



# FUNDAMENTALS OF INTERNET OF THINGS

# UNIT-I-PART 1

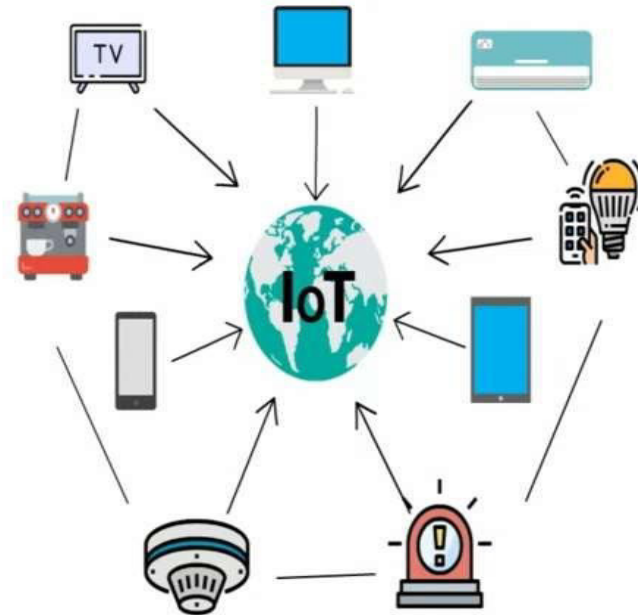
# INTRODUCTION

- **INTERNET:**

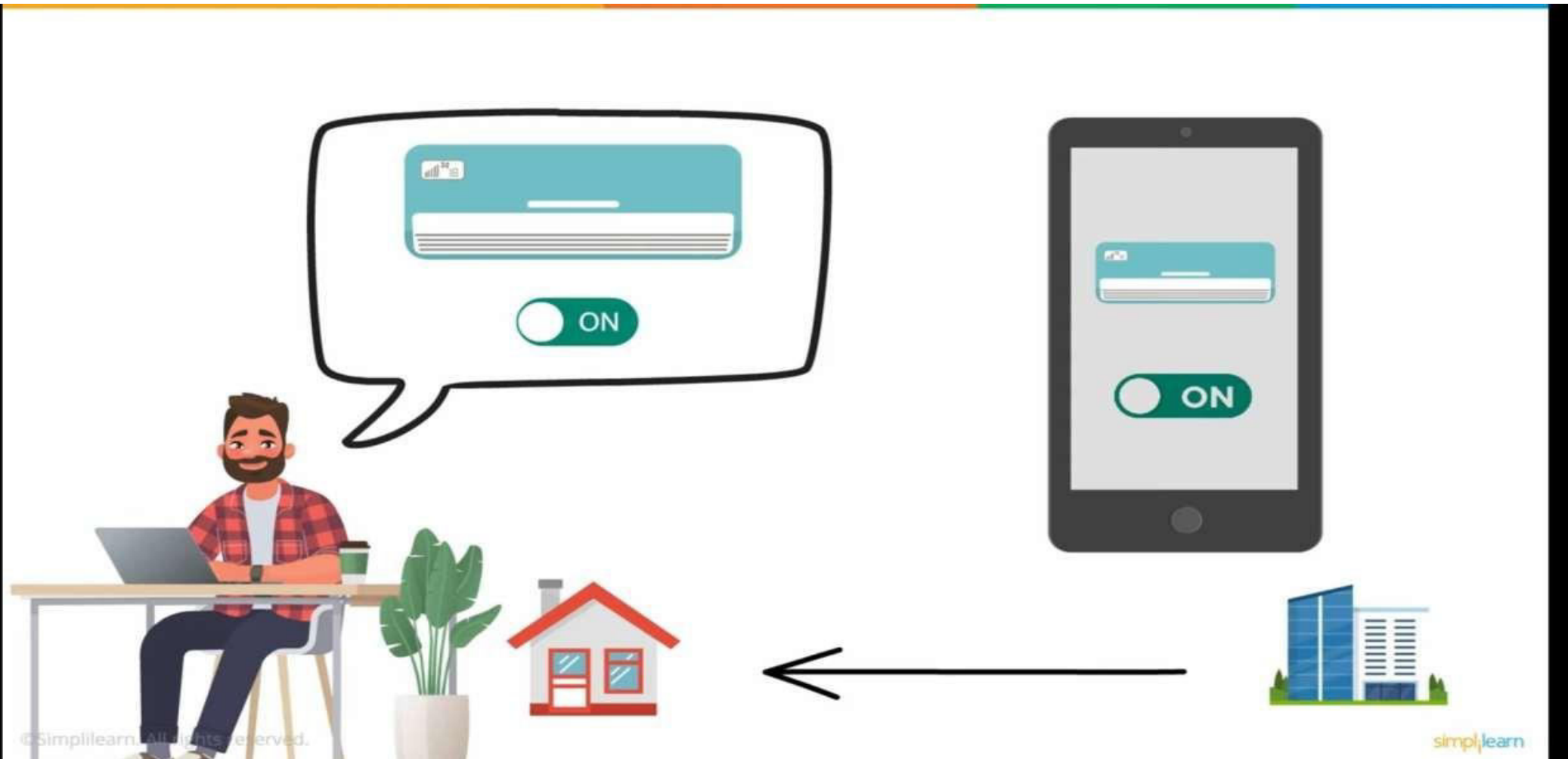
- Internet is a vast global network of connected servers, computers, tablets and mobiles that is governed by standard protocols for connected systems.
- It enables sending, receiving, or communication of information, connectivity with remote servers, cloud and analytics platforms.

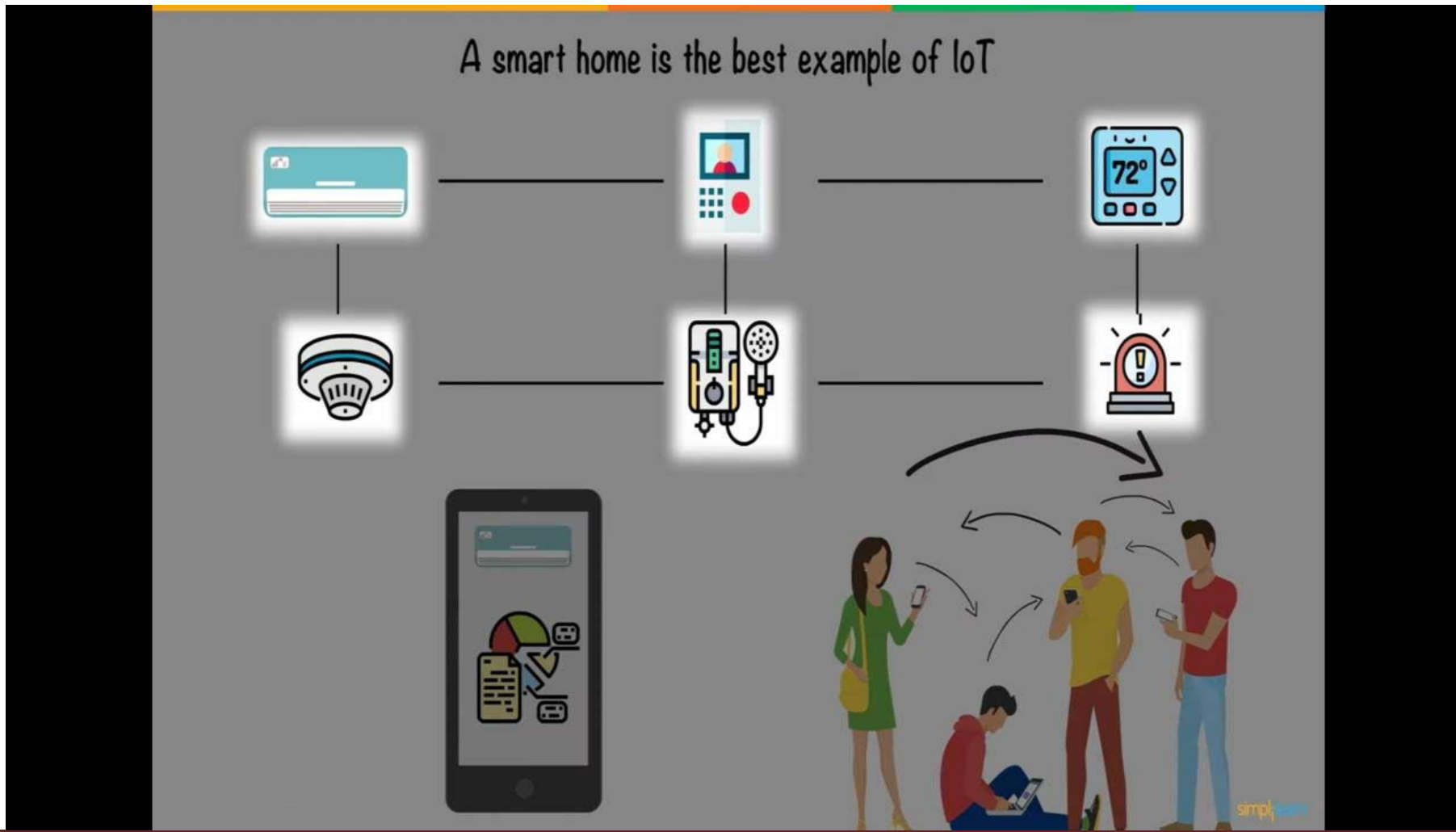
## Definition of iot

- A dynamic global network infrastructure with self configuring capabilities based on standard and
- interoperable communication protocols where physical and virtual “things” have identities, physical
- attributes, and virtual personalities and use intelligent interfaces and are seamlessly integrated into
- the information network, often communicate data associated with users and their environment.
- **Physical object+ controller, sensor and actuator + internet =IOT**



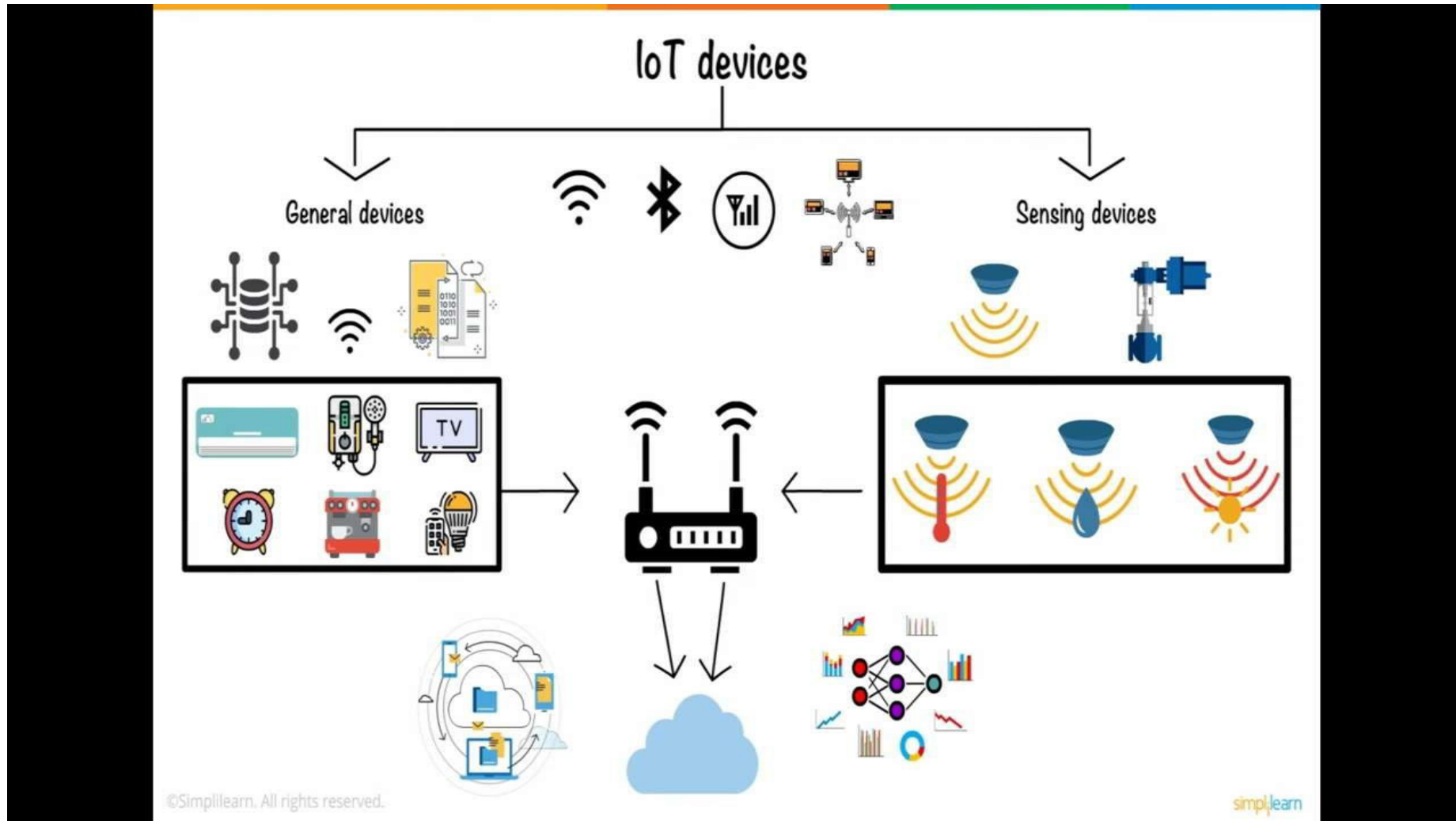
IoT is shaping the way we live our lives



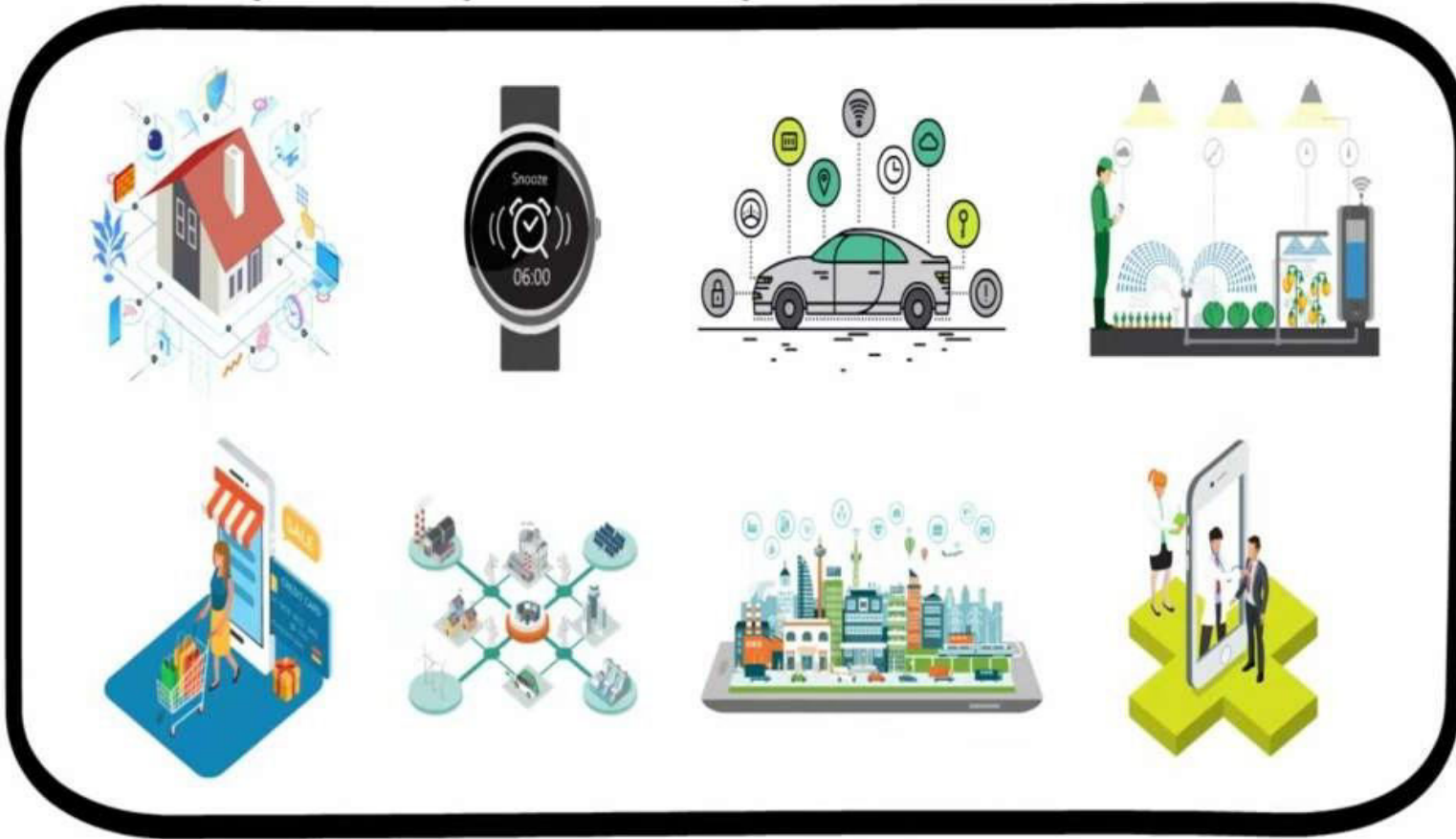




## FUNDAMENTALS OF INTERNET OF THINGS ( EC32110E)



Today, IoT is being used extensively to lessen the burden on humans



# IOT VISION

- IOT is a vision where things (wearable watches, alarm clocks, home devices, surrounding objects) become smart and function like living entities by sensing, computing and communicating through embedded devices which interact with remote objects (servers, clouds, applications, services and processes) or persons through the internet or Near-Field Communication (NFC).

## Characteristics of IOT

- ❖ Dynamic graded and self adapting
- ❖ Self configuring
- ❖ Interoperable communication protocols
- ❖ Unique identity
- ❖ Integrated into information network
- ❖ Heterogeneity
- ❖ Sensing
- ❖ Intelligence
- ❖ Large scale

## DYNAMIC AND SELF ADAPTING:

- IOT devices and systems may have the capability to dynamically adapt with the changing contexts and take action based on their operating conditions, users context or sensed environment.

## SELF CONFIGURING:

IOT devices may have self-configuring capability, allowing a large number of devices to work together to provide certain functionality like setup networking, fetch latest software upgrades.

## INTEROPERABLE COMMUNICATION PROTOCOLS:

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

## UNIQUE IDENTITY:

- Each IOT has unique identity and unique identifier (such as an IP address or a URI).
- This is helpful in tracking the equipment and at times to query its status.
- Device interface allow users to query the device, monitor the status, control them remotely.

## INTEGRATED INTO INFORMATION NETWORK:

- IOT devices are usually integrated into the information network that allows them to communicate and
- exchange data with other devices and systems.
- Can be dynamically discovered by other devices.
- Have the capability to describe themselves to other devices or user application.
- Integration into the information network makes IOT systems “smarter”.

## HETEROGENEITY:

- Devices in IOT are based on different hardware platforms and networks and can interact with other devices
- or service platform through different networks. IOT architecture should support direct network
- connectivity between heterogenous networks.

## SENSING

## INTELLIGENCE

## LARGE SCALE



# Physical Design of IoT:

Things in IoT  
IoT Protocols

# Things in IoT:

Refers to IoT devices which have unique identities that can perform sensing, actuating and monitoring capabilities.

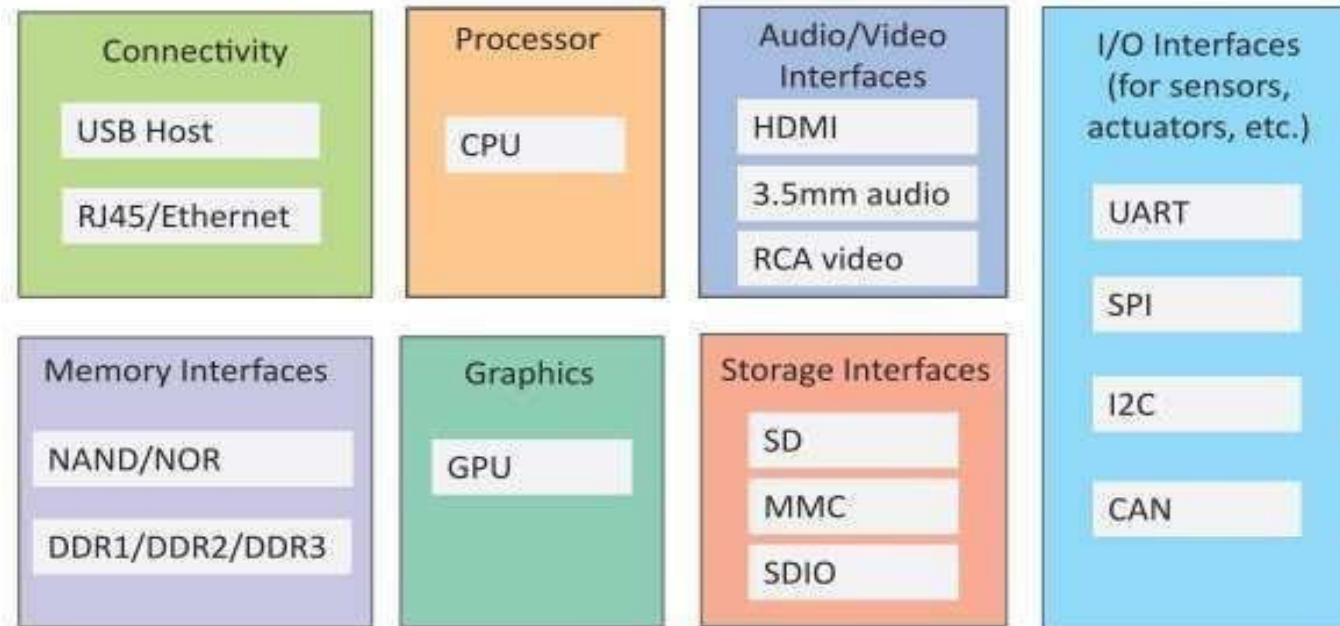
IoT devices can exchange data with other connected devices or collect data from other devices and process the data either locally or send the data to centralized servers or cloud – based application back-ends for processing the data.

# Generic Block Diagram of an IoT Device:

An IoT device may consist of several interfaces for connections to other devices, both wired and wireless.

