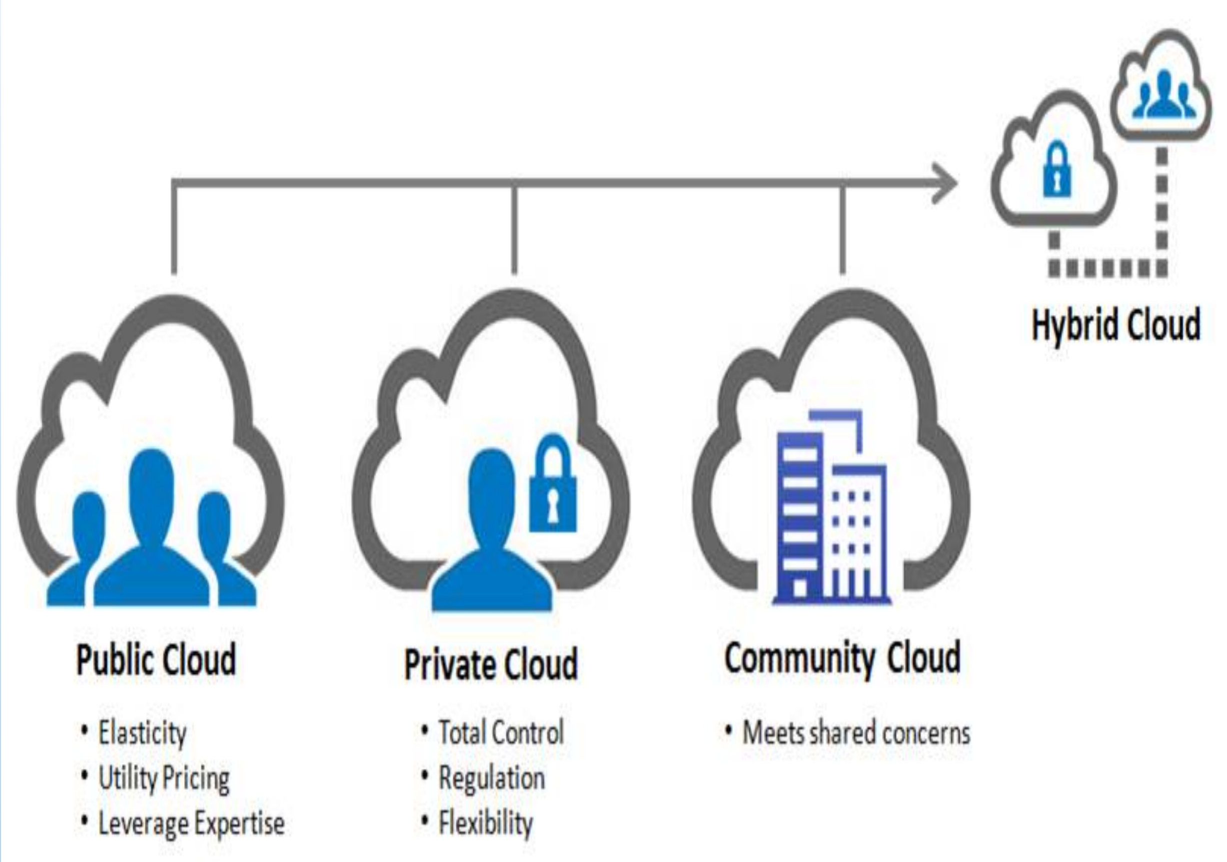
Cloud Computing :

- It provides us the means by which we can access applications as utilities over the internet. Cloud means something which is present in remote locations.
- With Cloud computing, users can access any resources from anywhere like databases, webservers, storage, any device, and any software over the internet.