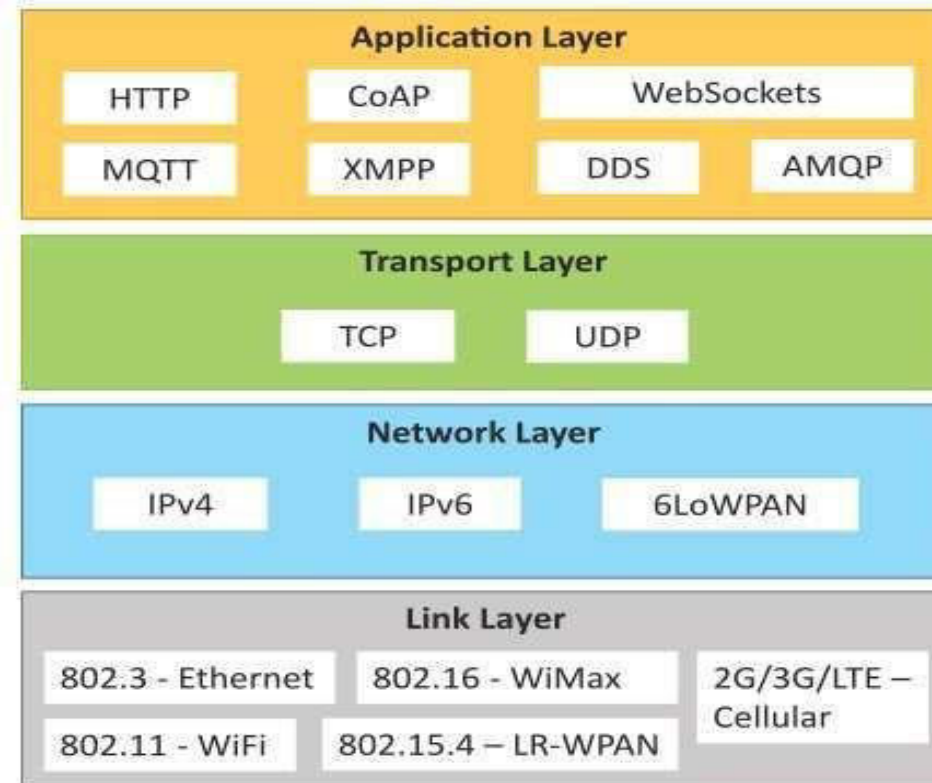
- I/O interfaces for sensors
- Interfaces for internet connectivity, Memory and storage interfaces ,Audio/video interfaces

## FUNDAMENTALS OF INTERNET OF THINGS ( EC32110E)



# IoT Protocols:

- Link Layer
- Network/Internet Layer
- Transport Layer
- Application Layer



# Link Layer:

The link layer is the group of methods and communications protocols confined to the link that a host is physically connected.

## Ethernet Standard:

<b>Sr. No</b>	<b>Standard</b>	<b>Shared medium</b>
<b>1</b>	<b>802.3</b>	<b>Coaxial Cable</b>
<b>2</b>	<b>802.3 .i</b>	<b>Copper Twisted pair</b>
<b>3</b>	<b>802.3 .j</b>	<b>Fiber Optic</b>
<b>4</b>	<b>802.3</b>	<b>Fiber.....10Gbits/s</b>

	<b>.ae</b>	
--	------------	--

S.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

**WiFi:** Data Rates  
from 1 Mb/s to  
6.75 Gb/s

**WiMax:** Data  
Rates from  
1.5Mb/s to 1  
Gb/s



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

- 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

2G/3G/4G –Mobile Communication: Data Rates from 9.6Kb/s (for 2G) to up to 100Mb/s (for 4G)

# 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

- 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

# 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

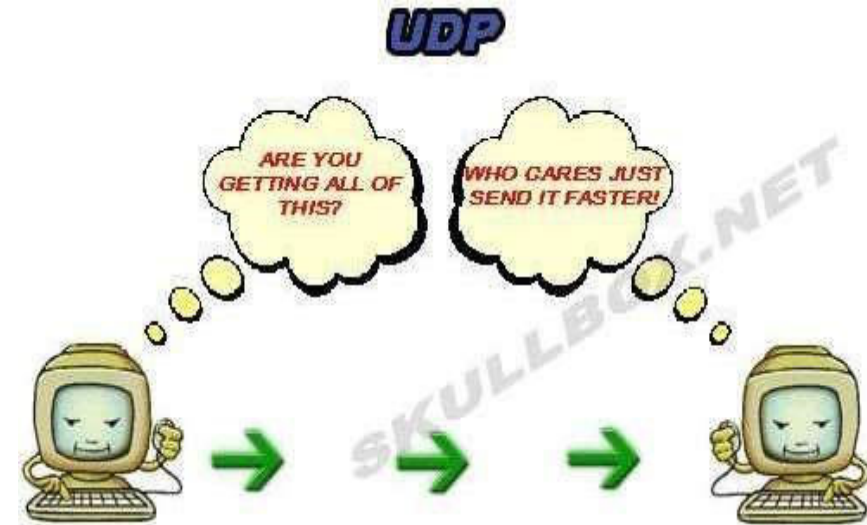
# TCP:

- Transmission Control Protocol
- Connection Oriented
- Ensures Reliable transmission
- Provides Error Detection Capability to ensure no duplicity 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



# 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



# 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

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



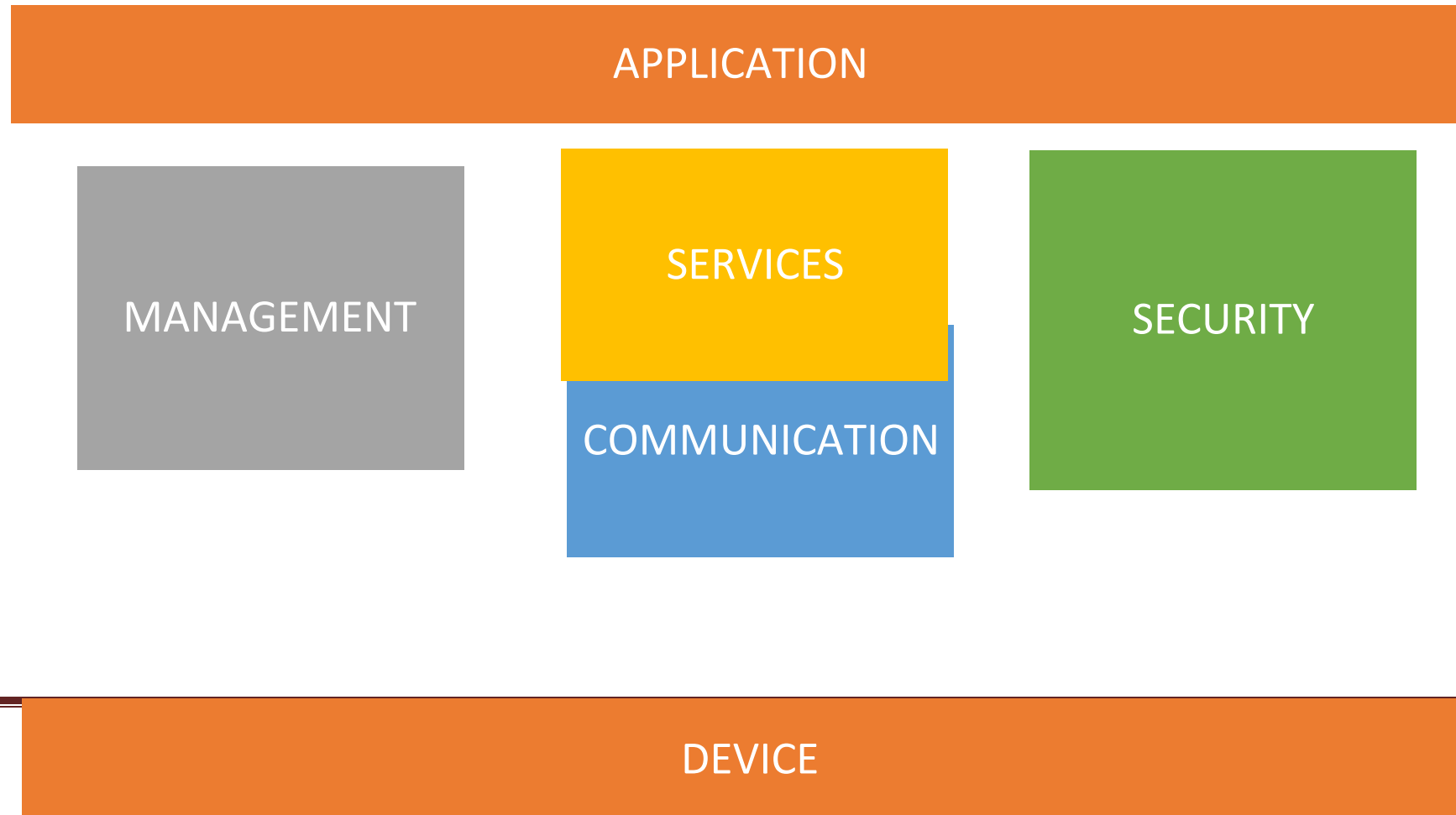
- **WebSocket:** Allows full-duplex communication over single socket, Based on TCP, Client can be a browser, IoT device or mobile application
- **MQTT:** Message Queue Telemetry Transport , light- weight messaging protocol, Based on publish- subscribe model, Well suited for constrained environments where devices have limited processing, low memory and n/w bandwidth requirement
- **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



# Functional blocks of IOT

- An IOT system comprises of a number of functional blocks that provide the system the capabilities for
- identification, sensing, actuation, communication and management.

# Functional Block of IOT



# Functional blocks are:

- **Device:** An IoT system comprises of devices that provide sensing, actuation, monitoring and control functions.
- **Communication:** Handles the communication for the IoT system.
- **Services:** services for device monitoring, device control service, data publishing services and services for device discovery.

- **Management:** This block provides various functions to govern the IoT system.
- **Security:** This block secures the IoT system and by providing functions such as authentication, authorization, message and content integrity, and data security.
- **Application:** This is an interface that the users can use to control and monitor various aspects of the IoT system.

Application also allow users to view the system status and view or analyze the processed data.

# TRANSDUCER



- A transducer is any device which converts one form of energy into another. Examples of common transducers include the following
  - A **microphone** converts sound into electrical impulses and a loudspeaker converts electrical impulses into sound (i.e., sound energy to electrical energy and vice versa).
  - A **solar cell** converts light into electricity and a thermocouple converts thermal energy into electrical energy.
  - An **incandescent light bulb** produces light by passing a current through a filament. Thus, a light bulb is a transducer for converting electrical energy into optical energy.
  - An **electric motor** is a transducer for conversion of electricity into mechanical energy or motion.



# SENSORS

- A sensor is a device that receives and responds to a signal.
- This signal must be produced by some type of energy, such as heat, light, motion, or chemical reaction.
- Once a sensor detects one or more of these signals (an input), it converts it into an analog or digital representation of the input signal.
- sensors are used in all aspects of life to detect and/or measure many different conditions.
- Human beings are equipped with 5 different types of sensors.



Detects  
Light



Detects  
Sound



Detects  
Certain Chemicals



Detects  
Pressure & Temperature



## Basic Concepts of Sensors

- **Sensors detect the presence of energy, changes in or the transfer of energy. Sensors detect by receiving a signal from a device such as a transducer, then responding to that signal by converting it into an output that can easily be read and understood.**
- **Typically sensors convert a recognized signal into an electrical – analog or digital – output that is readable. In other words, a transducer converts one form of energy into another while the sensor converts the output of the transducer to a readable format.**

- Consider the previous examples of transducers. They convert one form of energy to another, but they do not quantify the conversions.
- The light bulb converts electrical energy into light and heat; however, it does not quantify how much light or heat.
- A battery converts chemical energy into electrical energy but it does not quantify exactly how much electrical energy is being converted.
- 
- If the purpose of a device is to quantify an energy level, it is a sensor.

# Different types of sensors

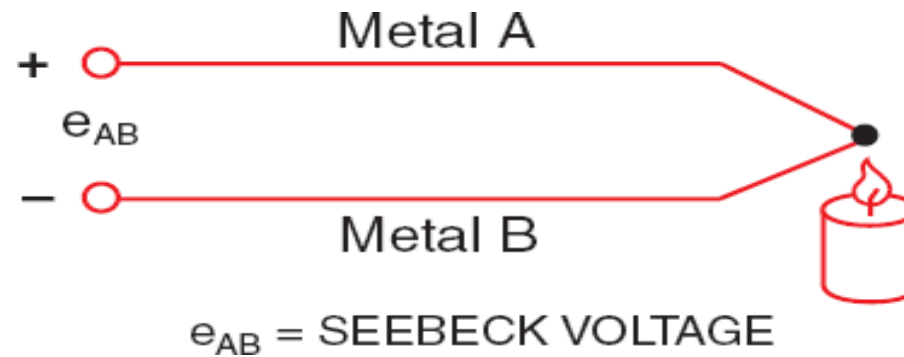
- **Basically sensors are being subdivided into**
- Thermal Sensors
- Mechanical Sensors
- Electrical Sensors
- Chemical Sensors
- **Thermal Sensors**
- Thermometer – measures absolute temperature (*discussed in the previous section*)
- Thermocouple gauge– measures temperature by its affect on **two dissimilar metals**
- Calorimeter – measures the heat of chemical reactions or physical changes and heat capacity

**A thermocouple is a device that directly converts thermal energy into electrical energy.**

When two dissimilar metal wires are connected at one end forming a junction, and that junction is heated, a voltage is generated across the junction.

If the opposite ends of the wires are connected to a meter, the amount of generated voltage can be measured.

This effect was discovered by Thomas Seebeck, and thus named the Seebeck Effect or Seebeck coefficient. **The voltage created in this situation is proportional to the temperature of the junction..**



## Mechanical Sensors

- Pressure sensor – measures pressure
- Barometer – measures atmospheric pressure
- Altimeter – measures the altitude of an object above a fixed level
- Liquid flow sensor – measures liquid flow rate
- Gas flow sensor – measures velocity, direction, and/or flow rate of a gas

- Accelerometer – measures acceleration

### Pressure sensor:

A pressure sensor senses the pressure applied ie, force per unit area, and it converts into an electrical signal. It has high importance in weather forecasting.

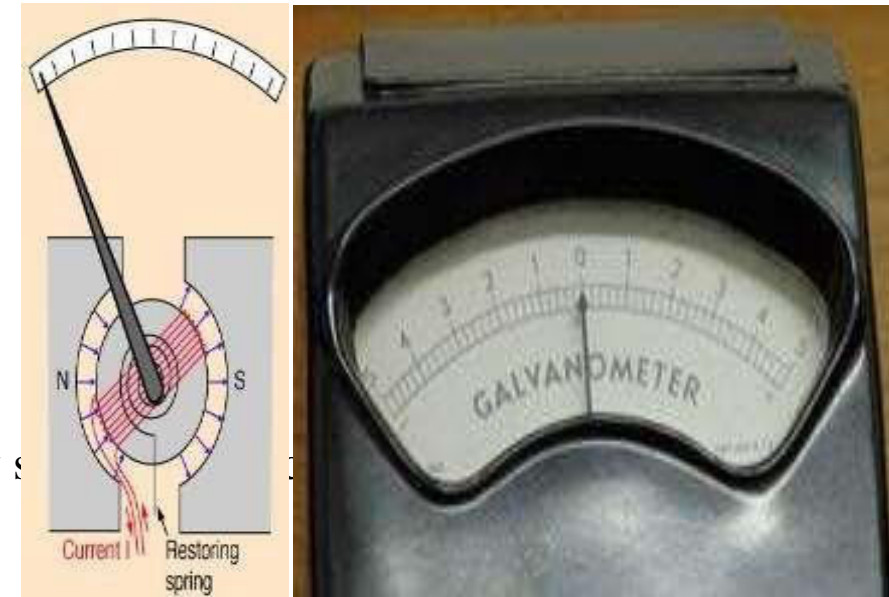
There are various Pressure sensors available in the market for many purposes.

**For example, all the smartphones, wearables have these biometric pressure sensors integrated into them.**



## Electrical Sensors

- Ohmmeter – measures resistance
- Voltmeter – measures voltage
- Galvanometer – measures current
- Watt-hour meter – measures the amount of electrical energy consumed in business



## **Chemical Sensors**

Chemical sensors detect the presence of certain chemicals or classes of chemicals and quantify the amount and/or type of chemical detected.

- Oxygen sensor – measures the percentage of oxygen in a gas or liquid being analyzed
- Carbon dioxide detector – detects the presence of CO<sub>2</sub>

- Chemical sensing is an application that really benefits from the use of microtechnology.
- 
- chemical sensors can detect a wide variety of different gases.
- The advantage of the MEMS sensors is that they can be incorporated into objects for continuous sensing of a gas or selection of gases.
- These devices have numerous medical, industrial, and commercial applications such as environmental, quality control, food processing, and medical diagnosis.

### Other Types of Sensors

#### Optical

- Light sensors (photodetectors) – detects light and electromagnetic energy
- Photocells (photoresistor) – a variable resistor affected by intensity changes in ambient light.
- Infra-red sensor – detects infra-red radiation

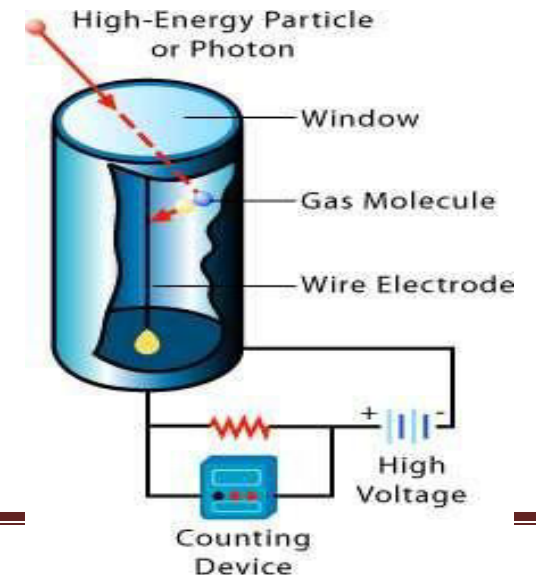
#### Acoustic

- Seismometers – measures seismic waves
- Acoustic wave sensors – measures the wave velocity in the air or an environment to detect the chemical species present

#### Other

- Motion – detects motion
- Speedometer – measures speed
- Geiger counter – detects atomic radiation (*see graphic*)
- Biological – monitors human cells

*Geiger Counter: Detects Atomic Radiation*



### **Proximity sensor:**

A proximity sensor is a sensor able to detect the presence of nearby objects without any physical contact.

A proximity sensor often emits an electromagnetic field or a beam of electromagnetic radiation and looks for changes in the field or return signal.

A most common application of this sensor is used in cars. While you are taking the reverse , it detects or objects or obstacles and you will be alarmed.

### **Accelerator sensor:**

Accelerometers in mobile phones are used to detect the orientation of the phone.

### **INFRARED SENSORS:**

- An infrared sensor is an electronic device, which senses certain characteristics of its surroundings by emitting infrared radiation
- It has the ability to measure the heat being emitted by an object and also measures the distance.
- It has implemented in various applications. It is used in radiation thermometers depend on the material of the object.
- Level sensors
- Image sensors
- Water quality sensors
- Chemical sensors
- Gas sensors
- Smoke sensors
- Humidity sensors

- optical sensors

## Basic Concepts of Actuators

1)**An actuator is something that actuates or moves something.** More specifically, an actuator is a device that converts energy into motion or mechanical energy. Therefore, an actuator is a specific type of a transducer.

2)**Thermal Actuators One type of thermal actuator is a bimetallic strip.** This device directly converts thermal energy into motion. This is accomplished by utilizing an effect called thermal expansion. Thermal expansion is the manifestation of a change in thermal energy in a material.

When a material is heated, the average distance between atoms (or molecules) increases. The amount



of distance differs for different types of material. This microscopic increase in distance is unperceivable to the human eye. However, because of the huge numbers of atoms (or molecules) in a piece of material, the material expands considerably and, at times, is noticeable to the human eye.

The opposite reaction occurs for a decrease in temperature when most materials contract.

When exposed to the elements, a material constantly expands and contracts with ambient temperature changes. Consider a piece of steel 25 meters long.

If the temperature of the steel increases by 36°C, (the difference between a cold winter day and a hot summer day), that piece of steel lengthens approximately 12 cm.

This change in length is the thermal linear expansion. It is calculated by using the following formula:  $\delta_L = aL_o\Delta T$

Where  $\delta_L$  is the change in length,  $a$  is the coefficient of linear expansion,  $L_o$  is the original length, and  $\Delta T$  is the change in temperature in Celsius.

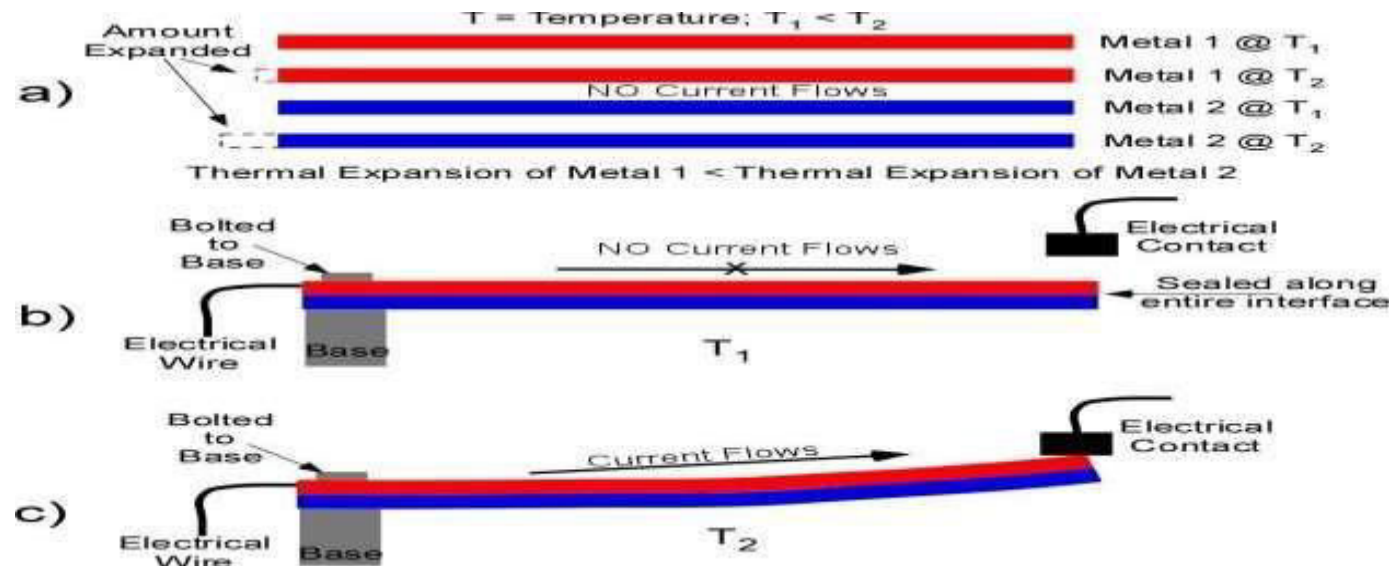
If we are considering steel, the coefficient of linear expansion is  $1.3 \times 10^{-5}$ , the original length is 25 meters, and of course the change of temperature is  $36^\circ\text{C}$ . This results in an expansion of .12 m or 12 cm.<sup>11,12</sup>

Now consider 40 pieces of steel 25 meters long laid end to end to make a 1 km long bridge. The bridge's length will change roughly 480 cm between the winter and summer! Fortunately, expansion joints are built into bridges allowing for this expansion, ensuring bridges are safe in all seasons

A bimetallic strip takes advantage of the thermal expansion effect to generate motion.

Two dissimilar strips of metal are joined together along their entire lengths. When heat is applied, the bimetallic strip bends in the direction of the metal with the smaller coefficient of thermal expansion, (see the figure below).

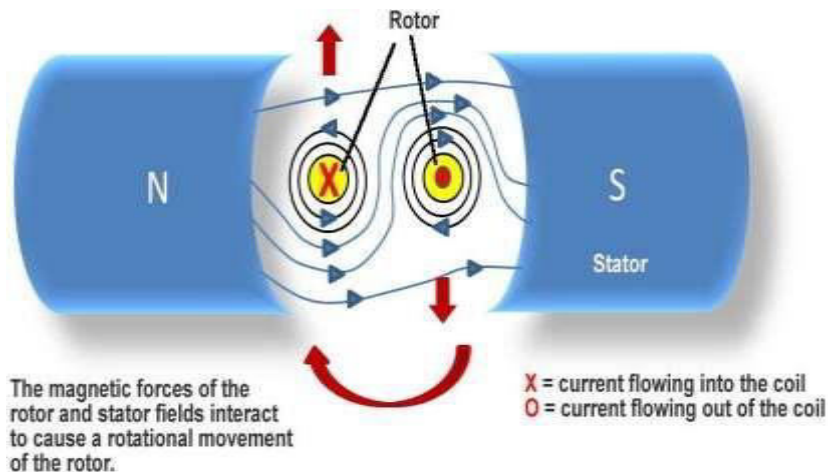
Bimetallic strips have many uses. One common use is in thermostats used to control the temperature in homes and offices. At the microscale, bimetallic actuators are used in microthermostats and as microvalves.



Schematic showing how a bimetallic strip works. This particular bimetallic strip is being used as a

thermostat. a) Two dissimilar strips of metal are used that have different coefficients of thermal expansion, b) The two strips of metal are joined along their entire interface at some temperature ( $T_1$ ), c) When the temperature increases temperature,  $T_2$ , the bimetallic strips deflect enough to touch the upper contact and allow a current to flow in the bimetallic strip turning on the air conditioner.

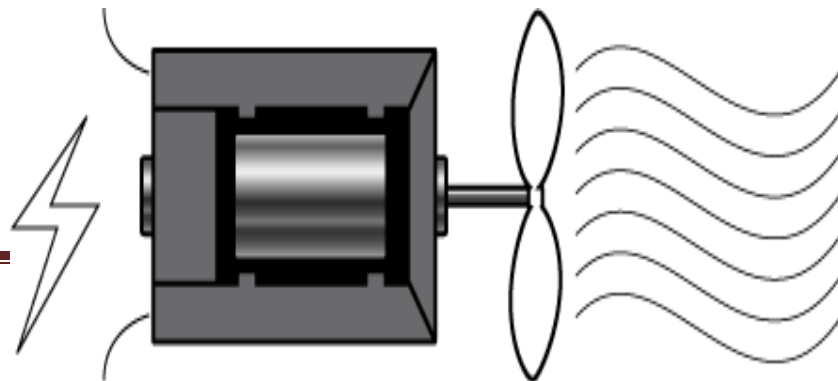
## Electric Actuators



An electric motor is a type of an electric actuator (see graphic). Most direct current (DC) motors operate by current flowing through a coil of wire and creating a magnetic field around the coil.

The coil is wrapped around the motor's shaft and is positioned between the poles of a large permanent magnet or electromagnet. The interaction of the two magnetic fields causes the coil to rotate on its axis, rotating the motor's shaft (see figure above).

Thus, an electric motor is a transducer AND an actuator because it converts electrical energy to magnetic energy to mechanical energy or motion



An electric motor is an actuator that transforms electrical energy into mechanical energy or motion

## Mechanical Actuators

Mechanical actuators convert a mechanical input (usually rotary) into linear motion.

A common example of a mechanical actuator is a screw jack.

The figure below shows a screw jack in operation. Rotation of the screw causes the legs of the jack to move apart or move together.

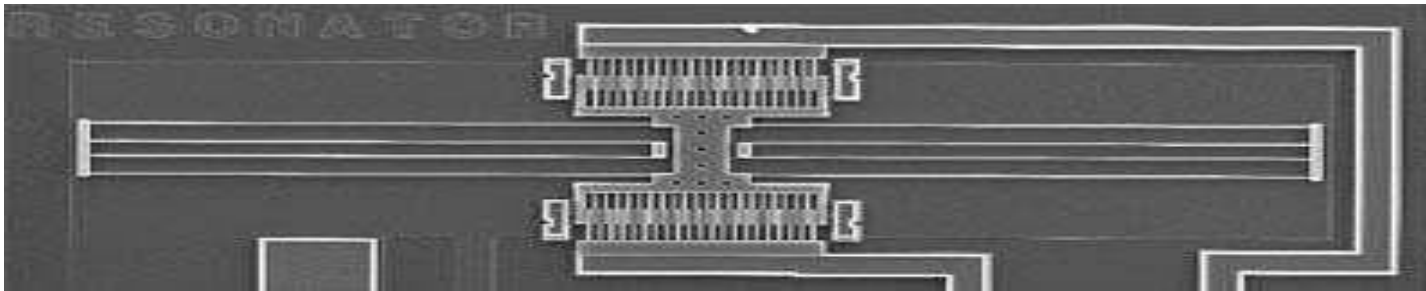
Inspecting the motion of the top point of the jack, this mechanical rotational input is clearly converted into linear mechanical motion.

Mechanical actuators can produce a rotational output with the proper gearing mechanism.



A screw jack converting rotational energy into linear motion (to lift a car possibly)

An example of a microdevice which acts as an actuator is the electrostatic combdrive. These combdrives are used in many MEMS applications such as resonators, microengines, and gyroscopes. The force generated is low, usually less than 50  $\mu$ N. However, these devices are predictable and reliable making them highly used.



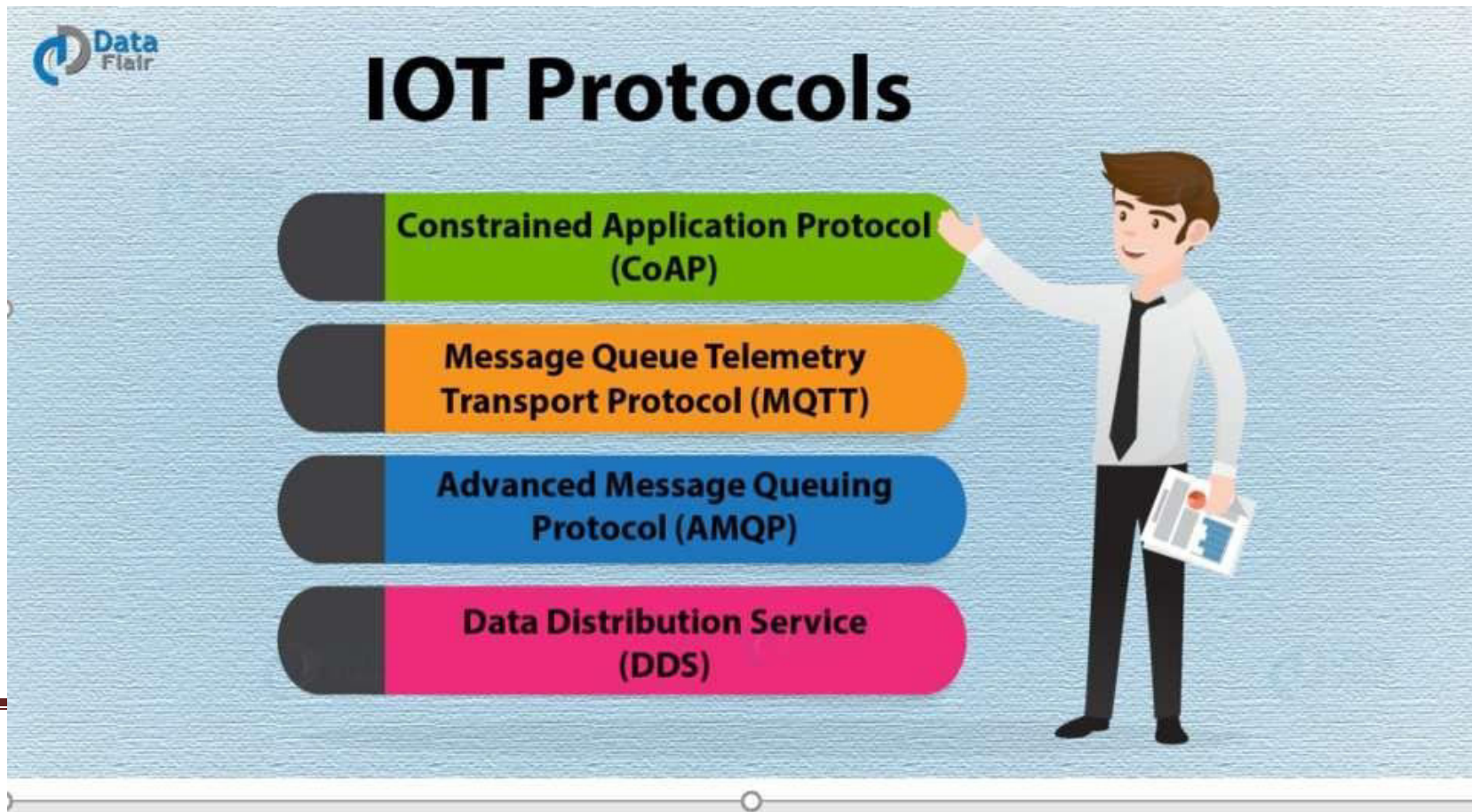
SEM of a typical comb-drive resonator

The image above is an example of a MEMS electrostatic combdrive resonator. A resonator is a device which naturally oscillates at its resonance frequencies. The oscillations in a resonator can either be electromagnetic or mechanical (i.e acoustic). Resonators are used to generate waves of desired frequencies or to extract specific frequencies from a given signal.

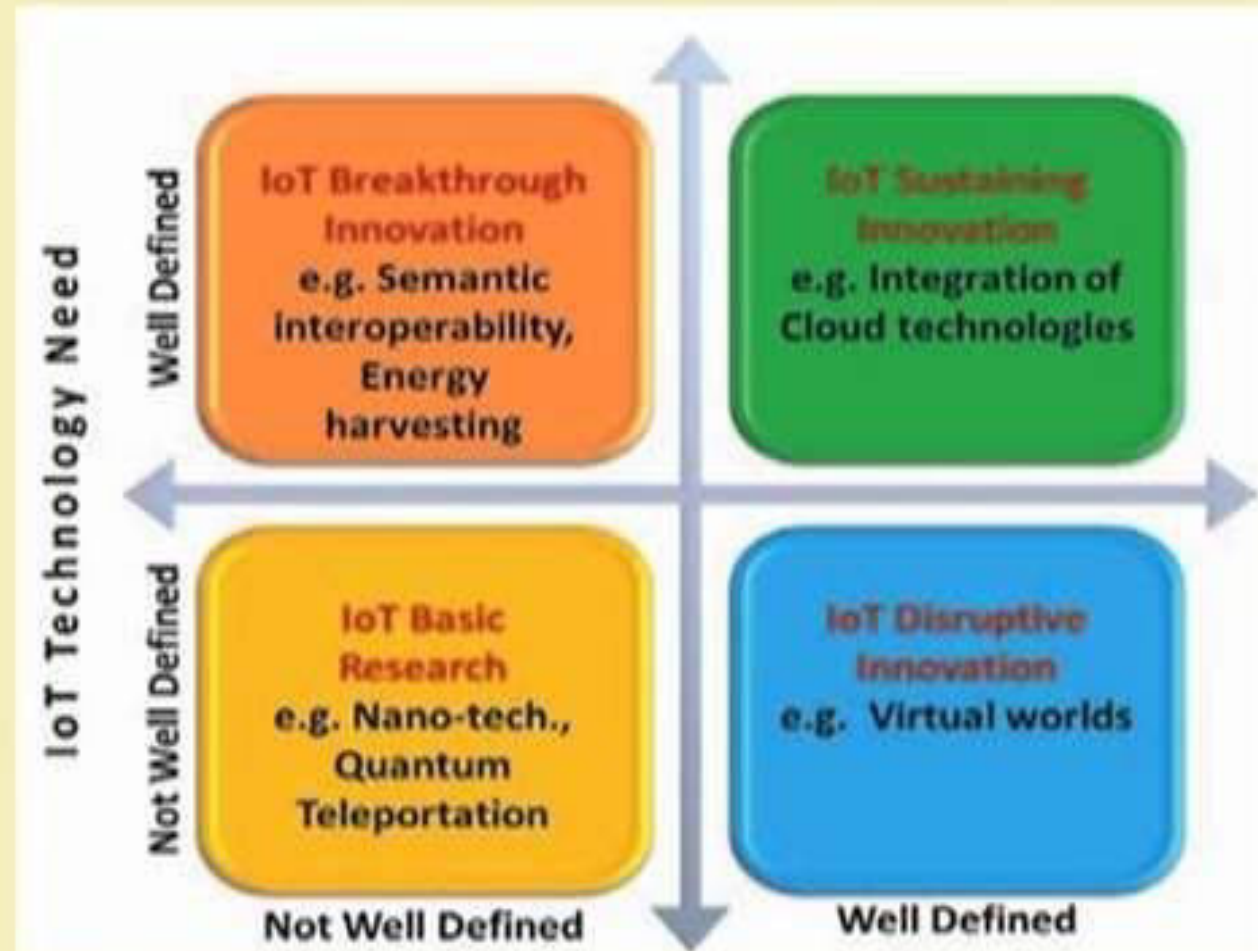


# UNIT-1 PART 2

# Basics of Networking



# Convergence of Domains



1)**IOT Basic research**, there has been a lot of research on the nanotechnology, the use of nanotechnology, the use of quantum teleportation. **Quantum teleportation** basically means that how the different information at the atomic level is sent from one point to another. So, it is transported from one point to another at the atomic level and nanotechnology, it involves things like nano IoT, nano nodes, nano networking, nodes, nano sensor nodes and nano networks. So, like this at the nanoscale and for quantum communication, there has been lot of advertisements that has been done for involving basic innovations, basic research innovations.

2)**IOT Break through Innovation**: Eg: Semantic Interoperability, Energy harvesting.

**Semantic Interoperability** : There has been lot of research on semantic in for interoperability. For example, let us see that a temperature sensor, it might be given the data as temp, another temperature sensor as temperature, another temperature sensor the third one. So, there has to be interoperability between all these different collisions, but they are all different to the same temperature, right. So, this is

basically taken care of by things like semantic interoperability

## IoT Components

Device (The Thing)

Local Network

Internet

Backend Services

Applications

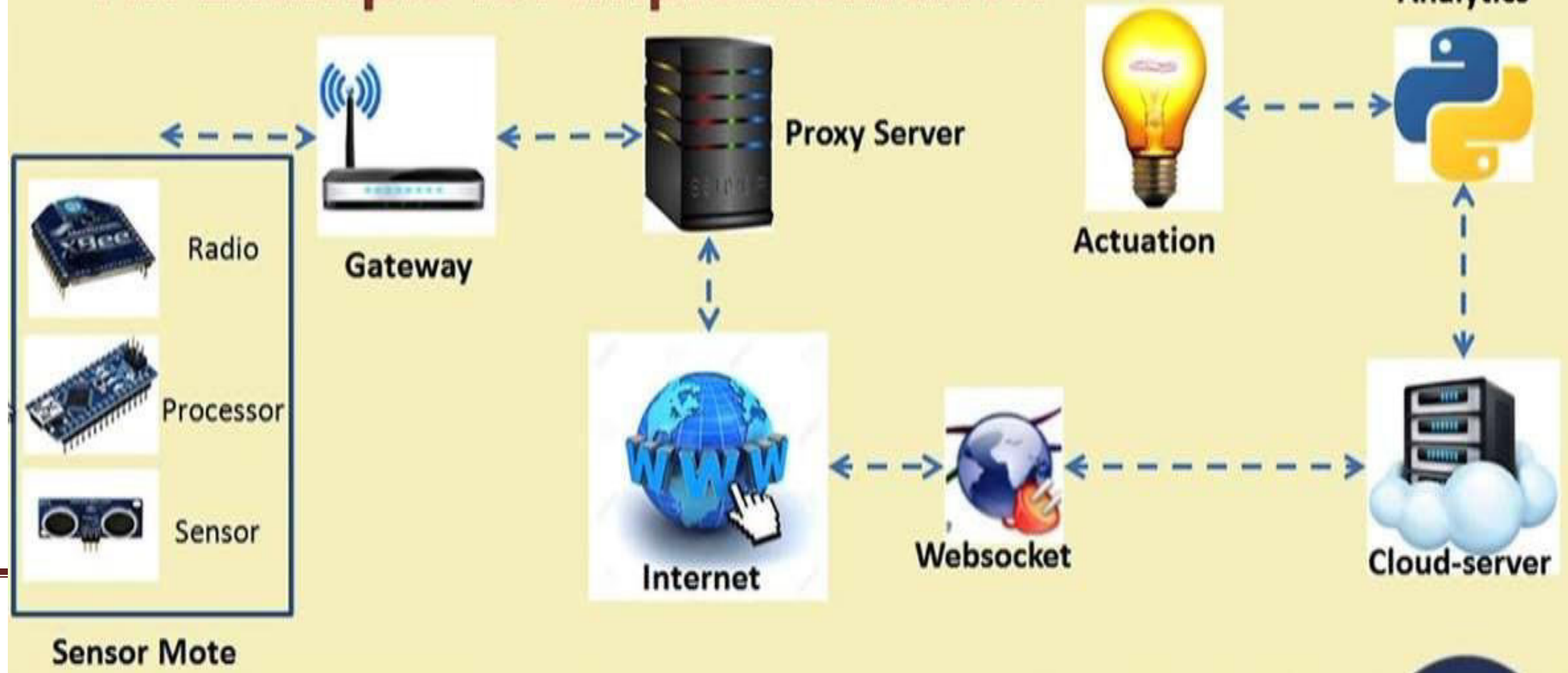




## Functional Components of IoT

- ✓ Component for interaction and communication with other IoT devices
- ✓ Component for processing and analysis of operations
- ✓ Component for Internet interaction
- ✓ Components for handling Web services of applications
- ✓ Component to integrate application services
- ✓ User interface to access IoT

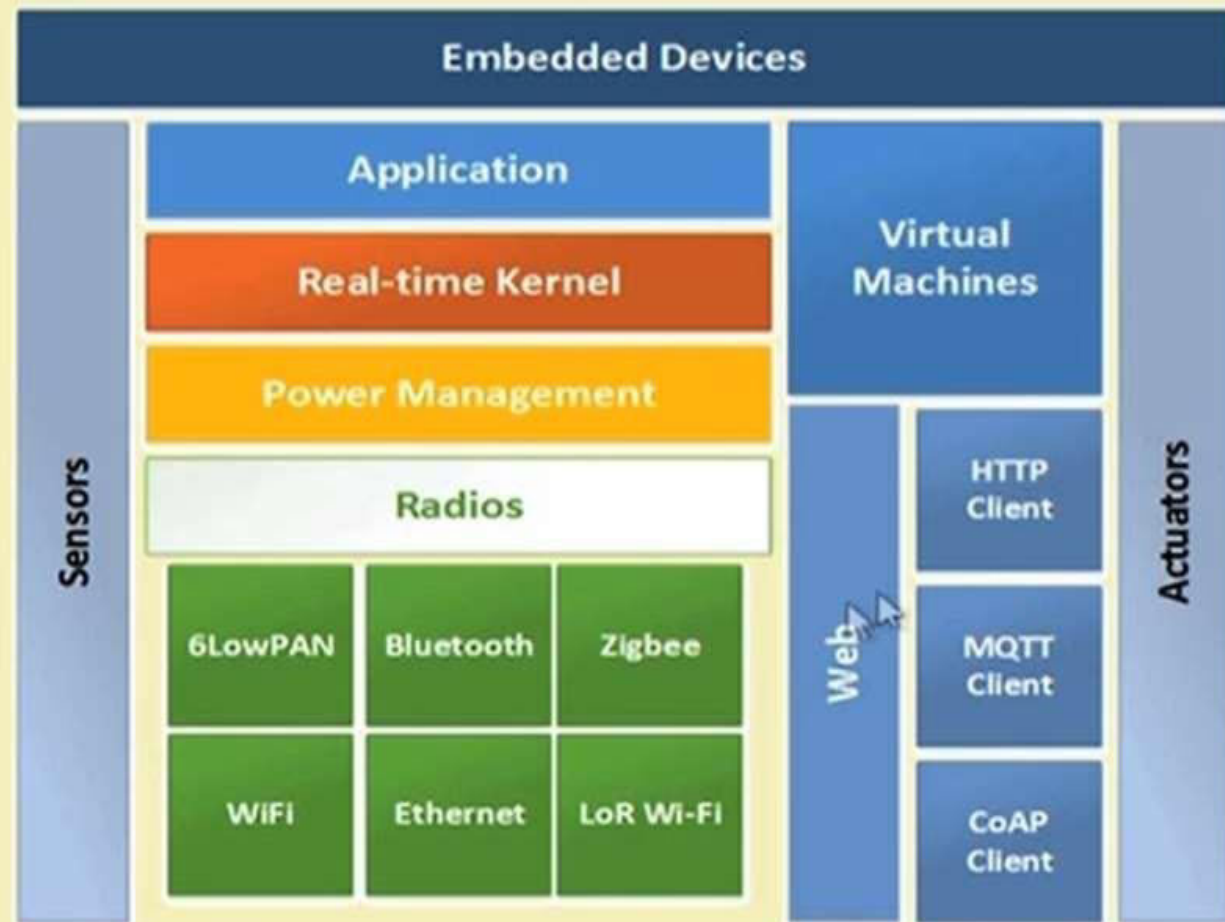
## An Example IoT Implementation



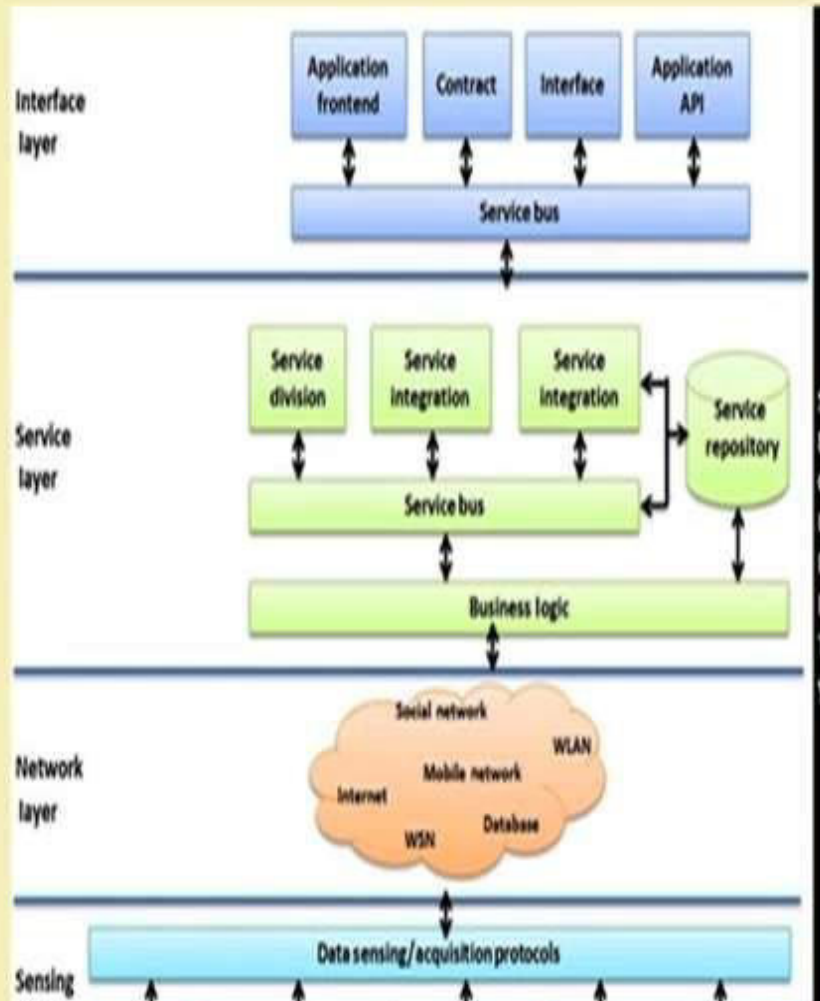




# IoT Interdependencies



# IoT Service Oriented Architecture



## IoT Categories

### ✓ Industrial IoT

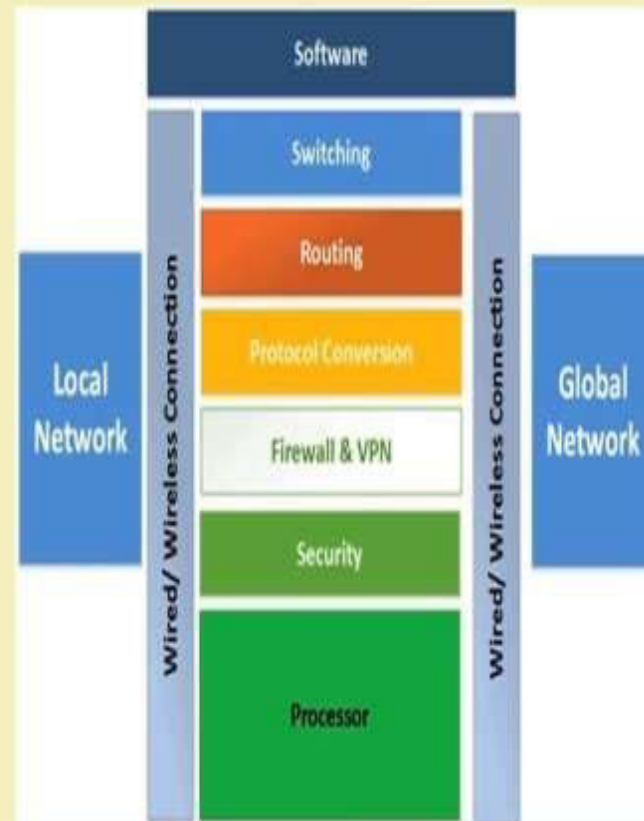
- IoT device connects to an IP network and the global Internet.
- Communication between the nodes done using regular as well as industry specific technologies.