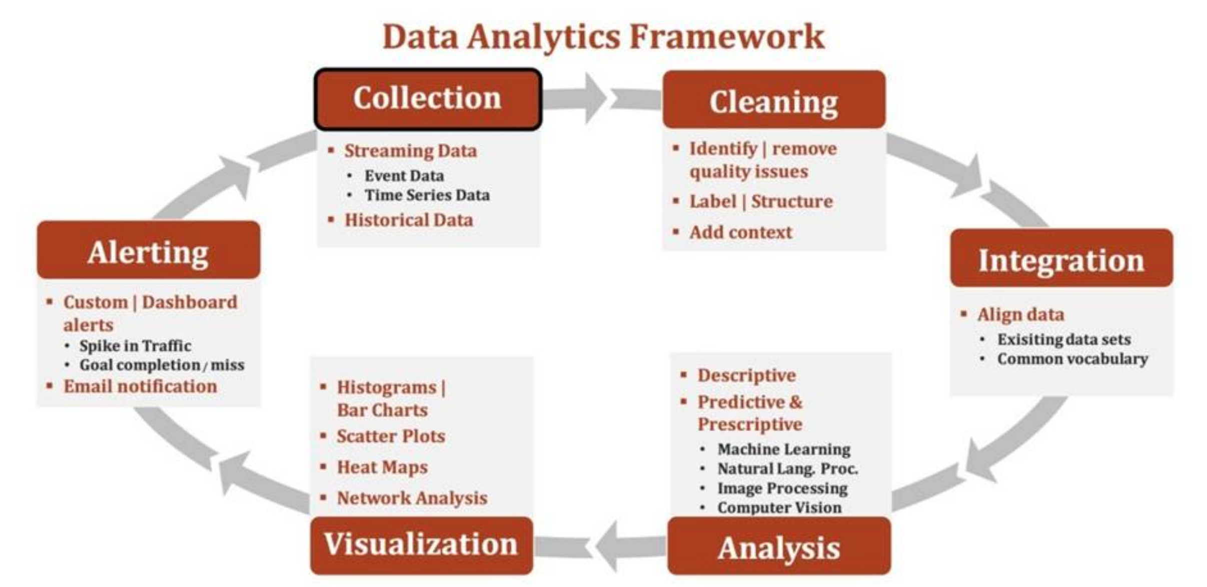
Characteristics:

- Broad network access
- On demand self-services
- Rapid scalability
- Pay-per-use


CLOUD COMPUTING
Communication, CRM, ERP, Email, Games
Web, Application Development, Database
Server, Storage, Network, Security, System Management
SAAS
Software as a Service
PAAS
Platform as a Service
IAAS
Infrastructure as a Service
USE
BUILD
MIGRATE


Hybrid Cloud
Public Cloud
• Elasticity
• Utility Pricing
• Leverage Expertise
Private Cloud
• Total Control
• Regulation
• Flexibility
Community Cloud
• Meets shared concerns

- Big Data Analytics :
- It refers to the method of studying massive volumes of data or big data. Collection of data whose volume, velocity or variety is simply too massive and tough to store, control, process and examine the data using traditional databases.
- Big data is gathered from a variety of sources including social network videos, digital images, sensors and sales transaction records.
- Several steps involved in analyzing big data –



Data Analytics Framework

Collection
- Streaming Data
  - Event Data
  - Time Series Data
- Historical Data

Cleaning
- Identify | remove quality issues
- Label | Structure
- Add context

Integration
- Align data
  - Exisiting data sets
  - Common vocabulary

Analysis
- Descriptive
- Predictive & Prescriptive
  - Machine Learning
  - Natural Lang. Proc.
  - Image Processing
  - Computer Vision

Visualization
- Histograms | Bar Charts
- Scatter Plots
- Heat Maps
- Network Analysis

Alerting
- Custom | Dashboard alerts
  - Spike in Traffic
  - Goal completion / miss
- Email notification

4. Communications Protocols :

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

They are used in

- Data encoding
- Addressing schemes

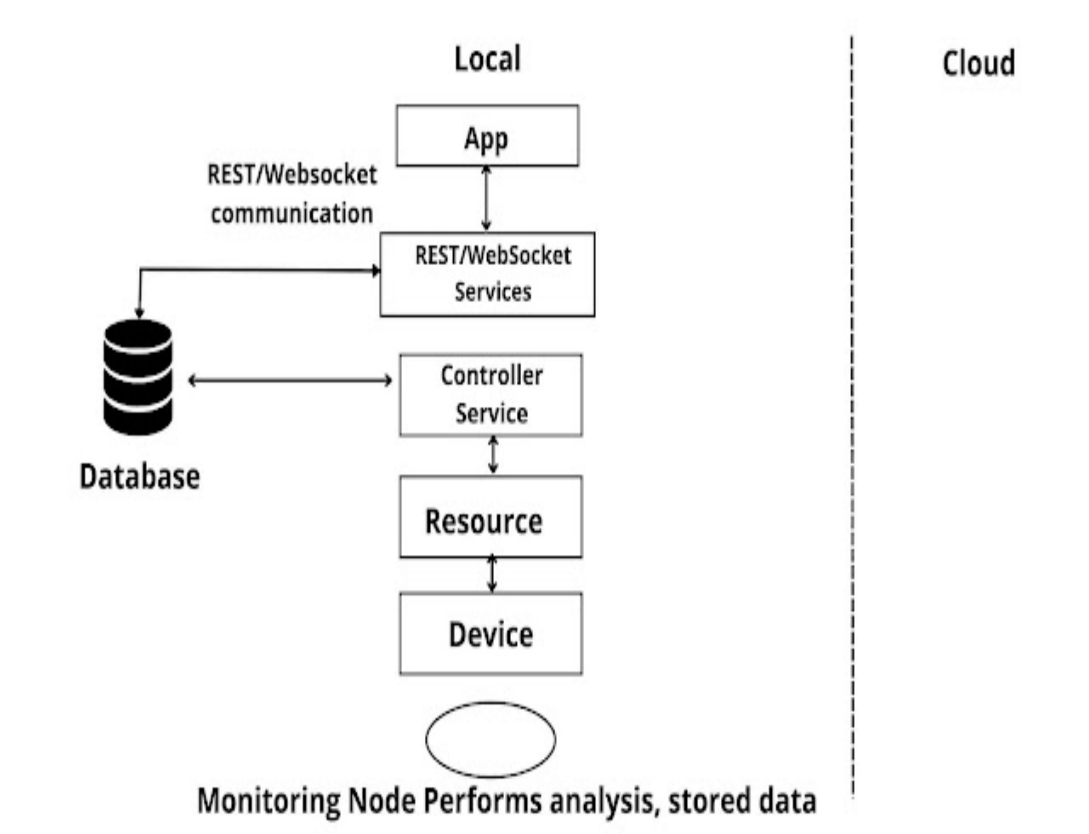
- Embedded Systems :
- It is a combination of hardware and software used to perform special tasks.
- It includes microcontroller and microprocessor memory, networking units (Ethernet Wi-Fi adapters), input output units (display keyword etc.) and storage devices (flash memory).
- It collects the data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

IoT Levels

- **Device**: These may be sensors or actuators capable of identifying, remote sensing, or monitoring.
- **Resources**: These are software components on IoT devices for accessing and processing.
- **Controller Service**: It is a service that runs on the device and interacts with web services.
- **Database**: Stores data generated from the device
- **Web Service**: It provides a link between IoT devices, applications, databases, and analysis components.
- **Analysis Component**: It performs an analysis of the data generated by the IoT device and generates results in a form which are easy for the user to understand.
- **Application**: It provides a system for the user to view the system status and view product data. It also allows users to control and monitor various aspects of the IoT system.

IoT Level-1

- IoT systems have a single device that performs sensing or actuation, stores and analyses it, and hosts the application
- IoT level-l is the best example for modeling of low complexity and low-cost solution where the analysis requirement is not comprehensive and the data involved is not big.