### ✓ Consumer IoT

- IoT device communicates within the locally networked devices.
- Local communication is done mainly via Bluetooth, Zigbee or WiFi.
- Generally limited to local communication by a Gateway.



## IoT Gateways



## IoT and Associated Technologies





# Technical Deviations from Regular Web







## IoT Challenges

- ✓ Security
- ✓ Scalability
- ✓ Energy efficiency
- ✓ Bandwidth management
- ✓ Modeling and Analysis
- ✓ Interfacing
- ✓ Interoperability
- ✓ Data storage
- ✓ Data Analytics
- ✓ Complexity management (e.g., SDN)

## Considerations

- ✓ Communication between the IoT device(s) and the outside world dictates the network architecture.
- ✓ Choice of communication technology dictates the IoT device hardware requirements and costs.
- ✓ Due to the presence of numerous applications of IoT enabled devices, a single networking paradigm not sufficient to address all the needs of the consumer or the IoT device.

## Complexity of Networks

- ✓ Growth of networks
- ✓ Interference among devices
- ✓ Network management
- ✓ Heterogeneity in networks
- ✓ Protocol standardization within networks

## Wireless Networks

- Traffic and load management
- Variations in wireless networks – Wireless Body Area Networks and other Personal Area Networks
- Interoperability
- Network management
- Overlay networks

## **Scalability**

- Flexibility within Internet
- IoT integration
- Large scale deployment
- Real-time connectivity of billions of devices

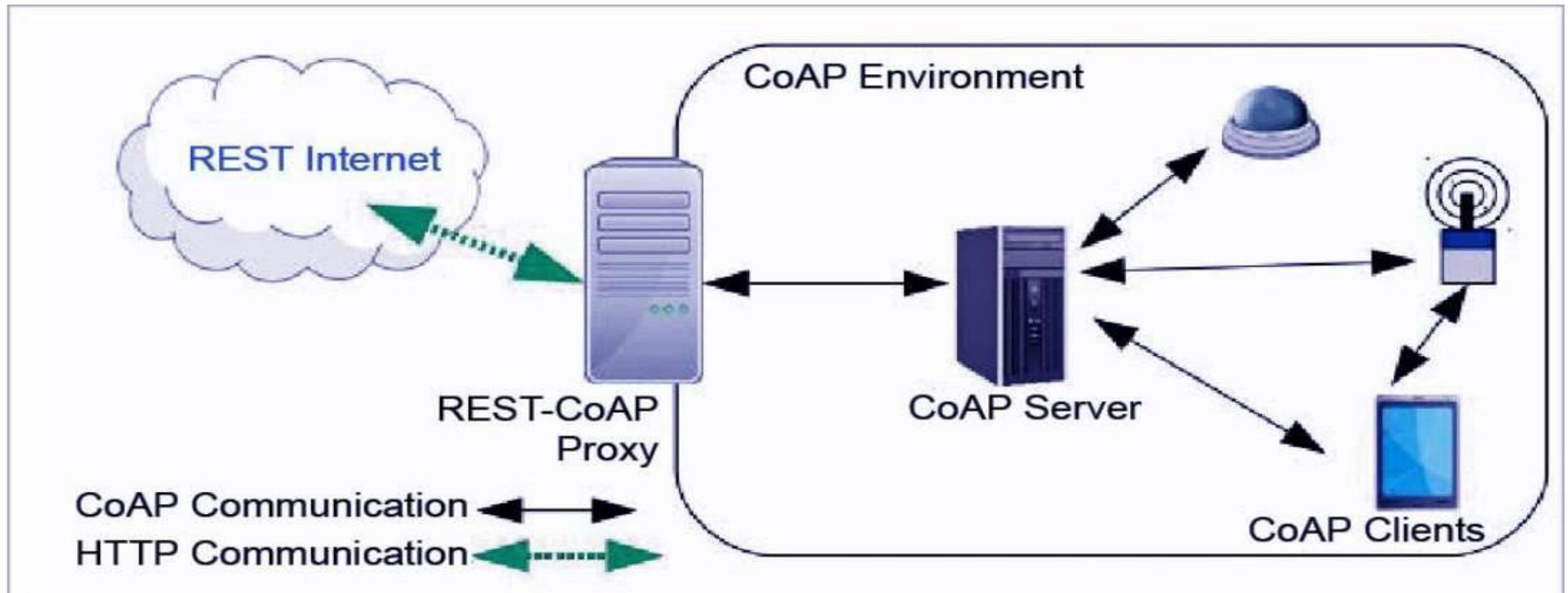


## Functionality-based IoT Protocol Organization

- ✓ **Connectivity** (6LowPAN, RPL)
- ✓ **Identification** (EPC, uCode, IPv6, URIs)
- ✓ **Communication / Transport** (WiFi, Bluetooth, LPWAN)
- ✓ **Discovery** (Physical Web, mDNS, DNS-SD)
- ✓ **Data Protocols** (MQTT, CoAP, AMQP, Websocket, Node)
- ✓ **Device Management** (TR-069, OMA-DM)
- ✓ **Semantic** (JSON-LD, Web Thing Model)
- ✓ **Multi-layer Frameworks** (Alljoyn, IoTivity, Weave, Homekit)

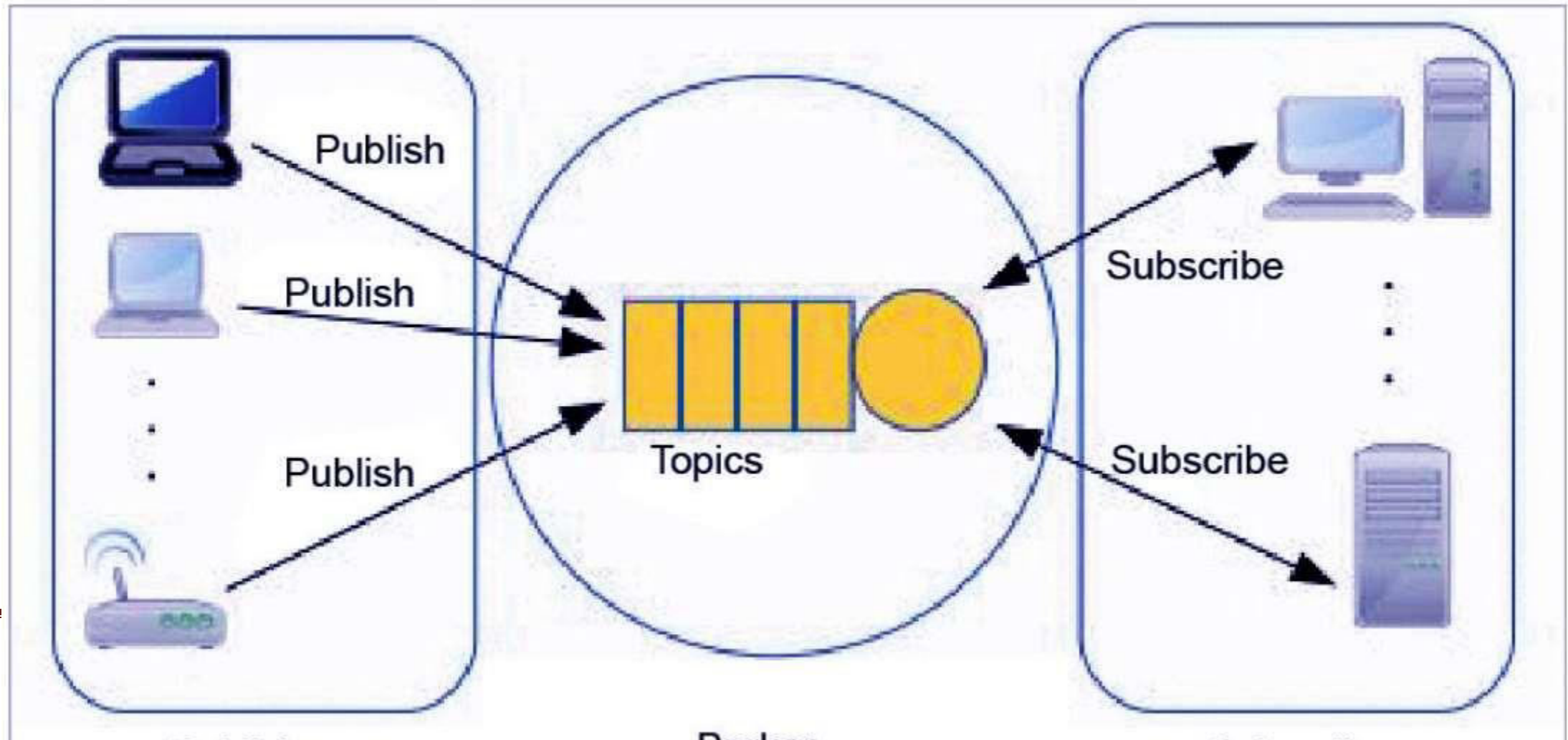
## 1. Constrained Application Protocol (CoAP)

CoAP is an internet utility protocol for constrained gadgets. It is designed to enable simple, constrained devices to join IoT through constrained networks having low bandwidth availability. This protocol is primarily used for machine-to-machine (M2M) communication and is particularly designed for IoT systems that are based on HTTP protocols.



## FUNDAMENTALS OF INTERNET OF THINGS ( EC32110E)

CoAP makes use of the UDP protocol for lightweight implementation. It also uses restful architecture, which is just like the HTTP protocol. It makes use of dtls for the cozy switch of statistics within the slipping layer.





## Introduction

- ✓ CoAP – **Constrained Application Protocol**.
- ✓ **Web transfer protocol** for use with constrained nodes and networks.
- ✓ **Designed for Machine to Machine (M2M)** applications such as smart energy and building automation.
- ✓ Based on **Request-Response model** between end-points
- ✓ Client-Server interaction is **asynchronous over a datagram oriented transport protocol** such as UDP

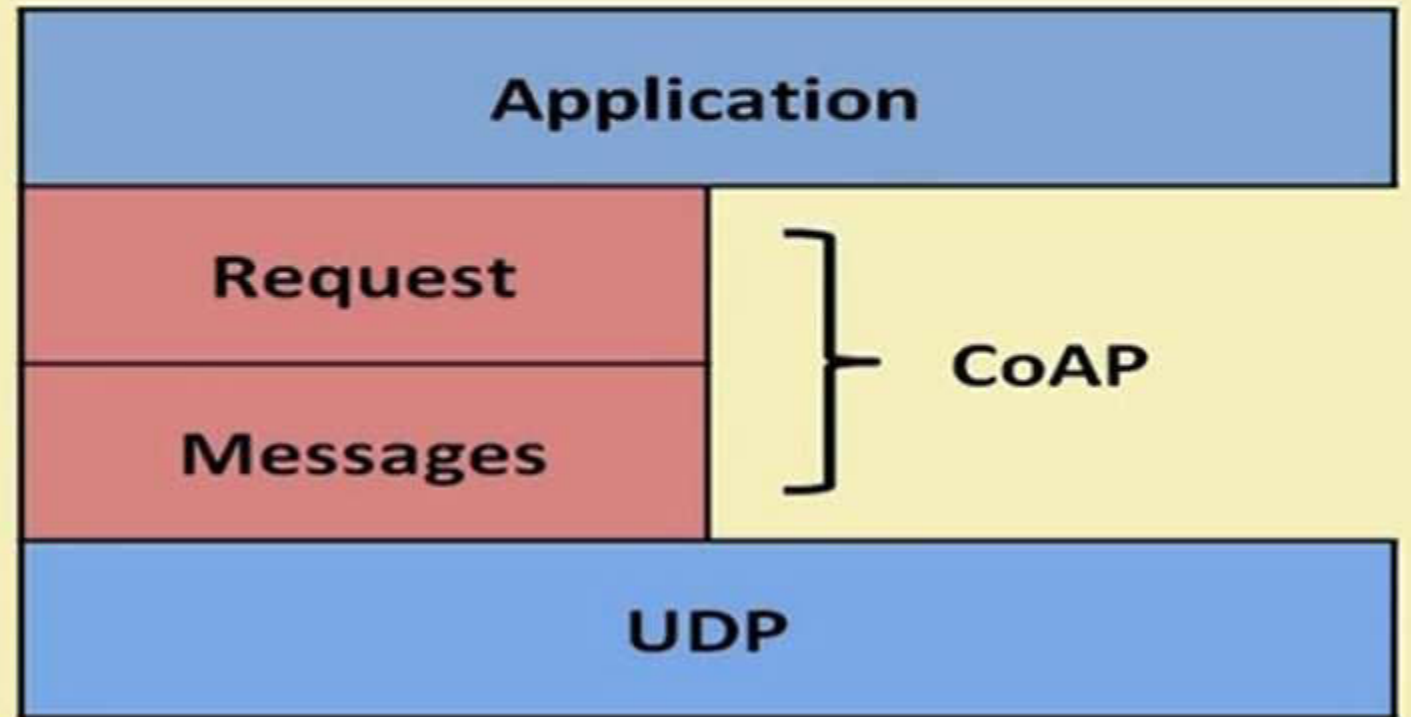


- ✓ The Constrained Application Protocol (CoAP) is a session layer protocol designed by IETF Constrained RESTful Environment (CoRE) working group to provide lightweight RESTful (HTTP) interface.
- ✓ Representational State Transfer (REST) is the standard interface between HTTP client and servers.
- ✓ Lightweight applications such as those in IoT, could result in significant overhead and power consumption by REST.
- ✓ CoAP is designed to enable low-power sensors to use RESTful services while meeting their power constraints.

- ✓ Built over UDP, instead of TCP (which is commonly used with HTTP) and has a light mechanism to provide reliability.
- ✓ CoAP architecture is divided into two main sub-layers:
  - Messaging
  - Request/response.
- ✓ The messaging sub-layer is responsible for reliability and duplication of messages, while the request/response sub-layer is responsible for communication.
- ✓ CoAP has four messaging modes:
  - Confirmable
  - Non-confirmable
  - Piggyback



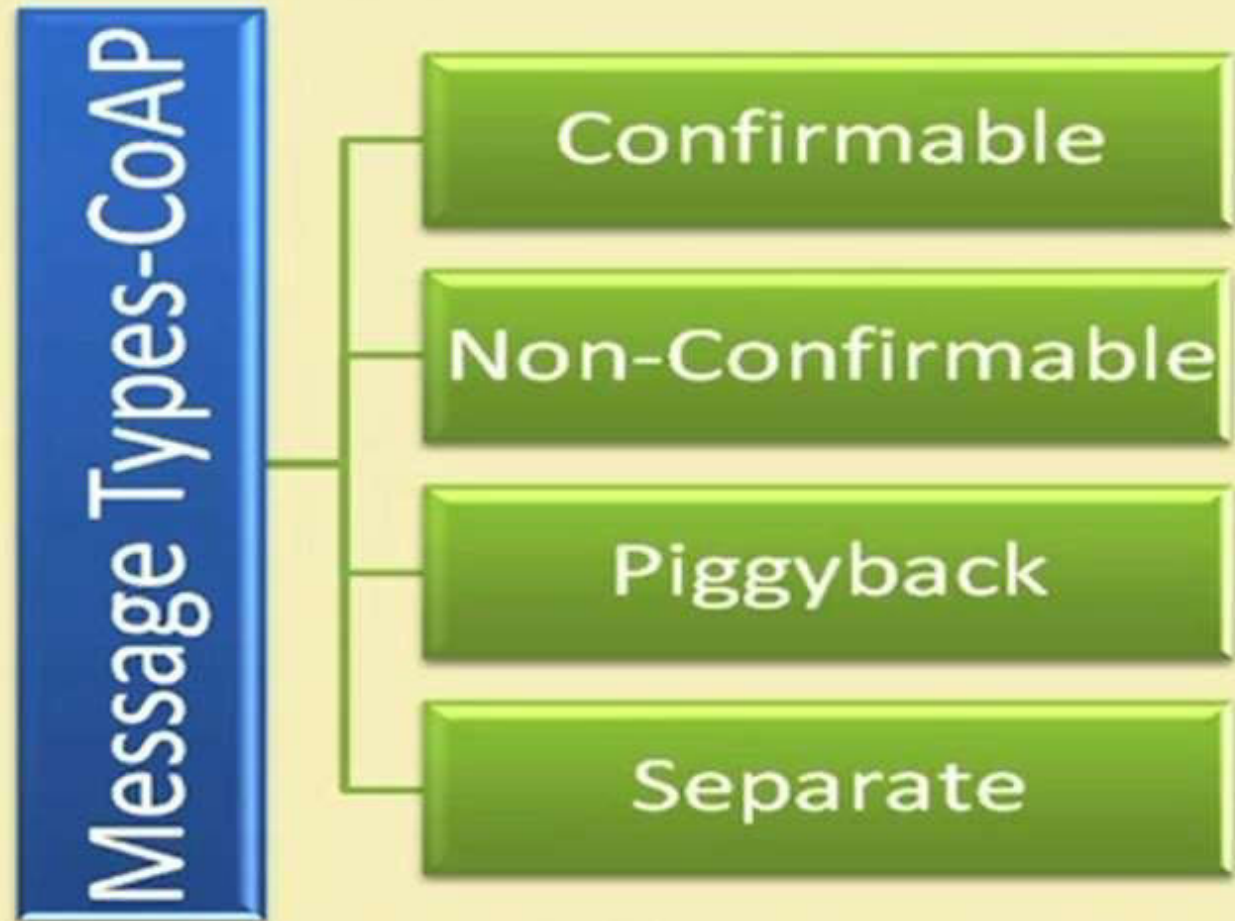
# CoAP Position



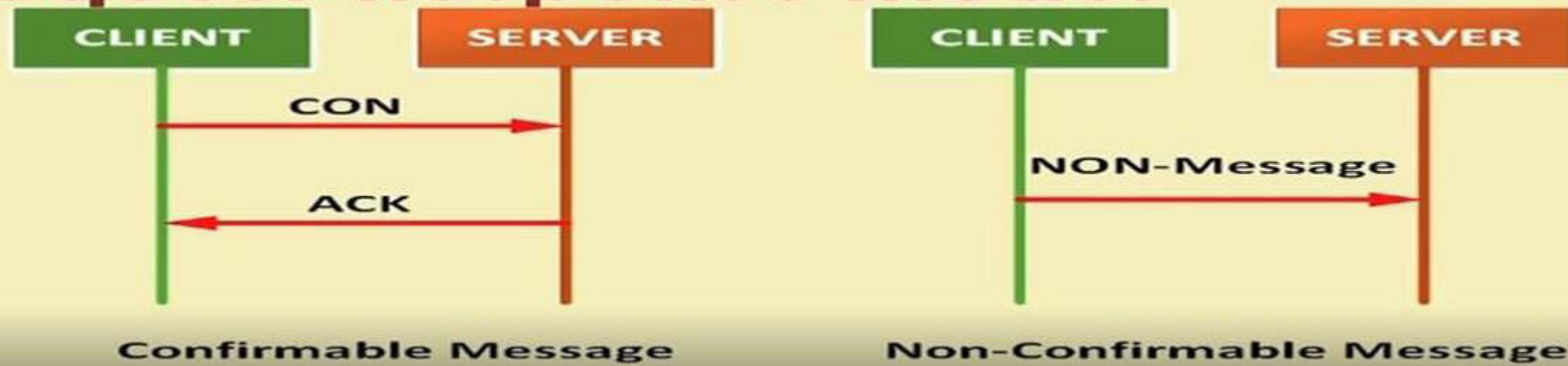




# CoAP Message Types



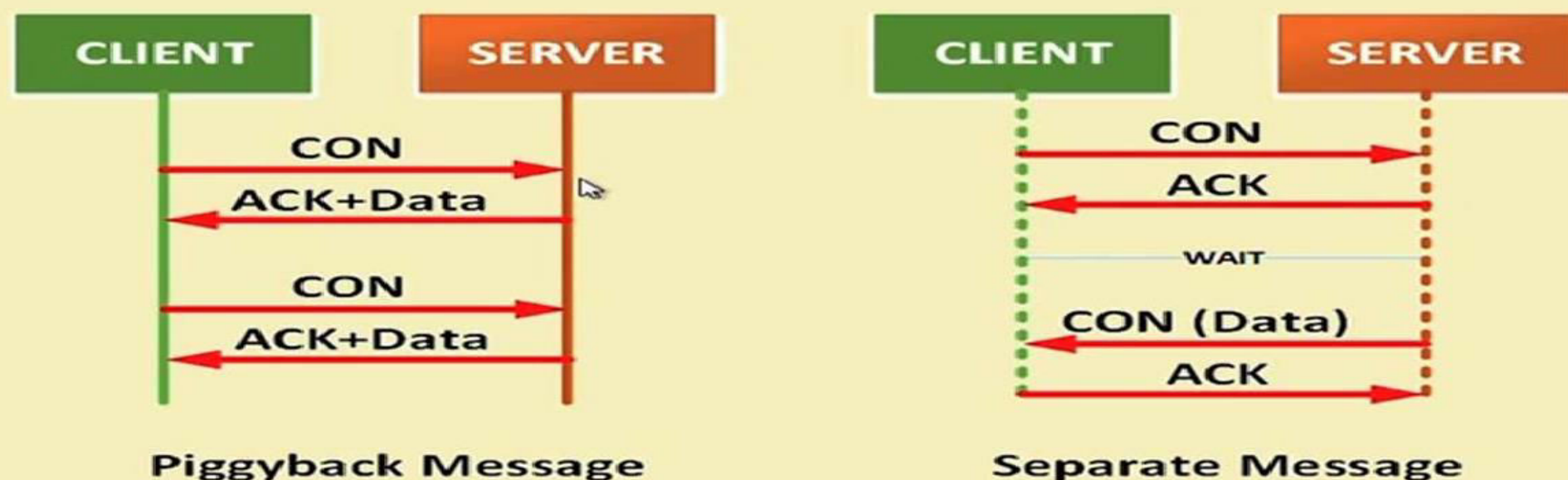
# CoAP Request-Response Model



- ✓ Confirmable and non-confirmable modes represent the reliable and unreliable transmissions, respectively, while the other modes are used for request/response.
- ✓ Piggyback is used for client/server direct communication where the server sends its response directly after receiving the message, i.e., within the acknowledgment message.
- ✓ On the other hand, the separate mode is used when the server response comes in a message separate from the acknowledgment, and may take some time to be sent by the server.
- ✓ Similar to HTTP, CoAP utilizes GET, PUT, PUSH, DELETE messages requests to retrieve, create, update, and delete, respectively



# CoAP Request-Response Model



## Features

- ✓ Reduced overheads and parsing complexity.
- ✓ URL and content-type support.
- ✓ Support for the discovery of resources provided by known CoAP services.
- ✓ Simple subscription for a resource, and resulting push notifications.
- ✓ Simple caching based on maximum message age.



# xmpp

## Introduction

- ✓ **XMPP – Extensible Messaging and Presence Protocol.**
- ✓ A communication protocol for **message-oriented middleware** based on XML (Extensible Markup Language).
- ✓ Real-time exchange of structured data.
- ✓ It is an open standard protocol.



- ✓ XMPP uses a **client-server architecture**.
- ✓ As the model is **decentralized**, no central server is required.
- ✓ XMPP provides for the **discovery of services** residing locally or across a network, and the **availability information** of these services.
- ✓ Well-suited for cloud computing where virtual machines, networks, and firewalls would otherwise present obstacles to alternative service discovery and presence-based solutions.
- ✓ Open means to support machine-to-machine or peer-to-peer communications across a diverse set of networks.

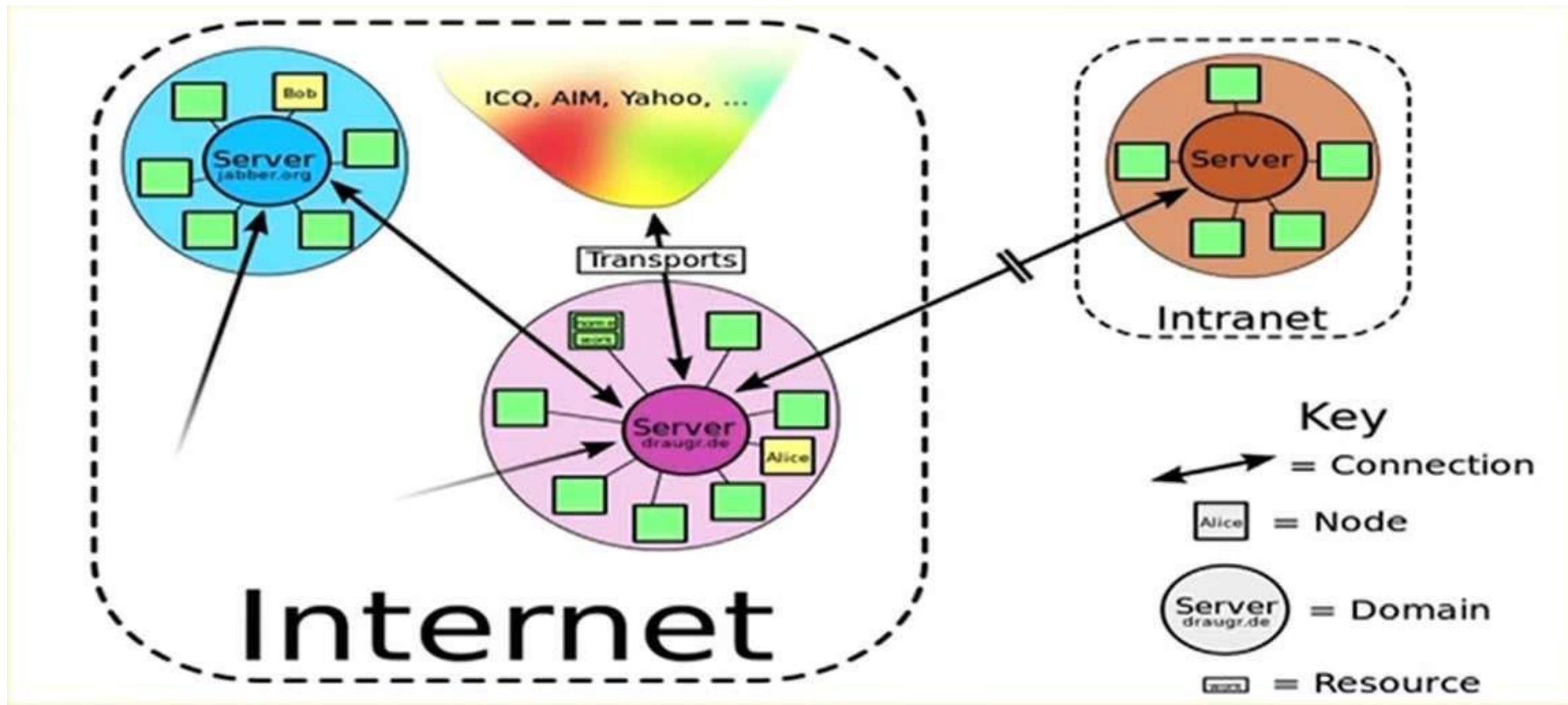


## Highlights

- ✓ Decentralization – No central server; anyone can run their own XMPP server.
- ✓ Open standards – No royalties or granted permissions are required to implement these specifications
- ✓ Security – Authentication, encryption, etc.
- ✓ Flexibility – Supports interoperability







# Core XMPP Technologies

## Core

- information about the core XMPP technologies for XML streaming

## Jingle

- multimedia signalling for voice, video, file transfer

## Multi-user Chat

- flexible, multi-party communication

## PubSub

- alerts and notifications for data syndication

## BOSH

- HTTP binding for XMPP

# Weaknesses

- ✓ Does not support QoS.
- ✓ Text based communications induces higher network overheads.
- ✓ Binary data must be first encoded to **base64** before transmission.





# Applications

- ✓ Publish-subscribe systems
- ✓ Signaling for VoIP
- ✓ Video
- ✓ File transfer
- ✓ Gaming
- ✓ Internet of Things applications
  - Smart grid
  - Social networking services

### 2. Message Queue Telemetry Transport Protocol (MQTT)

MQTT (*Message Queue Telemetry Transport*) is a messaging protocol developed with the aid of Andy Stanford-Clark of IBM and Arlen Nipper of Arcom in 1999 and is designed for M2M communication. It's normally used for faraway tracking in IoT. Its primary challenge is to gather statistics from many gadgets and delivery of its infrastructure. MQTT connects gadgets and networks with packages and middleware. All the devices hook up with facts concentrator servers like IBM's new message sight appliance. MQTT protocols paintings on top of TCP to offer easy and dependable streams of information.

These IoT protocols include 3 foremost additives: subscriber, publisher, and dealer. The writer generates the information and transmits the facts to subscribers through the dealer. The dealer guarantees safety by means of move-checking the authorization of publishers and subscribers.

## Introduction

- ✓ **Message Queue Telemetry Transport.**
- ✓ ISO standard (ISO/IEC PRF 20922).
- ✓ It is a publish-subscribe-based lightweight messaging protocol for use in conjunction with the TCP/IP protocol.
- ✓ MQTT was introduced by IBM in 1999 and standardized by OASIS in 2013.
- ✓ Designed to provide connectivity (mostly embedded) between applications and middle-wares on one side and networks and communications on the other side.

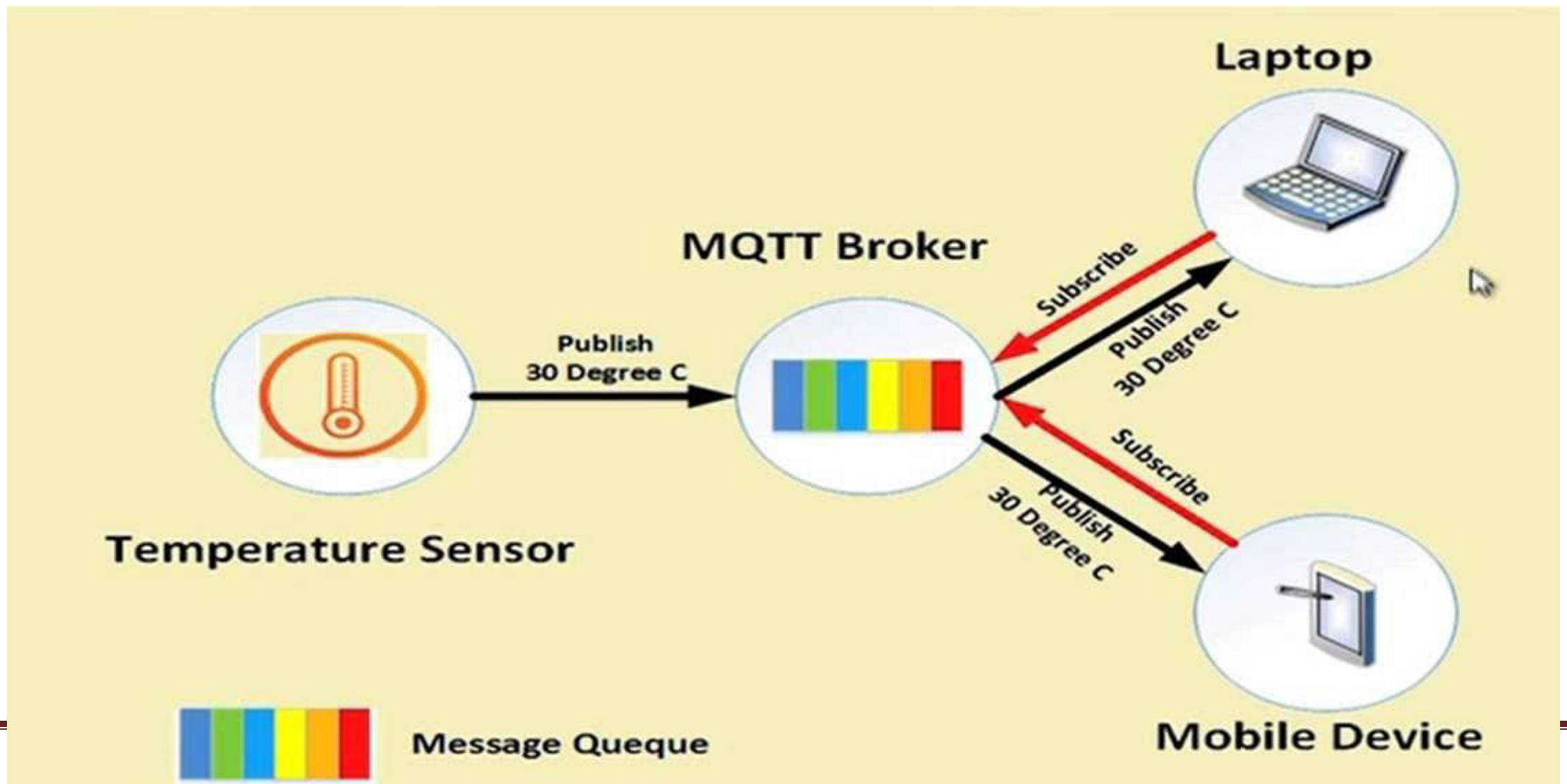


- ✓ A message broker controls the publish-subscribe messaging pattern.
- ✓ A topic to which a client is subscribed is updated in the form of messages and distributed by the message broker.
- ✓ Designed for:
  - Remote connections
  - Limited bandwidth
  - Small-code footprint

## MQTT Components







# Communication

- ✓ The protocol uses a **publish/subscribe** architecture (HTTP uses a request/response paradigm).
- ✓ Publish/subscribe is **event-driven** and enables messages to be pushed to clients.
- ✓ The central **communication point is the MQTT broker**, which is in charge of dispatching all messages between the senders and the rightful receivers.
- ✓ Each client that publishes a message to the broker, includes a **topic** into the message. The **topic is the routing information for the broker**



- ✓ Each client that wants to receive messages subscribes to a certain topic and the broker delivers all messages with the matching topic to the client.
- ✓ Therefore the clients don't have to know each other. They only communicate over the topic.
- ✓ This architecture enables highly scalable solutions without dependencies between the data producers and the data consumers.



## MQTT Topics

- ✓ A topic is a **simple string** that can have more hierarchy levels, which are separated by a slash.
- ✓ A sample topic for sending temperature data of the living room could be *house/living-room/temperature*.
- ✓ On one hand the client (e.g. mobile device) can subscribe to the exact topic or on the other hand, it can use a **wildcard**.



- ✓ The subscription to *house/+/temperature* would result in all messages sent to the previously mentioned topic *house/living-room/temperature*, as well as any topic with an arbitrary value in the place of living room, such as *house/kitchen/temperature*.
- ✓ The plus sign is a **single level wild card** and only allows arbitrary values for one hierarchy.
- ✓ If more than one level needs to be subscribed, such as, the entire sub-tree, there is also a **multilevel wildcard** (#).
- ✓ It allows to subscribe to all underlying hierarchy levels.
- ✓ For example *house/#* is subscribing to all topics beginning with *house*.

# Applications

- ✓ **Facebook Messenger** uses MQTT for online chat.
- ✓ **Amazon Web Services** use Amazon IoT with MQTT.
- ✓ **Microsoft Azure** IoT Hub uses MQTT as its main protocol for telemetry messages.
- ✓ The **EVERYTHING IoT platform** uses MQTT as an M2M protocol for millions of connected products.
- ✓ **Adafruit** launched a free MQTT cloud service for IoT experimenters called Adafruit IO.





## SMQTT

- ✓ **Secure MQTT** is an extension of MQTT which uses encryption based on lightweight attribute based encryption.
- ✓ The main advantage of using such encryption is the broadcast encryption feature, in which one message is encrypted and delivered to multiple other nodes, which is quite common in IoT applications.
- ✓ In general, the algorithm consists of four main stages: setup, encryption, publish and decryption.





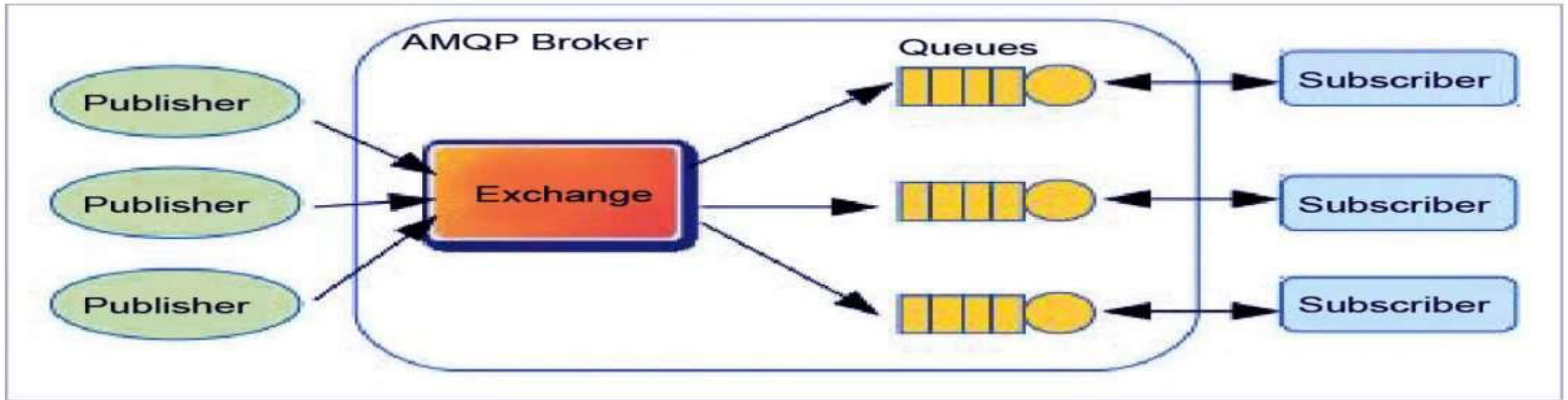
- ✓ In the setup phase, the subscribers and publishers register themselves to the broker and get a master secret key according to their developer's choice of key generation algorithm.
- ✓ When the data is published, it is encrypted and published by the broker which sends it to the subscribers, which is finally decrypted at the subscriber end having the same master secret key.
- ✓ The key generation and encryption algorithms are not standardized.
- ✓ SMQTT is proposed only to enhance MQTT security features.

### 3. Advanced Message Queuing Protocol (AMQP)

This was evolved by John O'Hara at JP Morgan Chase in London. AMQP is a software layer protocol for message-oriented middleware environment. It supports reliable verbal exchange through message transport warranty primitives like at-most-once, at least once and exactly as soon as shipping.

The AMQP – IoT protocols consist of hard and fast components that route and save messages within a broker carrier, with a set of policies for wiring the components together. The AMQP protocol enables patron programs to talk to the dealer and engage with the [AMQP model](#). This version has the following three additives, which might link into processing chains in the server to create the favored capabilities.

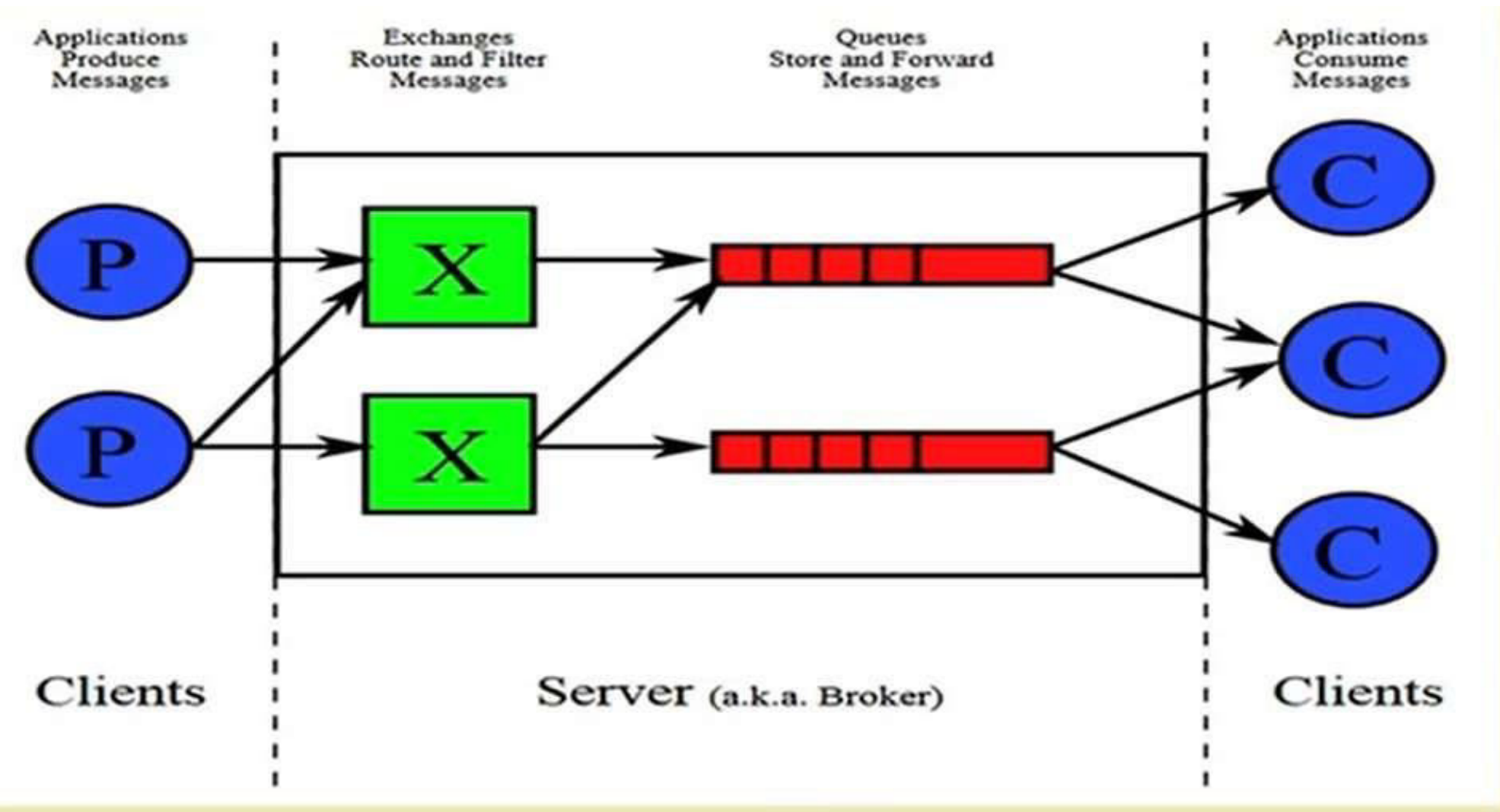
- **Exchange:** Receives messages from publisher primarily based programs and routes them to 'message queues'.
  - **Message Queue:** Stores messages until they may thoroughly process via the eating client software.
  - **Binding:** States the connection between the message queue and the change.
-





## Introduction

- ✓ **Advanced Message Queuing Protocol.**
- ✓ **Open standard for passing business messages** between applications or organizations.
- ✓ Connects between systems and business processes.
- ✓ It is a binary application layer protocol.
- ✓ Basic unit of data is a *frame*.
- ✓ ISO standard: **ISO/IEC 19464**



## AMQP Features



## Features



## AMQP Frame Types

- ✓ Nine AMQP frame types are defined that are used to initiate, control and tear down the transfer of messages between two peers:
  - Open (connection open)
  - Begin (session open)
  - Attach (initiate new link)
  - Transfer (for sending actual messages)
  - Flow (controls message flow rate)
  - Disposition (Informs the changes in state of transfer)
  - Detach (terminate the link)
  - End (session close)
  - Close (connection close)

## Message Delivery Guarantees

- ✓ *At-most-once*
  - each message is delivered once or never
- ✓ *At-least-once*
  - each message is certain to be delivered, but may do so multiple times
- ✓ *Exactly-once*
  - message will always certainly arrive and do so only once



## Components

**Exchange**

- Part of Broker
- Receives messages and routes them to Queues

**Queue**

- Separate queues for separate business processes
- Consumers receive messages from queues

**Bindings**

- Rules for distributing messages (who can access what message, destination of the message)

## AMQP Exchanges

**Direct**

**Fan-out**

**Topic**

**Header**



## **AMQP Features**

- ✓ Targeted QoS (Selectively offering QoS to links)
- ✓ Persistence (Message delivery guarantees)
- ✓ Delivery of messages to multiple consumers
- ✓ Possibility of ensuring multiple consumption
- ✓ Possibility of preventing multiple consumption
- ✓ High speed protocol

## **Applications**

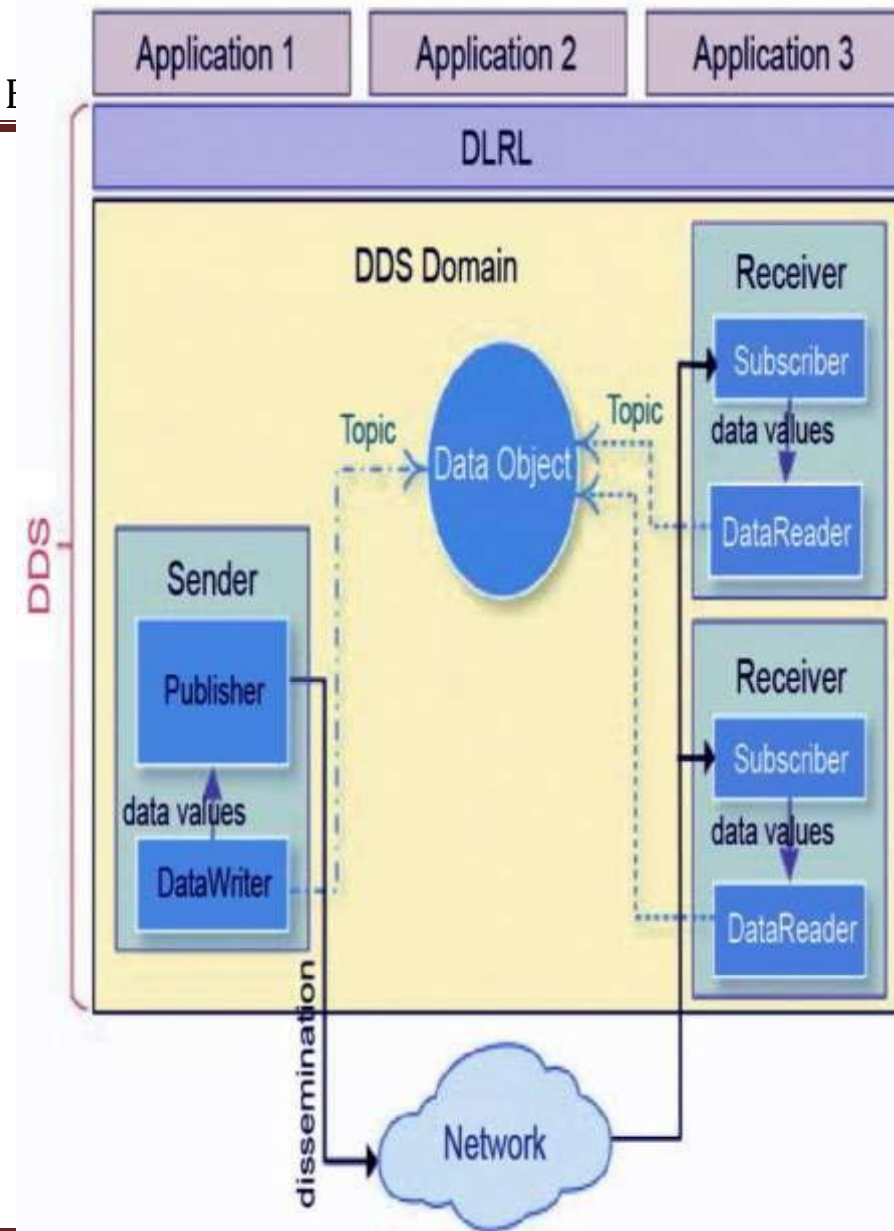
- ✓ Monitoring and global update sharing.
- ✓ Connecting different systems and processes to talk to each other.
- ✓ Allowing servers to respond to immediate requests quickly and delegate time consuming tasks for later processing.
- ✓ Distributing a message to multiple recipients for consumption.
- ✓ Enabling offline clients to fetch data at a later time.
- ✓ Introducing fully asynchronous functionality for systems.
- ✓ Increasing reliability and uptime of application deployments.