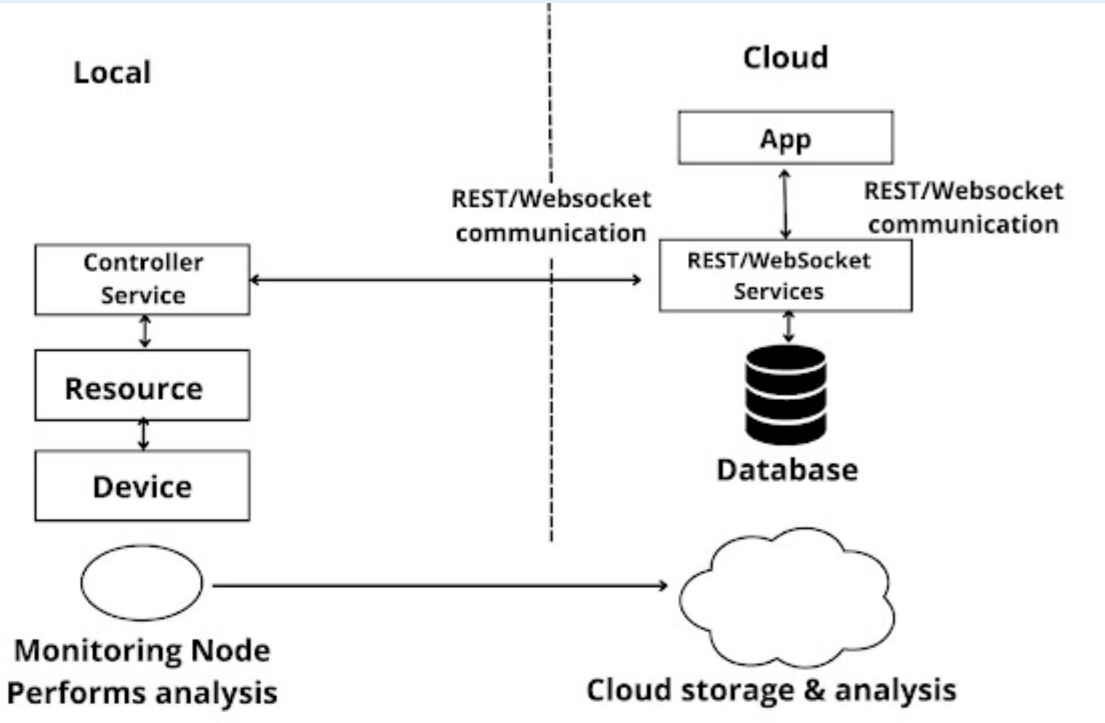


- Example: Let’s look at the IoT device that monitors the lights in a house. The lights are controlled through switches. The database has maintained the status of each light and also REST services deployed locally allow retrieving and updating the state of each light and trigger the switches accordingly. For controlling the lights and applications, the application has an interface. The device is connected to the internet and hence the application can be accessed remotely as well.

IoT level 2:

A node performs sensing/actuation and local analysis.

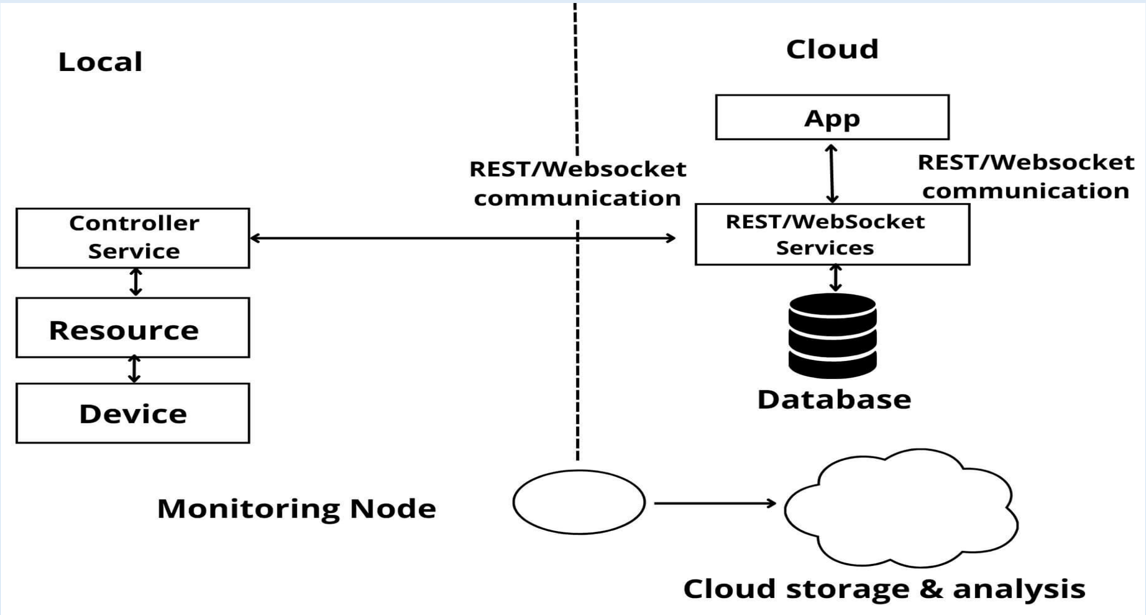
Data is stored in the cloud. This level is facilitated where the data involved is big and the primary analysis is not comprehensive.



- Example: Cloud-based application is used for monitoring and controlling the IoT system A single node monitors the soil moisture in the field Which is sent to the database on the cloud using REST APIS. The controller service continuously monitors moisture levels.

IoT level 3:

- At this level, the application is cloud-based.
- A single node monitors the environment and stores data in the cloud. This is suitable where data is comprehensive and analys is computationally intensive.



- Example: A node is monitoring a package using devices like an accelerometer and gyroscope. These devices track vibration levels. controller service sends sensor data to the cloud in the real time using WebSocket API. Data is stored in the cloud and visualized using a cloud-based application. The analysis component triggers an alert if vibration levels cross a threshold.

IoT – Level 3 Example: Tracking Package Handling

Sensors used

Accelrometer

sense movement or vibrations

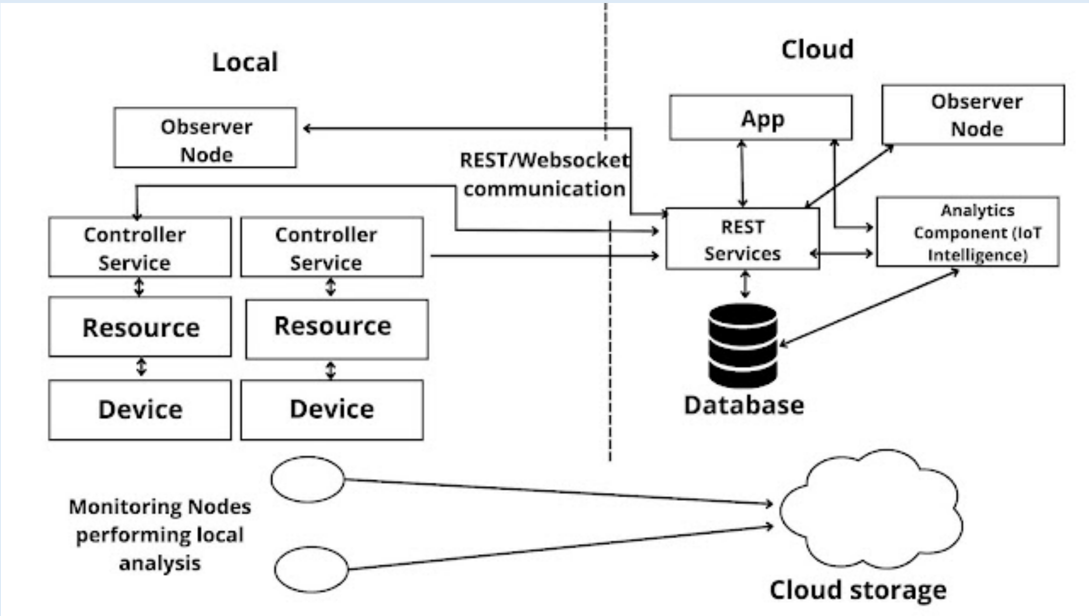
Gyroscope

Gives orientation info

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

IoT level 4

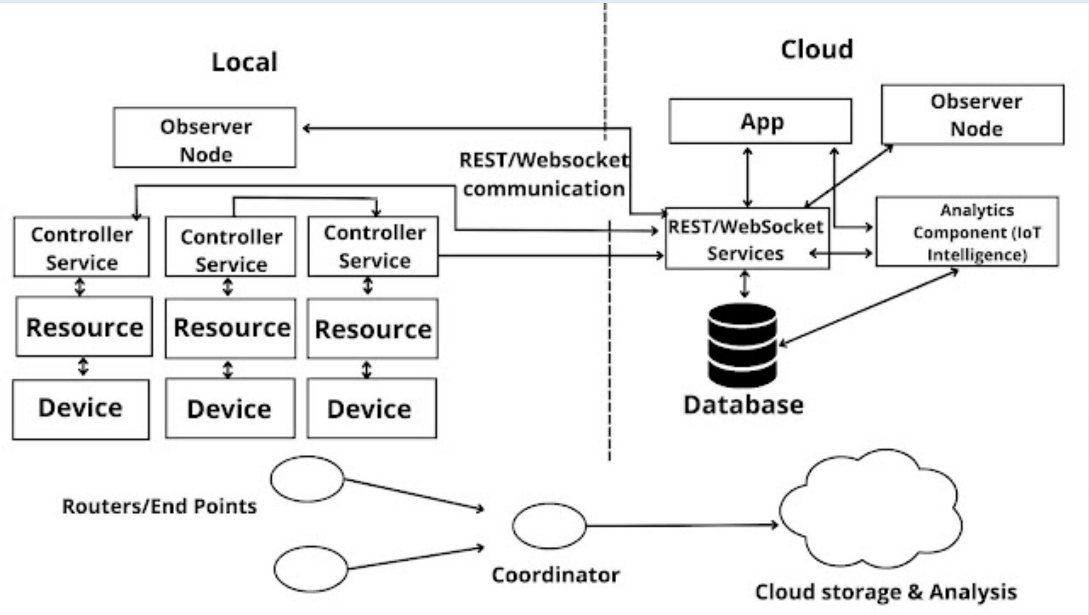
- At this level, Multiple nodes collect information and store it in the cloud.
- Local and rent server nodes are used to grant and receive information collected in the cloud from various devices.
- Observer nodes can process information and use it for applications but not perform control functions.
- This level is the best solution where data involvement is big, requirement analysis is comprehensive and multiple nodes are required.



- Example: Analysis is done on the cloud and the entire IoT system has monitored the cloud using an application. Noise monitoring of an area requires various nodes to function independently of each other. Each has its own controller service. Data is stored in a cloud database.