## 4. Data Distribution Service (DDS)

It enables a scalable, real-time, reliable, excessive-overall performance and interoperable statistics change via the submit-subscribe technique. DDS makes use of multicasting to convey high-quality QoS to applications.

DDS is deployed in platforms ranging from low-footprint devices to the cloud and supports green bandwidth usage in addition to the agile orchestration of system additives.

The DDS – IoT protocols have fundamental layers: facts centric submit-subscribe (dcps) and statistics-local reconstruction layer (dlrl). Dcps plays the task of handing over the facts to subscribers, and the dlrl layer presents an interface to dcps functionalities, permitting the sharing of distributed data amongst IoT enabled objects.



# UNIT-1 PART 2 (CONT)

# xmpp

## Introduction

- ✓ **XMPP – Extensible Messaging and Presence Protocol.**
- ✓ A communication protocol for **message-oriented middleware** based on XML (Extensible Markup Language).
- ✓ Real-time exchange of structured data.
- ✓ It is an open standard protocol.





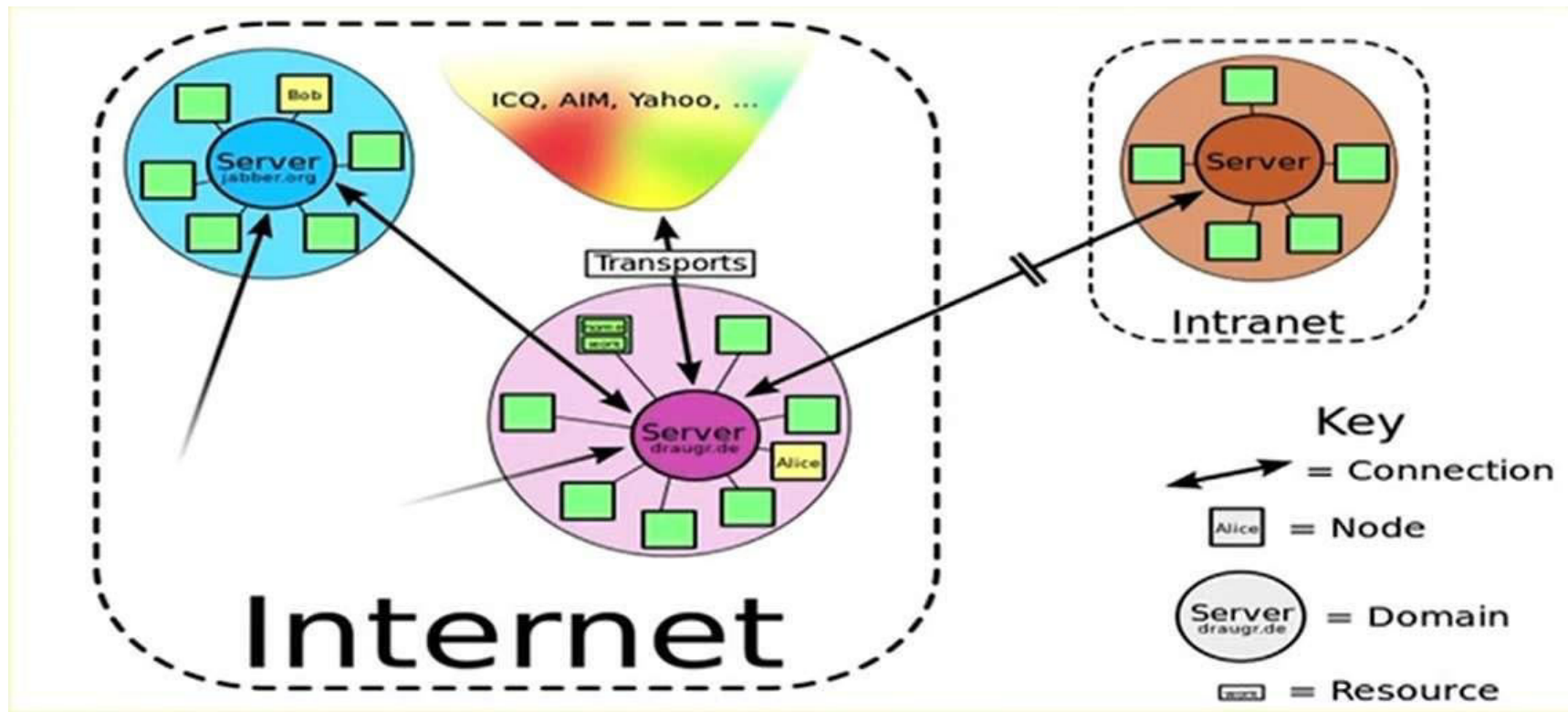
- ✓ XMPP uses a **client-server architecture**.
- ✓ As the model is **decentralized**, no central server is required.
- ✓ XMPP provides for the **discovery of services** residing locally or across a network, and the **availability information** of these services.
- ✓ Well-suited for cloud computing where virtual machines, networks, and firewalls would otherwise present obstacles to alternative service discovery and presence-based solutions.
- ✓ Open means to support machine-to-machine or peer-to-peer communications across a diverse set of networks.



## Highlights

- ✓ Decentralization – No central server; anyone can run their own XMPP server.
- ✓ Open standards – No royalties or granted permissions are required to implement these specifications
- ✓ Security – Authentication, encryption, etc.
- ✓ Flexibility – Supports interoperability





## Core XMPP Technologies

### Core

- information about the core XMPP technologies for XML streaming

### Jingle

- multimedia signalling for voice, video, file transfer

### Multi-user Chat

- flexible, multi-party communication

### PubSub

- alerts and notifications for data syndication

### BOSH

- HTTP binding for XMPP

## Weaknesses

- ✓ Does not support QoS.
- ✓ Text based communications induces higher network overheads.
- ✓ Binary data must be first encoded to **base64** before transmission.

# Applications

- ✓ Publish-subscribe systems
- ✓ Signaling for VoIP
- ✓ Video
- ✓ File transfer
- ✓ Gaming
- ✓ Internet of Things applications
  - Smart grid
  - Social networking services

## 2. Message Queue Telemetry Transport Protocol (MQTT)

MQTT (*Message Queue Telemetry Transport*) is a messaging protocol developed with the aid of Andy Stanford-Clark of IBM and Arlen Nipper of Arcom in 1999 and is designed for M2M communication. It's normally used for faraway tracking in IoT. Its primary challenge is to gather statistics from many gadgets and delivery of its infrastructure. MQTT connects gadgets and networks with packages and middleware. All the devices hook up with facts concentrator servers like IBM's new message sight appliance. MQTT protocols paintings on top of TCP to offer easy and dependable streams of information.

These IoT protocols include 3 foremost additives: subscriber, publisher, and dealer. The writer generates the information and transmits the facts to subscribers through the dealer. The dealer guarantees safety by means of move-checking the authorization of publishers and subscribers.



## Introduction

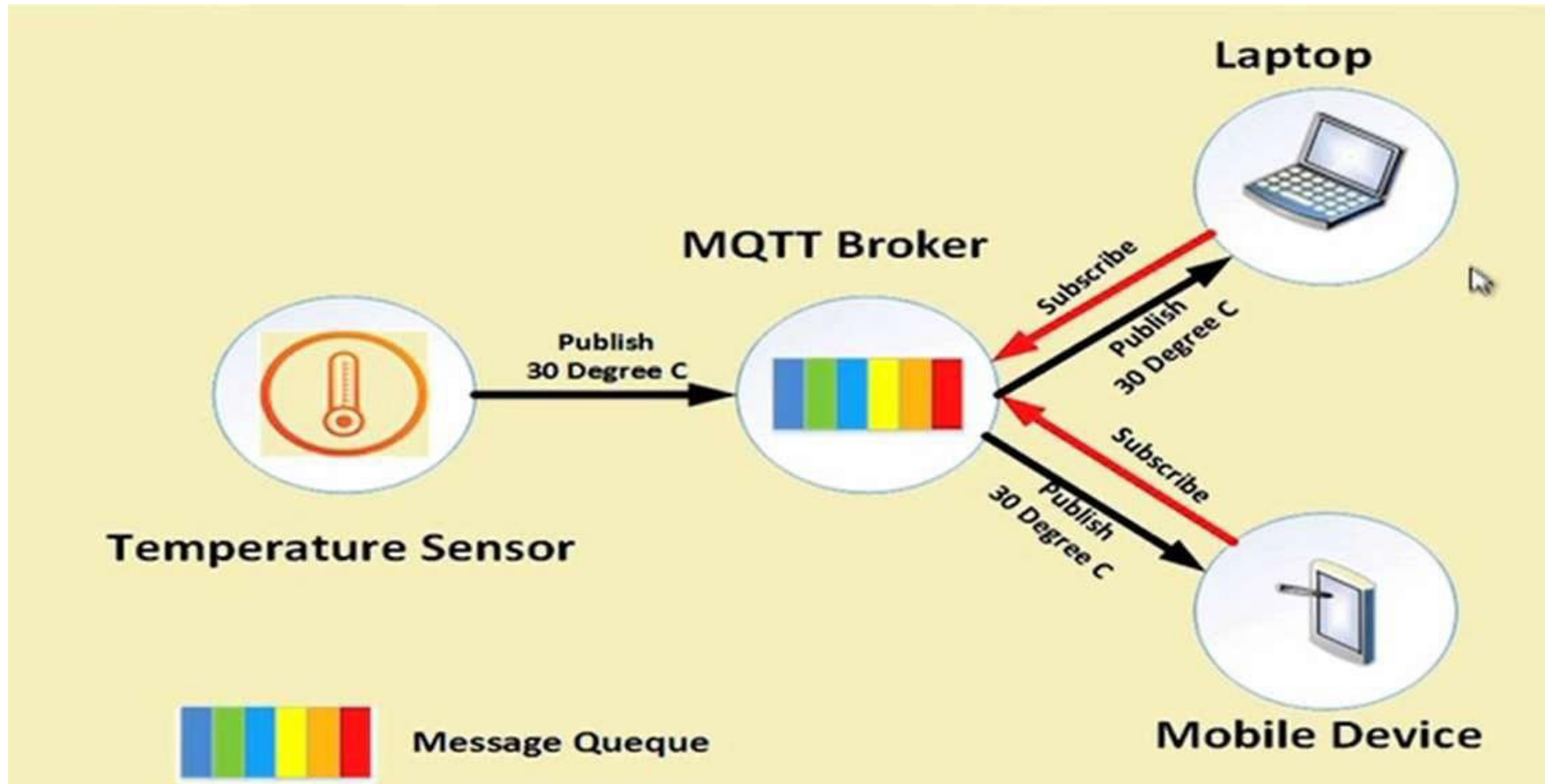
- ✓ **Message Queue Telemetry Transport.**
- ✓ ISO standard (ISO/IEC PRF 20922).
- ✓ It is a publish-subscribe-based lightweight messaging protocol for use in conjunction with the TCP/IP protocol.
- ✓ MQTT was introduced by IBM in 1999 and standardized by OASIS in 2013.
- ✓ Designed to provide connectivity (mostly embedded) between applications and middle-wares on one side and networks and communications on the other side.

- ✓ A message broker controls the publish-subscribe messaging pattern.
- ✓ A topic to which a client is subscribed is updated in the form of messages and distributed by the message broker.
- ✓ Designed for:
  - Remote connections
  - Limited bandwidth
  - Small-code footprint

## MQTT Components







## Communication

- ✓ The protocol uses a **publish/subscribe** architecture (HTTP uses a request/response paradigm).
- ✓ Publish/subscribe is **event-driven** and enables messages to be pushed to clients.
- ✓ The central **communication point is the MQTT broker**, which is in charge of dispatching all messages between the senders and the rightful receivers.
- ✓ Each client that publishes a message to the broker, includes a **topic** into the message. The **topic is the routing information for the broker**.

- ✓ Each client that wants to receive messages subscribes to a certain topic and the broker delivers all messages with the matching topic to the client.
- ✓ Therefore the clients don't have to know each other. They only communicate over the topic.
- ✓ This architecture enables highly scalable solutions without dependencies between the data producers and the data consumers.



## MQTT Topics

- ✓ A topic is a **simple string** that can have more hierarchy levels, which are separated by a slash.
- ✓ A sample topic for sending temperature data of the living room could be *house/living-room/temperature*.
- ✓ On one hand the client (e.g. mobile device) can subscribe to the exact topic or on the other hand, it can use a **wildcard**.

- ✓ The subscription to *house/+/temperature* would result in all messages sent to the previously mentioned topic *house/living-room/temperature*, as well as any topic with an arbitrary value in the place of living room, such as *house/kitchen/temperature*.
- ✓ The plus sign is a **single level wild card** and only allows arbitrary values for one hierarchy.
- ✓ If more than one level needs to be subscribed, such as, the entire sub-tree, there is also a **multilevel wildcard** (#).
- ✓ It allows to subscribe to all underlying hierarchy levels.
- ✓ For example *house/#* is subscribing to all topics beginning with *house*.

## Applications

- ✓ **Facebook Messenger** uses MQTT for online chat.
- ✓ **Amazon Web Services** use Amazon IoT with MQTT.
- ✓ **Microsoft Azure** IoT Hub uses MQTT as its main protocol for telemetry messages.
- ✓ The **EVERYTHING IoT platform** uses MQTT as an M2M protocol for millions of connected products.
- ✓ **Adafruit** launched a free MQTT cloud service for IoT experimenters called Adafruit IO.





## SMQTT

- ✓ **Secure MQTT** is an extension of MQTT which uses encryption based on lightweight attribute based encryption.
- ✓ The main advantage of using such encryption is the broadcast encryption feature, in which one message is encrypted and delivered to multiple other nodes, which is quite common in IoT applications.
- ✓ In general, the algorithm consists of four main stages: setup, encryption, publish and decryption.



- ✓ In the setup phase, the subscribers and publishers register themselves to the broker and get a master secret key according to their developer's choice of key generation algorithm.
- ✓ When the data is published, it is encrypted and published by the broker which sends it to the subscribers, which is finally decrypted at the subscriber end having the same master secret key.
- ✓ The key generation and encryption algorithms are not standardized.
- ✓ SMQTT is proposed only to enhance MQTT security features.

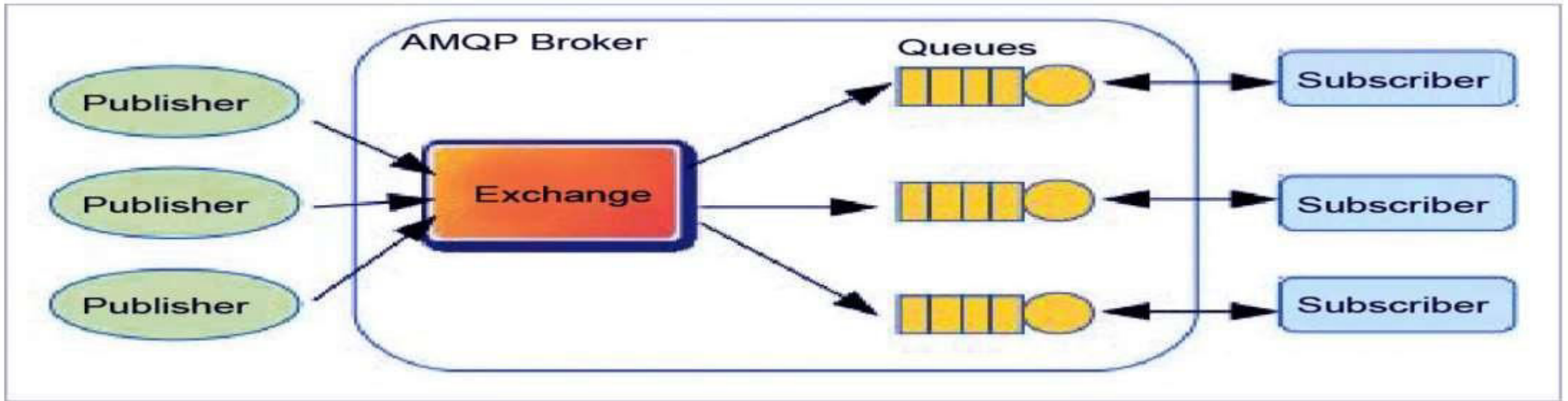


### 3. Advanced Message Queuing Protocol (AMQP)

This was evolved by John O'Hara at JP Morgan Chase in London. AMQP is a software layer protocol for message-oriented middleware environment. It supports reliable verbal exchange through message transport warranty primitives like at-most-once, at least once and exactly as soon as shipping.

The AMQP – IoT protocols consist of hard and fast components that route and save messages within a broker carrier, with a set of policies for wiring the components together. The AMQP protocol enables patron programs to talk to the dealer and engage with the [AMQP model](#). This version has the following three additives, which might link into processing chains in the server to create the favored capabilities.

- **Exchange:** Receives messages from publisher primarily based programs and routes them to 'message queues'.
  - **Message Queue:** Stores messages until they may thoroughly process via the eating client software.
  - **Binding:** States the connection between the message queue and the change.
-

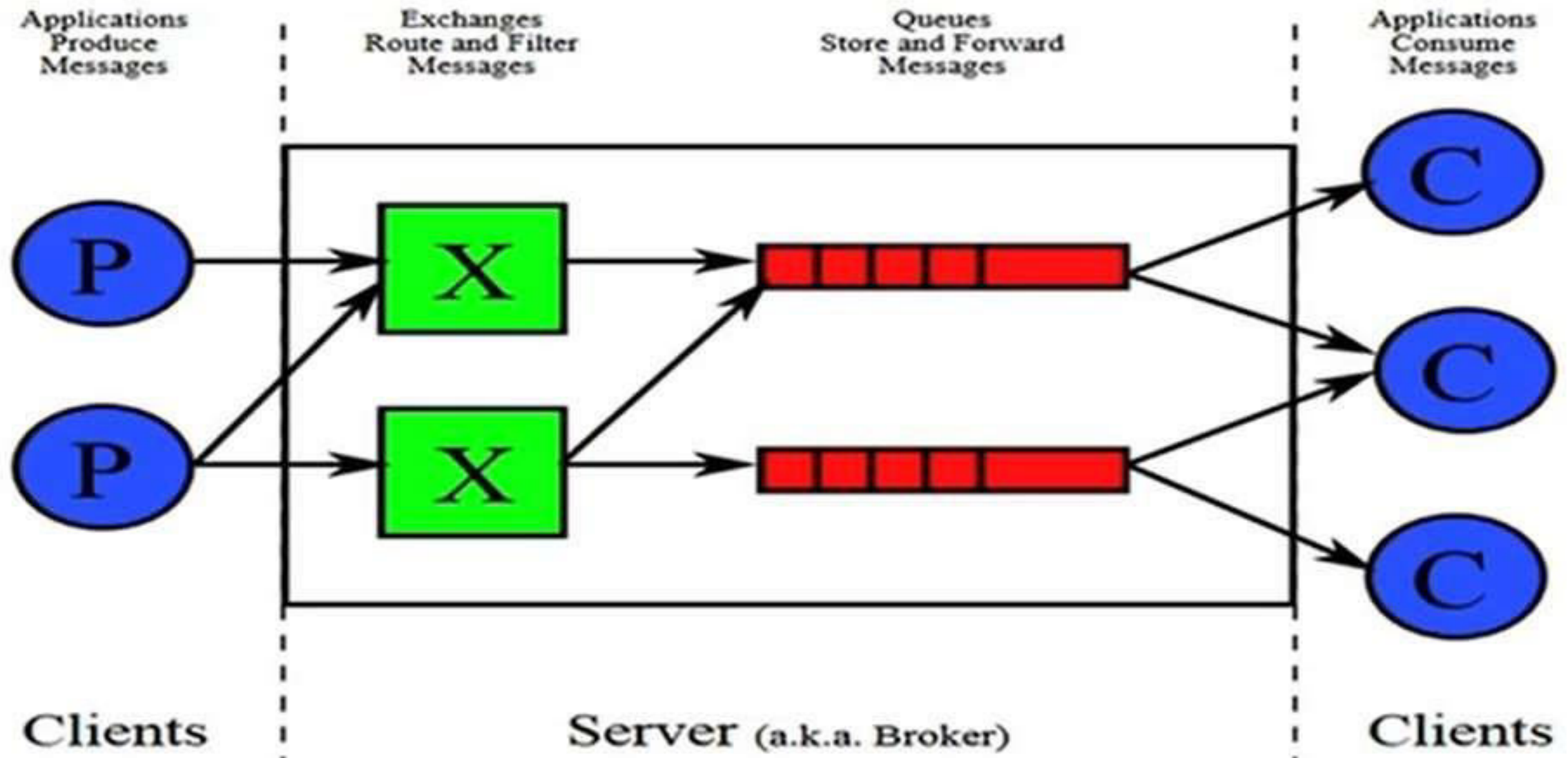




## Introduction

- ✓ **Advanced Message Queuing Protocol.**
- ✓ **Open standard for passing business messages** between applications or organizations.
- ✓ Connects between systems and business processes.
- ✓ It is a binary application layer protocol.
- ✓ Basic unit of data is a *frame*.
- ✓ ISO standard: **ISO/IEC 19464**

## FUNDAMENTALS OF INTERNET OF THINGS ( EC32110E)



# AMQP Features



## Features



## AMQP Frame Types

- ✓ Nine AMQP frame types are defined that are used to initiate, control and tear down the transfer of messages between two peers:
  - Open (connection open)
  - Begin (session open)
  - Attach (initiate new link)
  - Transfer (for sending actual messages)
  - Flow (controls message flow rate)
  - Disposition (Informs the changes in state of transfer)
  - Detach (terminate the link)
  - End (session close)
  - Close (connection close)



# Components

## Exchange

- Part of Broker
- Receives messages and routes them to Queues

## Queue

- Separate queues for separate business processes
- Consumers receive messages from queues

## Bindings

- Rules for distributing messages (who can access what message, destination of the message)

# AMQP Exchanges

## Direct

## Fan-out

## Topic

## Header

## AMQP Features

- ✓ Targeted QoS (Selectively offering QoS to links)
- ✓ Persistence (Message delivery guarantees)
- ✓ Delivery of messages to multiple consumers
- ✓ Possibility of ensuring multiple consumption
- ✓ Possibility of preventing multiple consumption
- ✓ High speed protocol

## Applications

- ✓ Monitoring and global update sharing.
- ✓ Connecting different systems and processes to talk to each other.
- ✓ Allowing servers to respond to immediate requests quickly and delegate time consuming tasks for later processing.
- ✓ Distributing a message to multiple recipients for consumption.
- ✓ Enabling offline clients to fetch data at a later time.
- ✓ Introducing fully asynchronous functionality for systems.
- ✓ Increasing reliability and uptime of application deployments.