IoT level 5

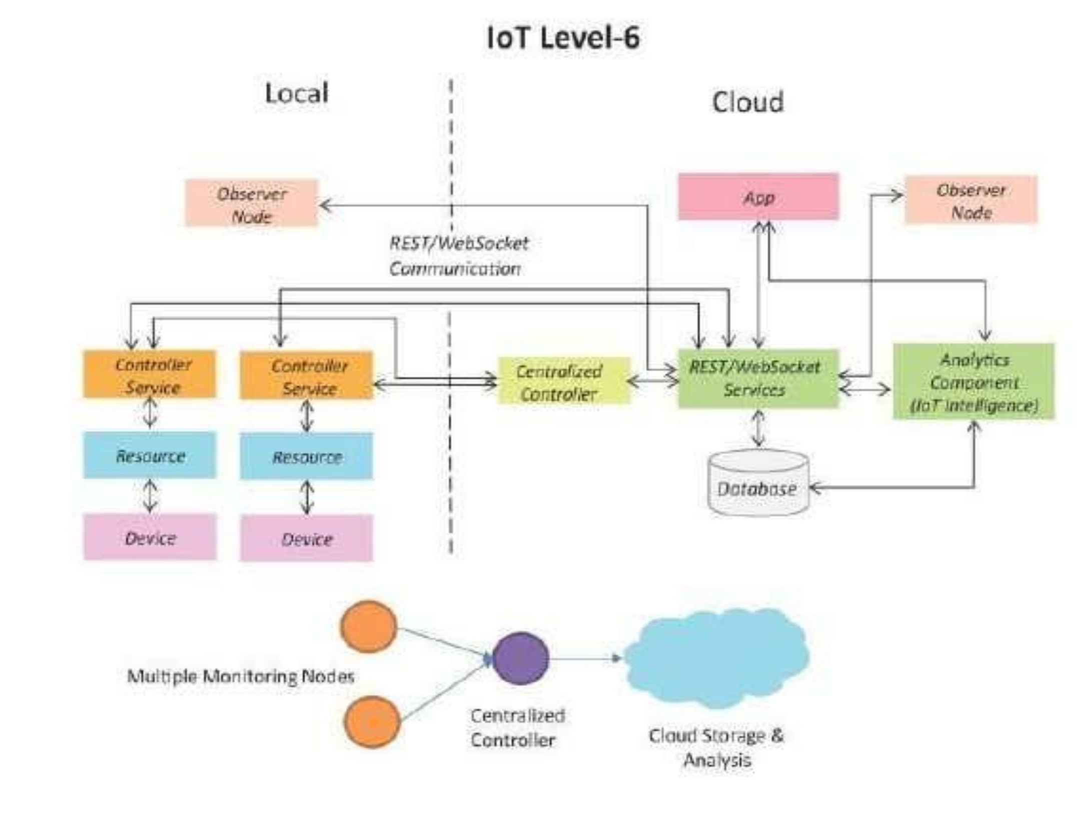
- Nodes present locally are of two types end odes and coordinator nodes
- End nodes collect data and perform sensing or actuation or both. Coordinator nodes collect data from end nodes and send it to the cloud.
- Data is stored and analyzed in the cloud. This level is best for WSN, where the data involved is big and the requirement analysis is comprehensive.

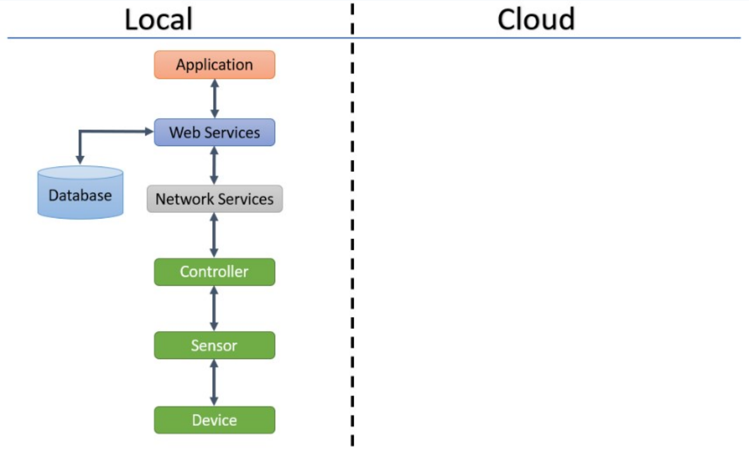


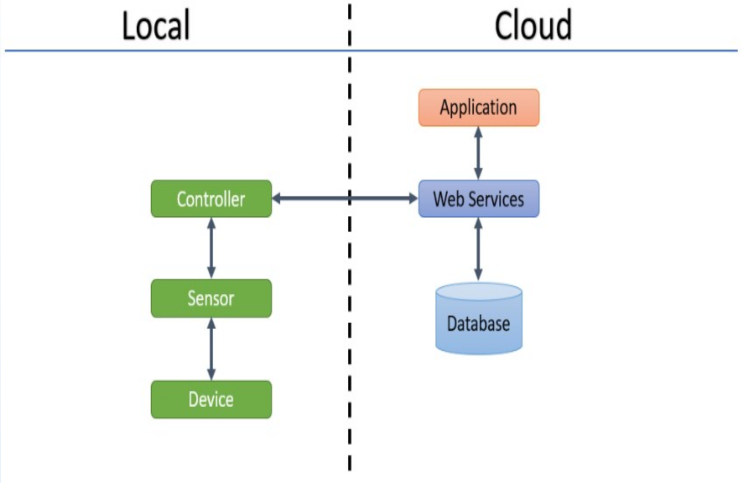
- Example: A monitoring system has various components: end nodes collect various data from the environment and send it to the coordinator node. The coordinator node acts as a gateway and allows the data to be transferred to cloud storage using REST API. The controller service on the coordinator node sends data to the cloud.

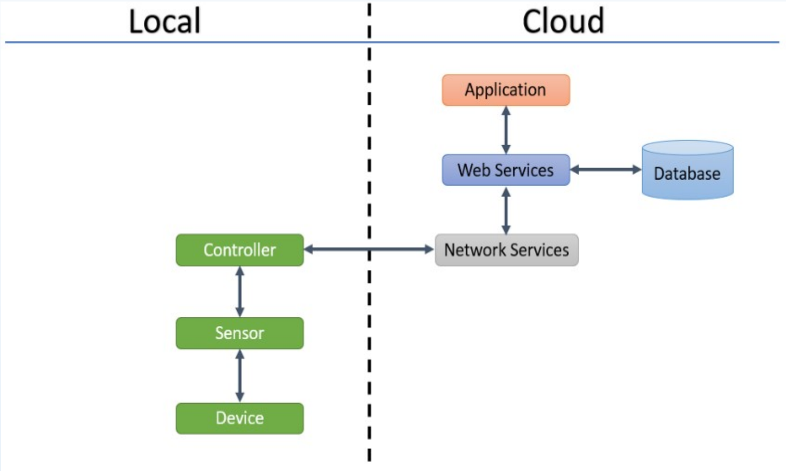
IoT Level 6

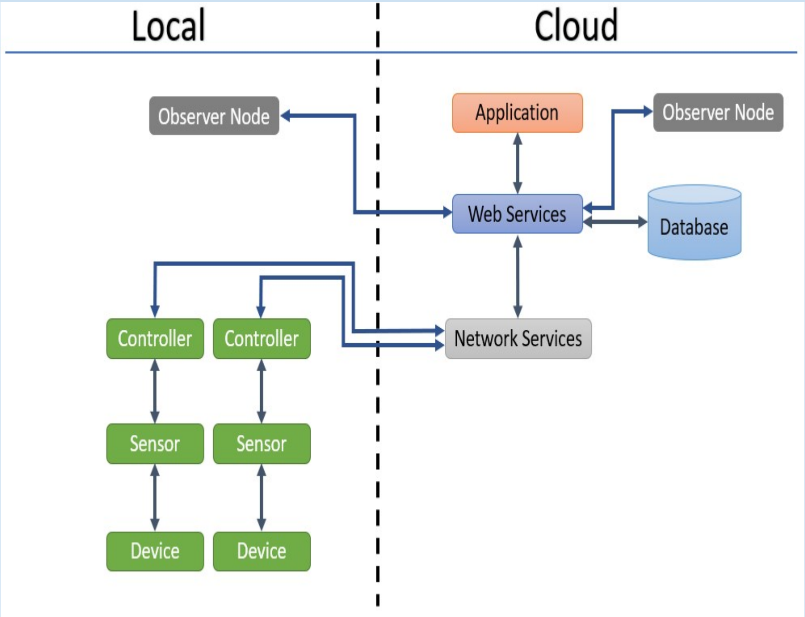
- At this level, the application is also cloud-based and data is stored in the cloud-like of levels.
- Multiple independent end nodes perform sensing and actuation and send d to the cloud.
- The analytics components analyze the data and store the results in the cloud database.
- The results are visualized with a cloud-based application.
- The centralized controller is aware of the status of all the end nodes and sends control commands to the nodes.