## 4. Data Distribution Service (DDS)

It enables a scalable, real-time, reliable, excessive-overall performance and interoperable statistics change via the submit-subscribe technique. DDS makes use of multicasting to convey high-quality QoS to applications.

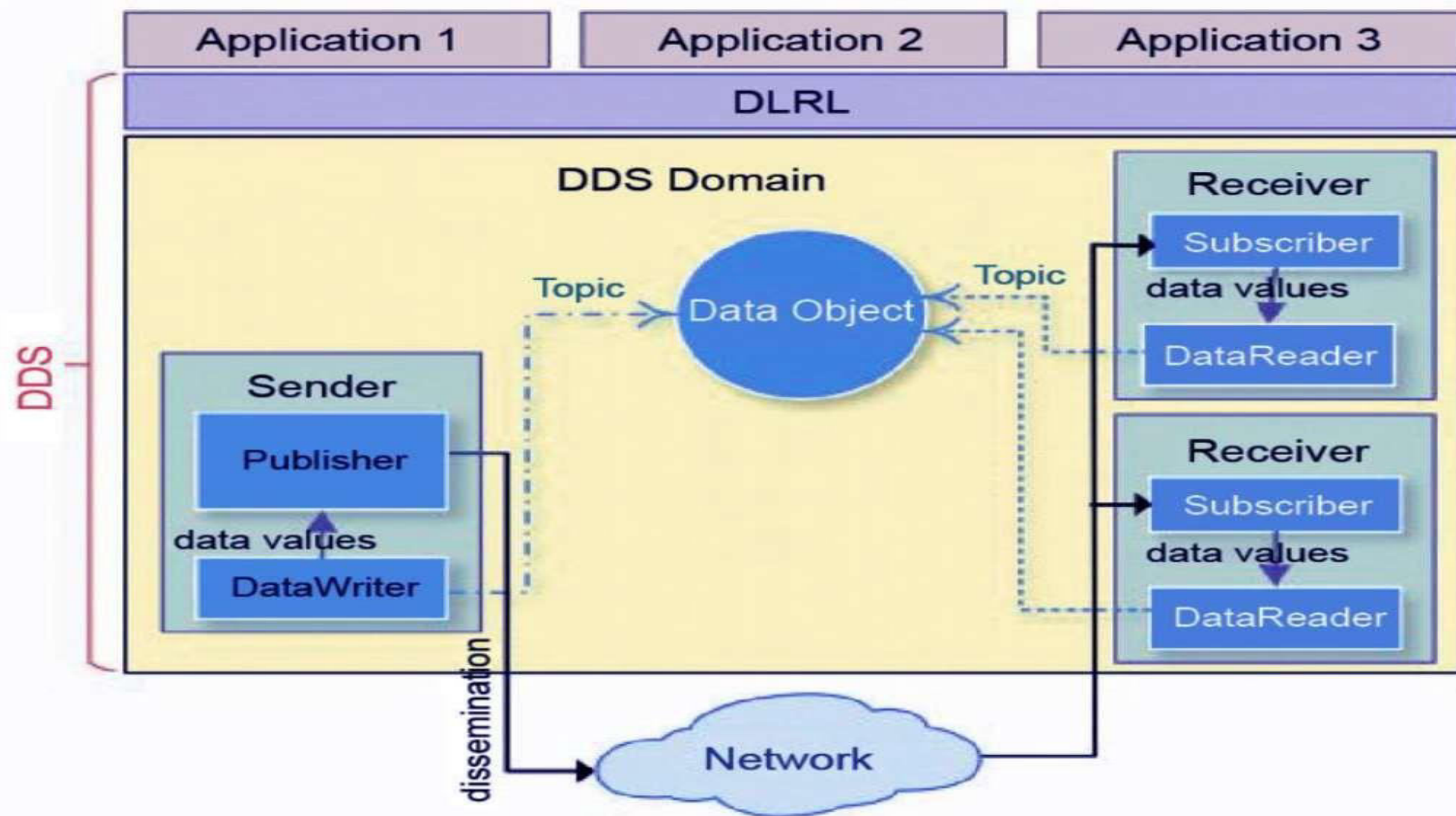
DDS is deployed in platforms ranging from low-footprint devices to the cloud and supports green bandwidth usage in addition to the agile orchestration of system additives.

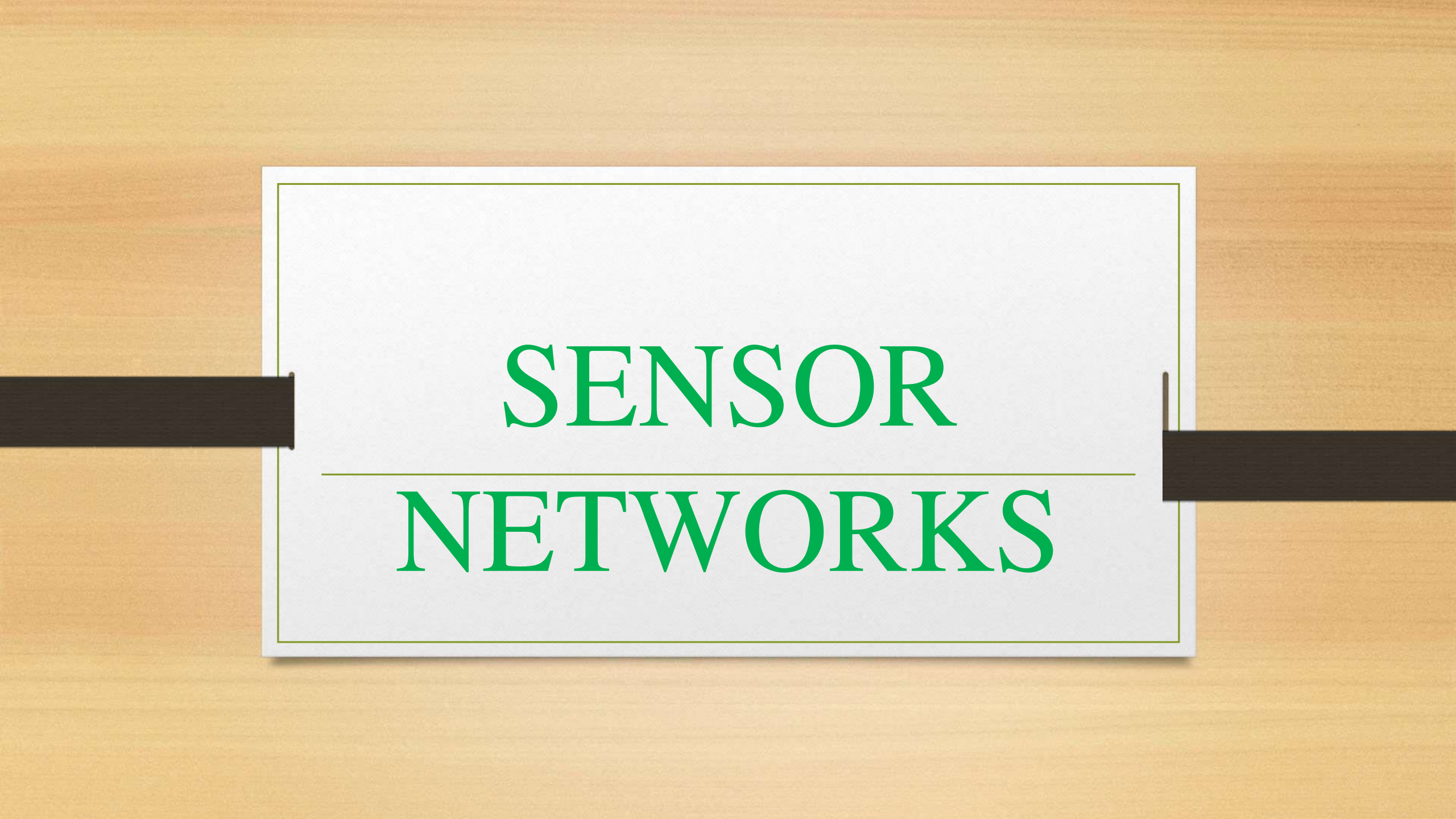
The DDS – IoT protocols have fundamental layers: facts centric submit-subscribe (dcps) and statistics-local reconstruction layer (dlrl). Dcps plays the task of handing over the facts to subscribers, and the dlrl layer presents an interface to dcps functionalities, permitting the sharing of distributed data amongst IoT enabled objects.

---



## FUNDAMENTALS OF INTERNET OF THINGS ( EC32110E)





# SENSOR NETWORKS

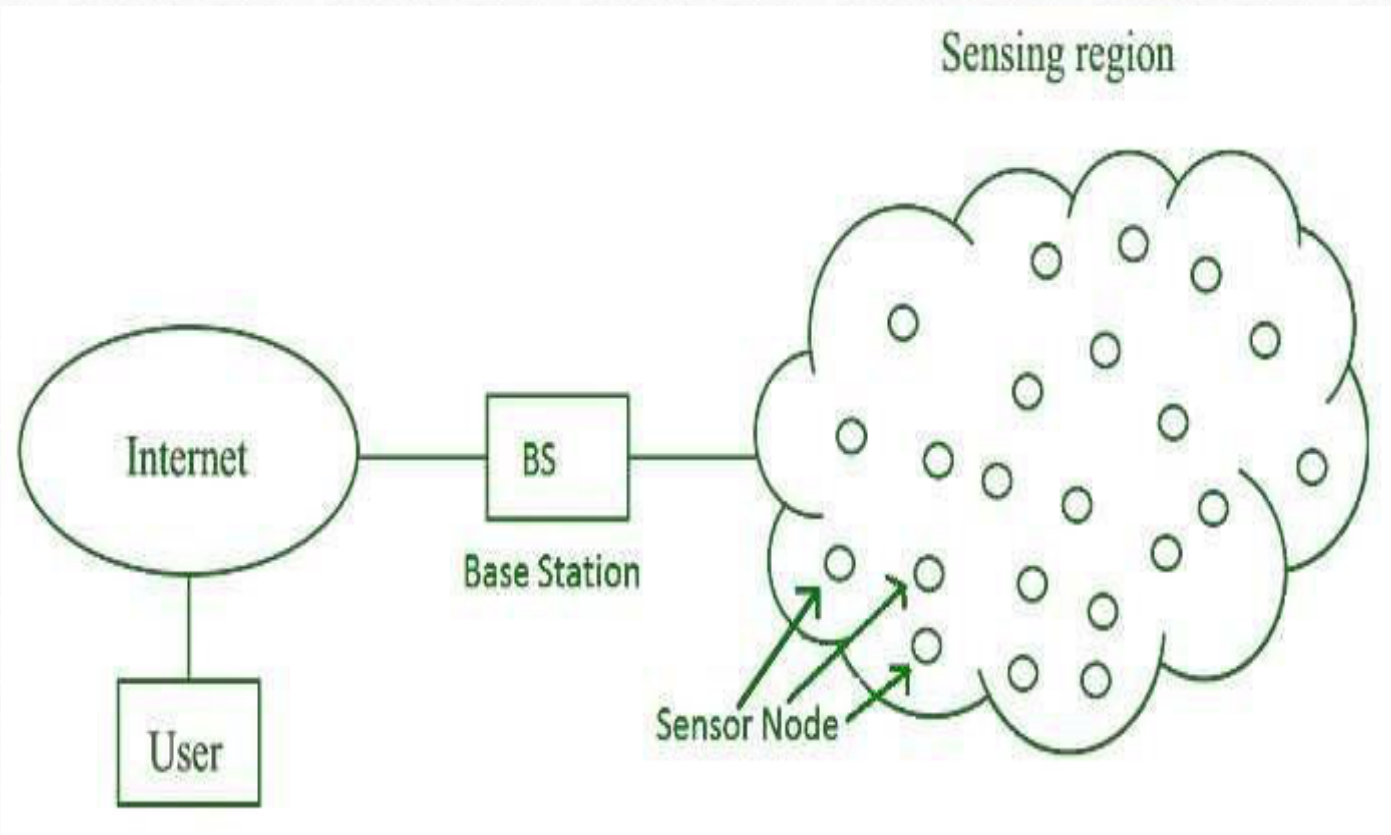




## Wireless Sensor Networks (WSNs)

- Consists of a large number of sensor nodes, densely deployed over an area.
- Sensor nodes are capable of collaborating with one another and measuring the condition of their surrounding environments (i.e. Light, temperature, sound, vibration).
- The sensed measurements are then transformed into digital signals and processed to reveal some properties of the phenomena around sensors.
- Due to the fact that the sensor nodes in WSNs have short radio transmission range, intermediate nodes act as relay nodes to transmit data towards the sink node using a multi-hop path.





# TYPES OF SENSOR NODE

---

- STATIONARY SENSOR NODE
- MOBILE SENSOR NODE
  1. AERIAL SENSOR NODE
  2. TERESTRIAL SENSOR NODE
  3. UNDERWATER SENSOR NODE

# SENSOR NODE

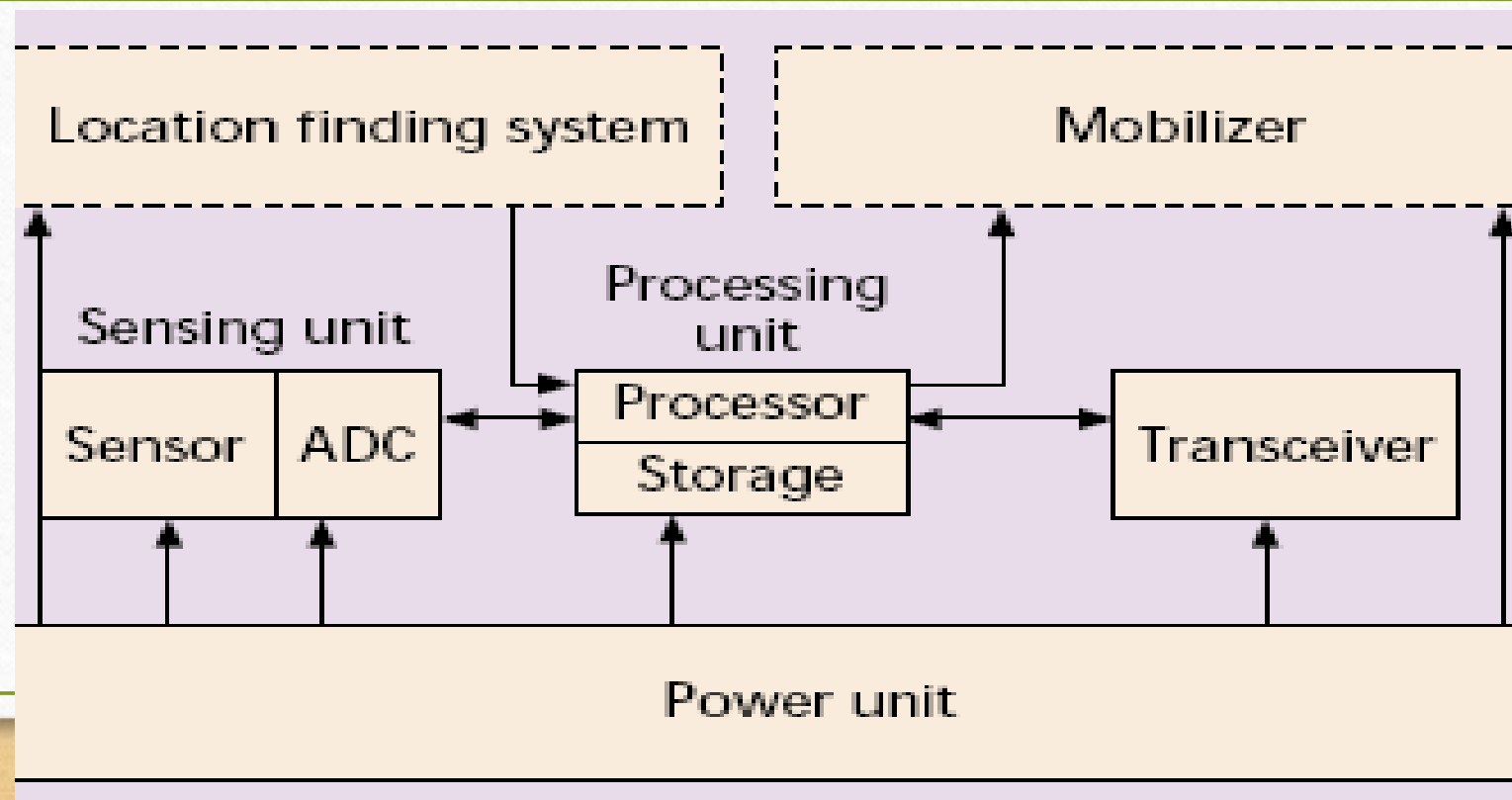
---

- SENSING UNIT
- COMMUNICATION UNIT (TRANS-RECEIVER)
- PROCESSING UNIT
- STORAGE UNIT
- ANALOG TO DIGITAL CONVERTER
- OPTIONAL UNITS (LOCATION FINDING SYSTEMS Eg. GPS)

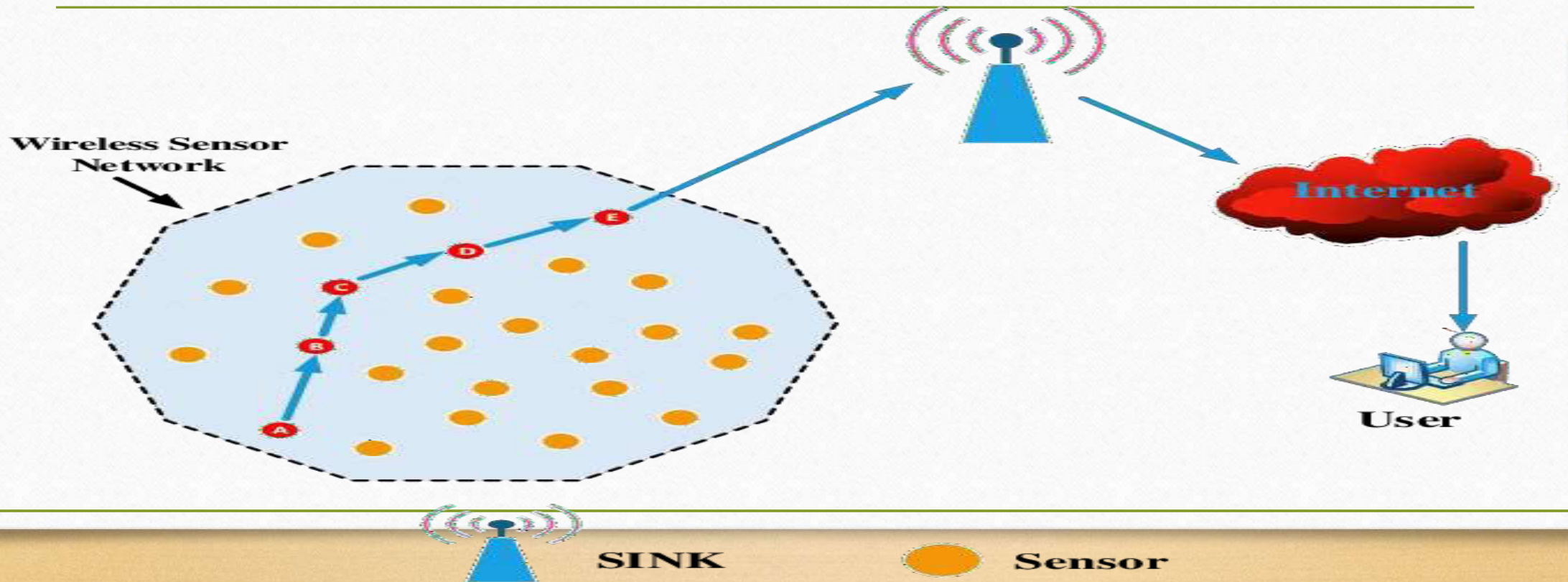
- POWER (BATTERY)



# BASIC COMPONENTS OF SENSOR NODE



# MULTI-HOP PATH







## Sensor Nodes

- **Multifunctional**
  - The number of sensor nodes used depends on the application type.
- **Short transmission ranges**
- **Have OS** (e.g., TinyOS).
- **Battery Powered** – Have limited life.