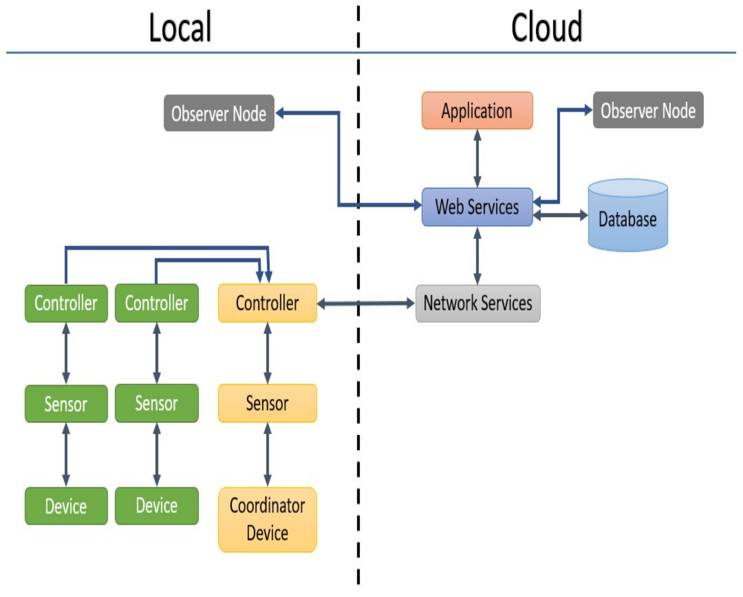


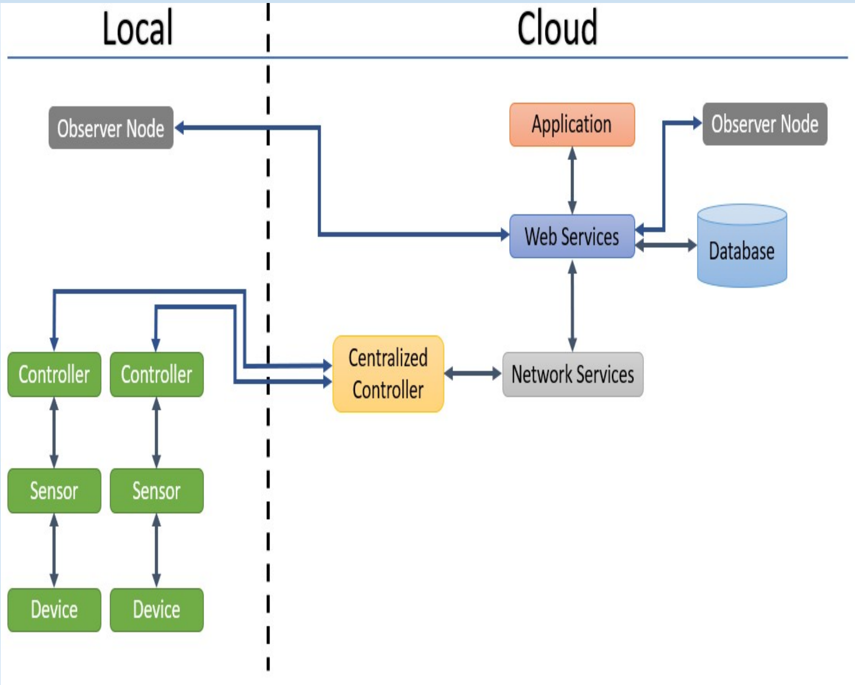

Local
Cloud
Application
Web Services
Database
Network Services
Controller
Sensor
Device


Local
Cloud
Application
Controller
Web Services
Sensor
Database
Device


Local
Cloud
Application
Web Services
Database
Controller
Network Services
Sensor
Device


Local
Cloud
Observer Node
Application
Observer Node
Web Services
Database
Controller
Controller
Network Services
Sensor
Sensor
Device
Device


Local
Cloud
Observer Node
Application
Observer Node
Web Services
Database
Controller
Controller
Controller
Network Services
Sensor
Sensor
Sensor
Device
Device
Coordinator Device


Local
Cloud
Observer Node
Application
Observer Node
Web Services
Database
Controller
Controller
Centralized Controller
Network Services
Sensor
Sensor
Device
Device

- Example: Weather monitoring consists of sensors that monitor different aspects of the system. The end nodes send data to cloud storage. Analysis of components, applications, and storage areas in the cloud. The centralized controller controls all nodes and provides inputs.