Image source: Wikimedia Commons

# CONSTRAINTS OF SENSOR NODES

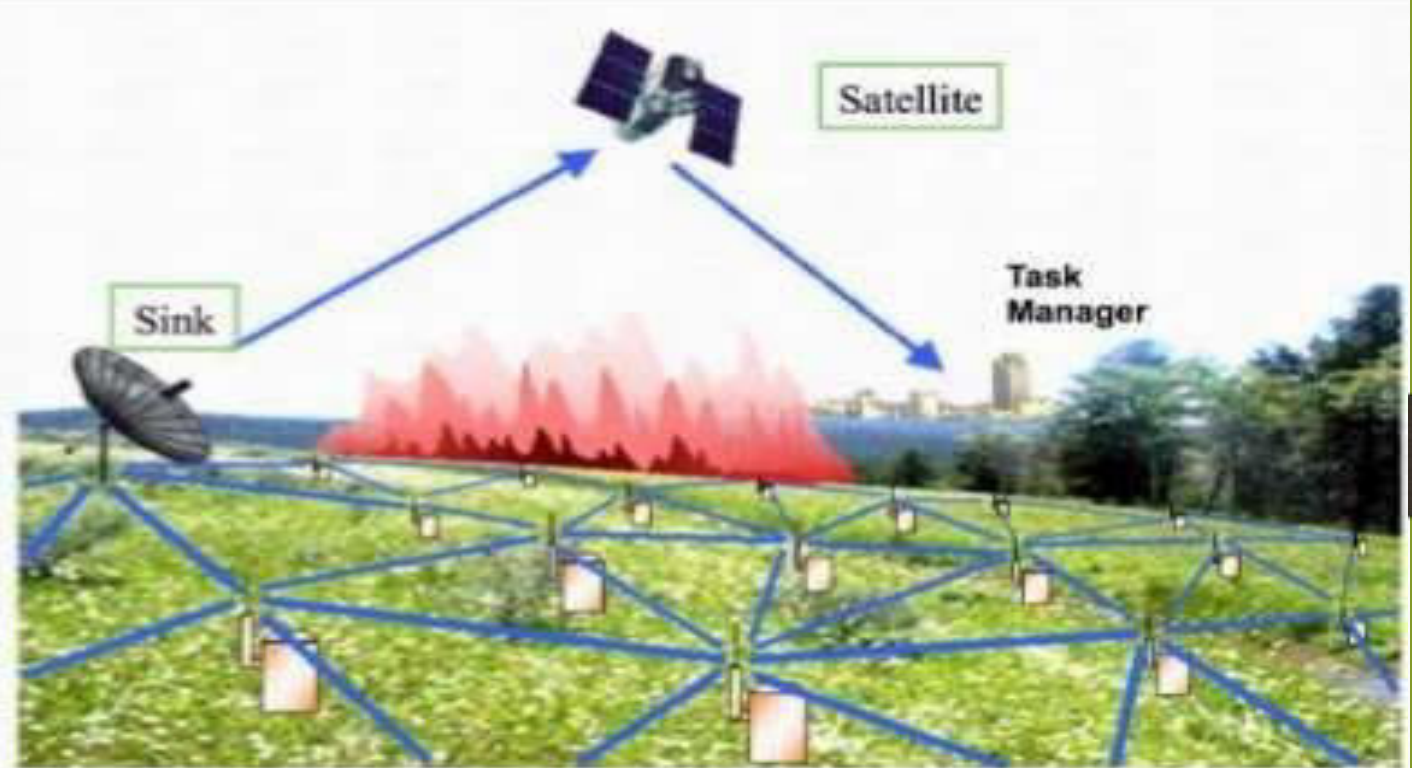
---

- Small size, typically less than a cubic cm.
- Must consume extremely low power.
- Operate in an unattended manner in a highly dense area.
- Should have low production cost and be dispensable.
- Be autonomous
- Be adaptive to the environment.



# Applications of WSN-

- The Military Applications
- The Medical Applications
- Environmental Monitoring
- Target tracking
- Industrial Application
- Infrastructure and protection application



# CHALLENGES

## SCALABILITY

---

- Providing acceptable levels of service in the presence of large number of nodes.
- Typically, throughput decreases at a rate of  $1/\sqrt{N}$ ,  $N$ =number of nodes

## QUALITY OF SERVICE

- Offering guarantees in terms of bandwidth, delay, jitter, packet loss probability.
- Limited bandwidth, unpredictable changes in RF channel characteristics.



## ENERGY EFFICIENCY

---

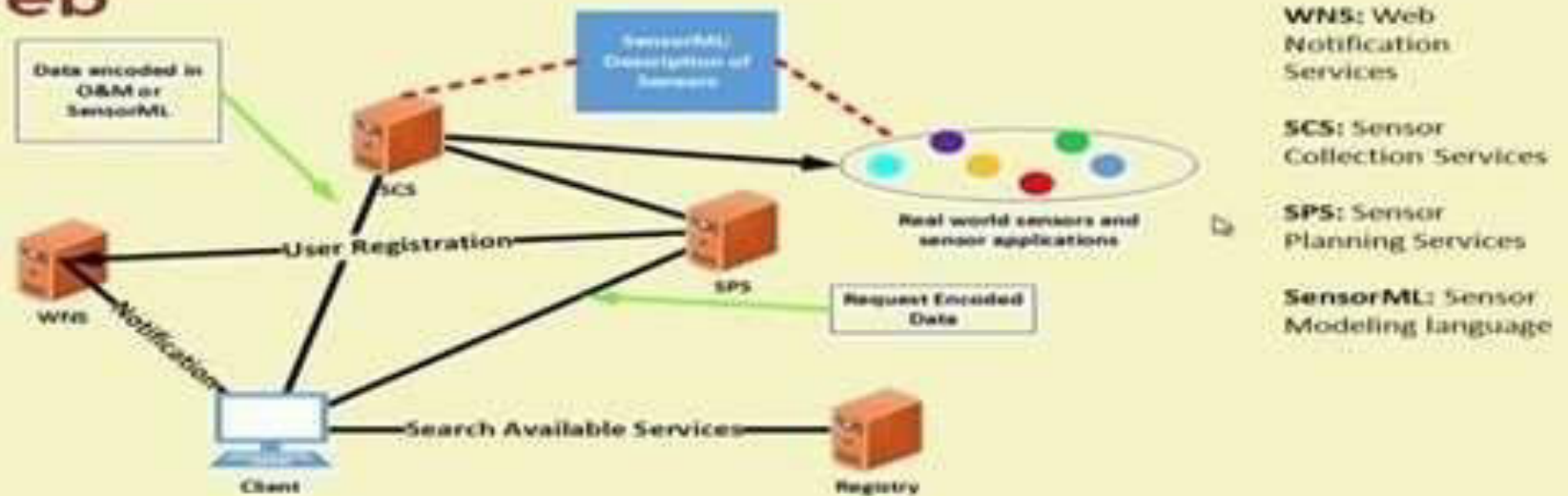
- Nodes have limited battery power
- Nodes need to cooperate with other nodes for relaying their information

## SECURITY

- Open medium
- Nodes prone to malicious attacks, infiltration, eavesdropping, interference.



# Sensor Web



Source: X. Chu and R. Buyya, "Service Oriented Sensor Web", Sensor Networks and Configuration, Springer, 2007, pp. 51-74.



## Sensor Web Entanglement

- Observations & measurements (O&M)
- Sensor model language (sensorml)
- Transducer model language (transducerml or TML)
- Sensor observations service (SOS)
- Sensor planning service (SPS)
- Sensor alert service (SAS)
- Web notification services (WNS)

# Node Behaviour in WSN

---

- In order to have communication among different sensor nodes, these nodes they basically have to cooperate with one another.
- In order for the network to function if the nodes do not cooperate they will not be able to function.
- To promote cooperation we need to understand the behavior of the different nodes in the network.



## Node Behavior in WSNs







## Node Behavior in WSNs (contd.)

- **Normal nodes** work perfectly in ideal environmental conditions
- **Failed nodes** are simply those that are unable to perform an operation; this could be because of power failure and environmental events.
- **Badly failed nodes** exhibit features of failed nodes but they can also send false routing messages which are a threat to the integrity of the network.



## Node Behavior in WSNs (contd.)

- **Selfish nodes** are typified by their unwillingness to cooperate, as the protocol requires whenever there is a personal cost involved. Packet dropping is the main attack by selfish nodes.
- **Malicious nodes** aim to deliberately disrupt the correct operation of the routing protocol, denying network service if possible.







## Dynamic Misbehavior: Dumb Behavior

- Detection of such temporary misbehavior in order to preserve normal functioning of the network – coinage and discovery of dumb behavior
- In the presence of adverse environmental conditions (high temperature, rainfall, and fog) the communication range shrinks
- A sensor node can sense its surroundings but is unable to transmit the sensed data
- With the resumption of favorable environmental conditions, dumb nodes work normally
- Dumb behavior is temporal in nature (as it is dependent on the effects of environmental conditions)



## Detection and Connectivity Re-establishment

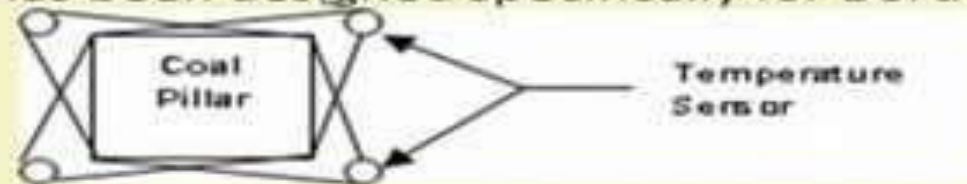
- The presence of dumb nodes impedes the overall network performance
- Detection, and, subsequently, the re-establishment of network connectivity is crucial
- The sensed information can only be utilized if the connectivity between each dumb node with other nodes in the network could be re-established
- Before restoration of network connectivity, it is essential to detect the dumb nodes in the network.
- CoRD and CoRAD are two popular schemes that re-establish the connectivity between dumb nodes with others.





## Applications of WSNs: Mines

- ✓ Fire Monitoring and Alarm System for Underground Coal Mines Bord-and-Pillar Panel Using Wireless Sensor Networks
  - WSN-based simulation model for building a fire monitoring and alarm (FMA) system for Bord & Pillar coal mine.
  - The fire monitoring system has been designed specifically for Bord & Pillar based mines



Source: S. Bhattacharjee, P. Roy, S. Ghosh, S. Misra, M. S. Obaidat, "Fire Monitoring and Alarm System for Underground Coal Mines Bord-and-Pillar Panel Using Wireless Sensor Networks", *Journal of Systems and Software* (Elsevier), Vol. 85, No. 3, March 2012, pp. 571-581.





## Applications of WSNs: Mines (contd.)

- It is not only capable of providing real-time monitoring and alarm in case of a fire, but also capable of providing the exact fire location and spreading direction by continuously gathering, analysing, and storing real time information



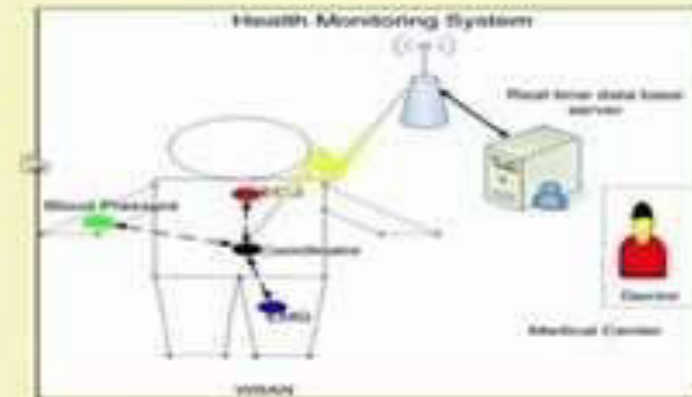
**Sources:** S. Bhattacharjee, P. Roy, S. Ghosh, S. Misra, M. S. Obaidat, "Fire Monitoring and Alarm System for Underground Coal Mines Bord-and-Pillar Panel Using Wireless Sensor Networks", *Journal of Systems and Software* (Elsevier), Vol. 85, No. 3, March 2012, pp. 571-581.



## Applications of WSNs: Healthcare

### ✓ Wireless Body Area Networks

- Wireless body area networks (WBANs) have recently gained popularity due to their ability in providing innovative, cost-effective, and user-friendly solution for continuous monitoring of vital physiological parameters of patients.
- Monitoring chronic and serious diseases such as cardiovascular diseases and diabetes.
- Could be deployed in elderly persons for monitoring their daily activities.









## Target Tracking

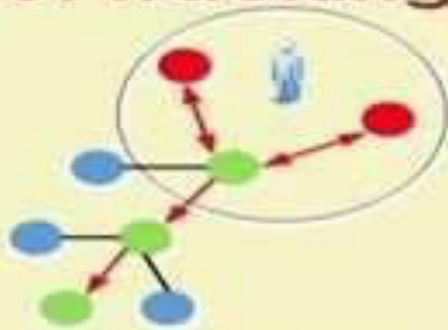


Fig. a: Push-based formulation: Nodes compute the position of the target and periodically notify the sink node. A cluster structure is commonly used in this case

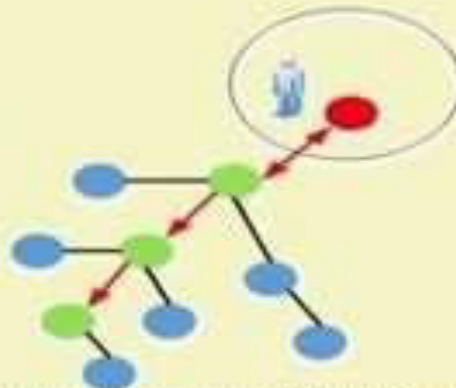


Fig. b: Poll-based formulation: Nodes register the presence of the target to permit a low-cost query. Data reports are sent toward the sink only when there is a query to be answered. Tree structure is often used in this case.

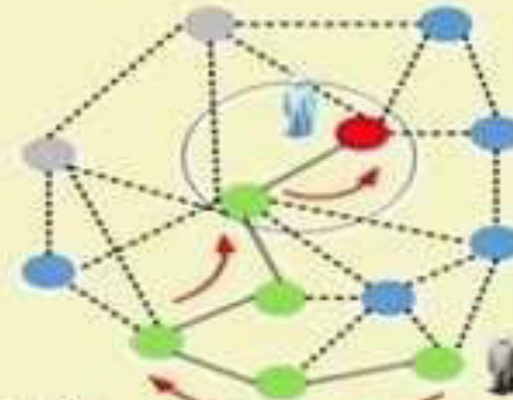
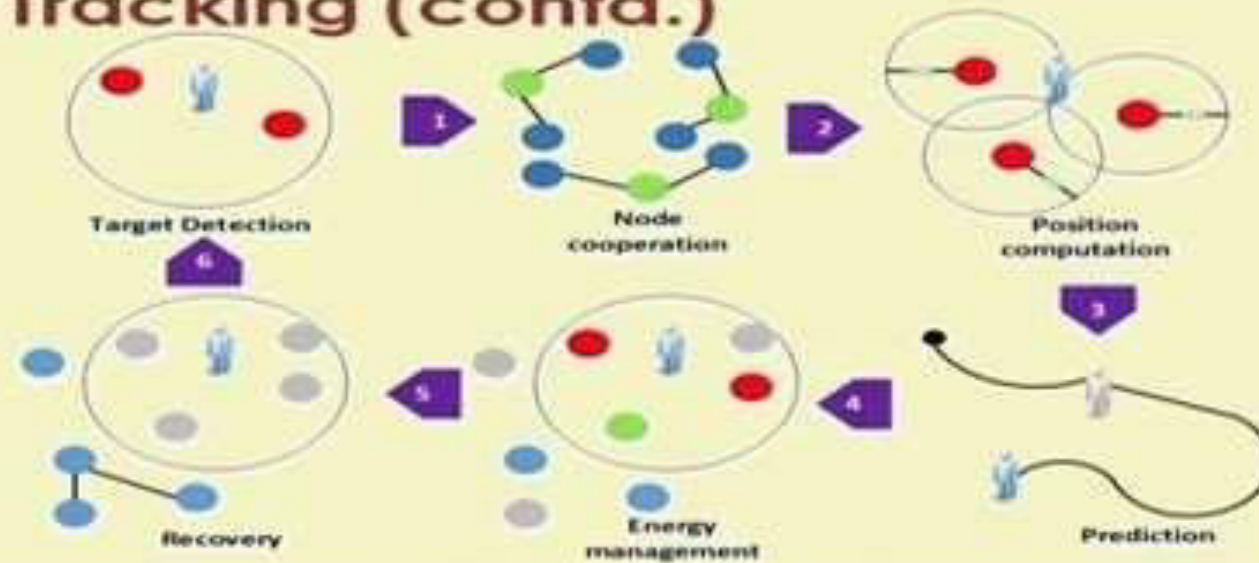


Fig. c: Guided formulation: Some nodes (beacon nodes) define a trajectory to the target. The tracker follows this trail to intercept the target. Face structure is often used in this case

Sources: Efrén L. Souza, Eduardo F. Nakamura, and Richard W. Pazzi. 2016. Target Tracking for Sensor Networks: A Survey. *ACM Computing Survey*, 49, 2, 2016



## Target Tracking (contd.)



Source: Éfren L. Souza, Eduardo F. Nakamura, and Richard W. Pazzi, 2016. Target Tracking for Sensor Networks: A Survey. ACM Computing Survey, 49, 2, 2016





## WSNs in Agriculture

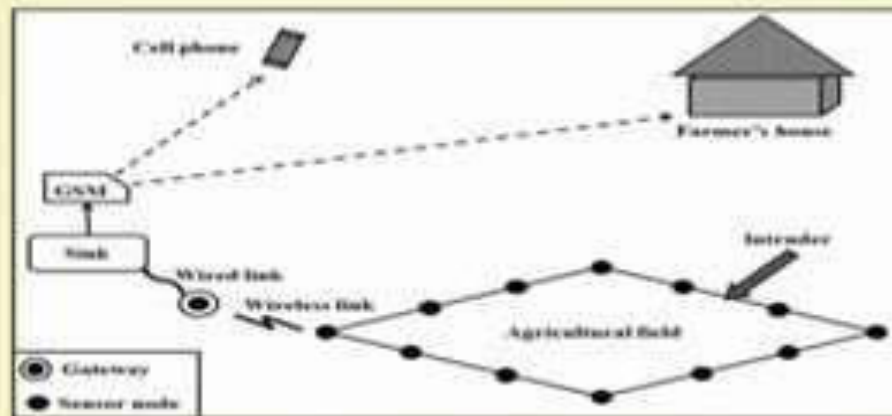
- ✓ AID: A Prototype for Agricultural Intrusion Detection Using Wireless Sensor Network
- A set of sensor nodes are deployed over an agricultural field
- Each of the board are enabled with two type of sensors:
  - a) Passive Infrared (PIR)
  - b) Ultrasonic
- When an intruder enters into the field through the boundary (perimeter) of the field, the PIR sensor detects the **object**.
- The ultrasonic sensor senses the **distance** at which the object is located

Sources: Sanjukta Kumar Roy, Arijit Roy, Sudip Misra, Narendra S Raghunwarshi, Mohammad S Obaidat, AID: A Prototype for Agricultural Intrusion Detection Using Wireless Sensor Network, IEEE International Conference on Communications (ICC), 2015





## WSNs in Agriculture (contd.)





## Wireless Multimedia Sensor Networks (WMSNs)

- Incorporation of low cost camera (typically CMOS ) to wireless sensor nodes
- Camera sensor (CS) nodes
  - capture multimedia (video, audio, and the scalar) data, expensive and resource hungry, directional sensing range
- Scalar sensor (SS) nodes
  - sense scalar data (temperature, light, vibration, and so on), omni-directional sensing range , and low cost
- WMSNs consist of less number of CS nodes and large number of SS nodes







## Wireless Multimedia Sensor Networks (WMSNs)

- WMSNs Application
  - In security surveillance, wild-habitat monitoring, environmental monitoring, SS nodes cannot provide precise information
  - CS nodes replace SS nodes to get precise information
  - Deployment of both CS and SS nodes can provide better sensing and prolong network lifetime

# WSN COVERAGE

- 
- Coverage- area of interest is covered satisfactorily
  - Connectivity- all the nodes are connected in the network, so that sensed data can reach to sink node
  - Sensor Coverage studies how to deploy or activate sensor to cover monitoring area
    - Sensor placement
    - Density control
  - Two modes
    - Static sensors
    - Mobile sensors



- **Definitions:**

- Sensing range  $[r_s]$
- Transmission range  $[r_t]$

- **Relationship between coverage and connectivity**

- If transmission range  $\geq 2 \times$  sensing range,
- coverage implies connectivity
- Most sensors satisfy the condition
  - Coverage is the main issue





# COVERAGE

---

- Determine how well the sensing field is monitored or tracked by sensors
- To determine, with respect to application-specific performance criteria,
  - In case of static sensors, where to deploy and /or activate them
  - In case of mobile sensors, how to plan the trajectory of the mobile sensors
- These two cases are collectively termed as the coverage problem in wireless sensor networks.



# Coverage

---

- The purpose of deploying a WSN is to collect relevant data for processing or reporting
- Two types of reporting
  - Event driven
    - E.g. forest fire monitoring
  - On demand
    - E.g. inventory control system

- Objective is to use a minimum number of sensors and maximize the network lifetime

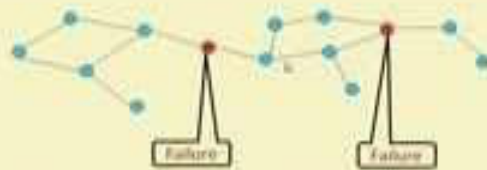
# SATIONARY WIRELESS SENSOR NETWORKS

---

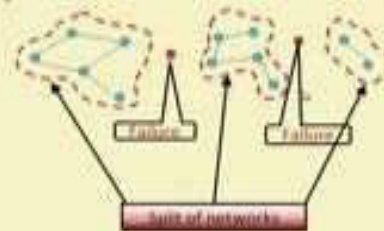
- Sensor nodes are static
- Advantages
  - Easy deployment
  - Node can be placed in an optimized distance- Reduce the total number of nodes
  - Easy topology maintenance
- Disadvantages
  - Node failure may result in partition of networks

- Topology cannot be changed automatically

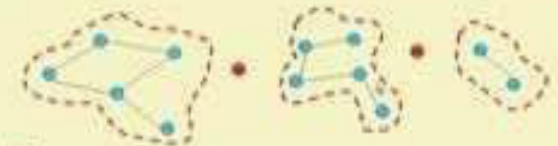
### Stationary Wireless Sensor Networks (Contd.)



### Stationary Wireless Sensor Networks (Contd.)



### Stationary Wireless Sensor Networks (Contd.)



Solution?

To mobilize the sensor nodes

Mobile Wireless Sensor Networks (MWSN)



# MOBILE WIRELESS SENSOR NETWORKS

---

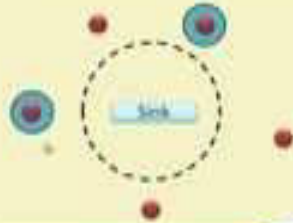
- MWSN is Mobile Ad hoc Network (MANET)
- MANET- infrastructure less network of mobile devices connected wirelessly which follow the Self- CHOP properties
- Self-Configure
- Self-Heal
- Self-Optimize

- Self-Protect

### Components of MWSN

- **Mobile Sensor Nodes**

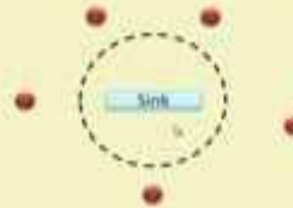
- Sense physical parameters
- from the environment



### Components of MWSN

- **Mobile Sensor Nodes**

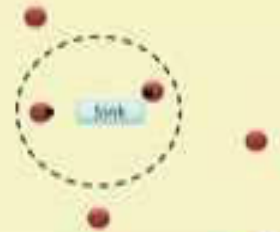
- Sense physical parameters
- from the environment
- When these nodes come in close proximity of sink,
- deliver data



### Components of MWSN

- **Mobile Sensor Nodes**

- Sense physical parameters
- from the environment
- When these nodes come in close proximity of sink,
- deliver data



### Components of MWSN

- **Mobile Sink**

- Moves in order to collect data from sensor nodes
- Based on some algorithm sink moves to different nodes in the networks



### Components of MWSN

- **Data Mules**

- A mobile entity
- Collects the data from sensor nodes
- Goes to the sink and delivers the collected data from different sensor nodes









### Underwater MWSNs

- ✓ Senses different parameters under the sea or water levels
- ✓ Can be linked with Autonomous Underwater Vehicles (AUVs)
- ✓ Applications: Monitoring-marine life, water quality etc.



### Terrestrial MWSNs

- ✓ Sensor nodes typically deployed over land surface
- ✓ Can be linked with Unmanned Aerial Vehicles (UAVs)
- ✓ Applications: Wildlife monitoring, surveillance, object tracking

## Aerial MWSNs

- ✓ Nodes fly on the air and sense data (physical phenomena or multimedia data)
- ✓ Typical example is Unmanned Aerial Vehicles (UAVs)
- ✓ Applications: Surveillance, Multimedia data gathering