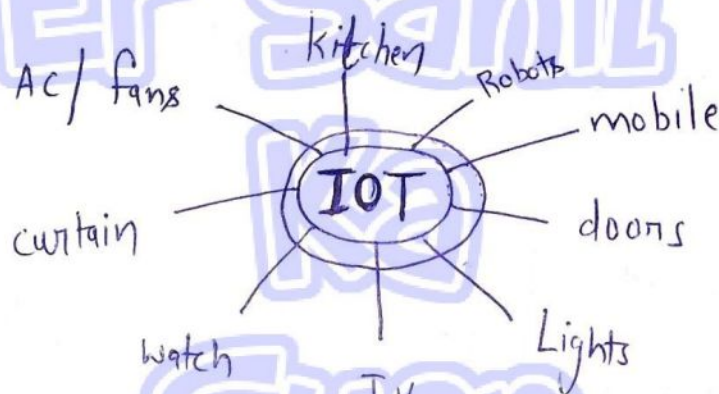


IOT

The internet of things (IOT) describes the network of physical objects - "things" - that are embedded with sensors, slw and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.



Smart Home

characteristics of IOT

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

Physical Design of IOT \Rightarrow

\rightarrow The "Things" in IOT usually refers to IOT devices which have unique identities and can perform remote sensing, actuating & monitoring.

IOT devices can

- Exchange data with connected devices
- Collect data from other devices & process the data
- Send data to centralized servers
- Perform some tasks locally.

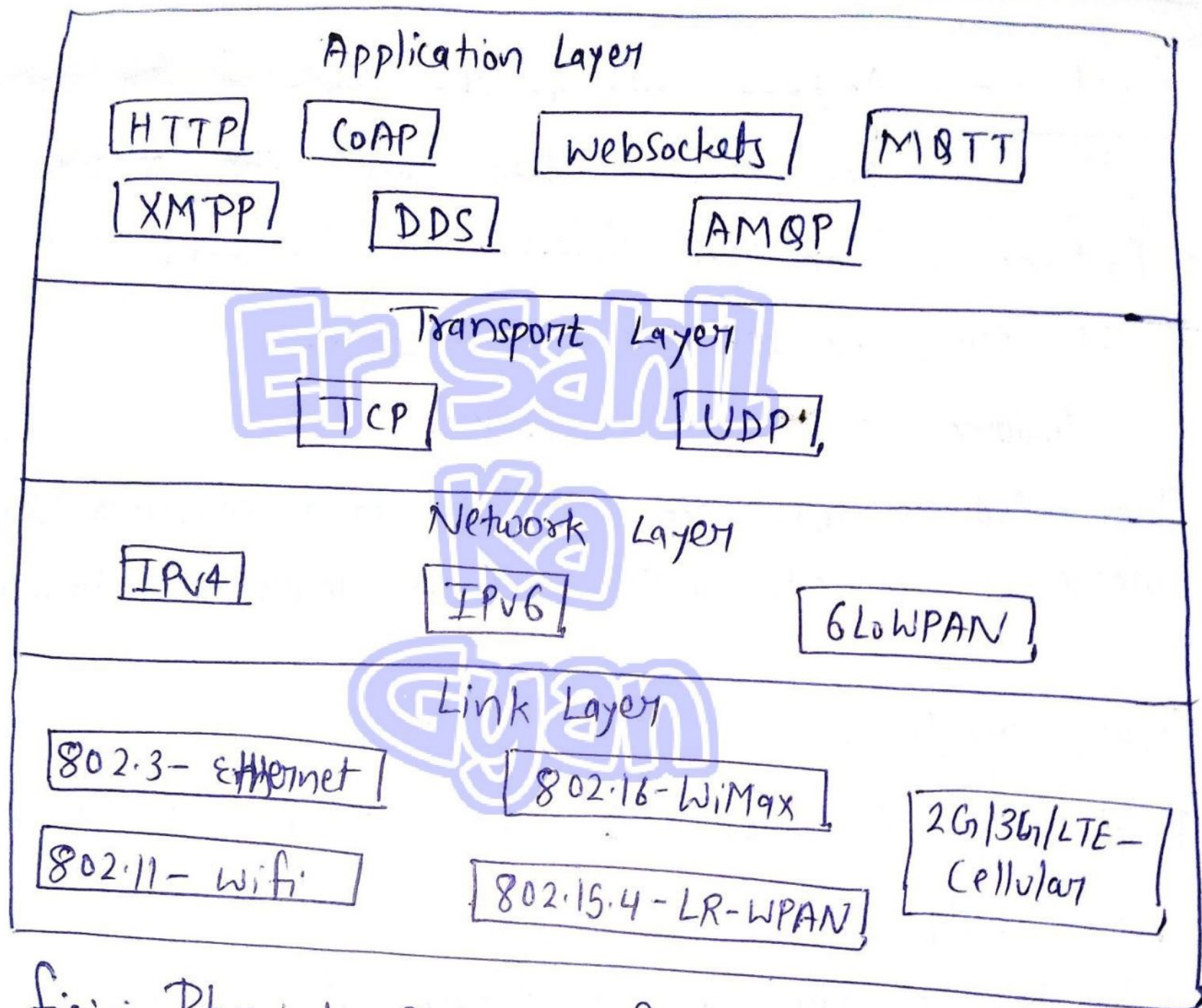
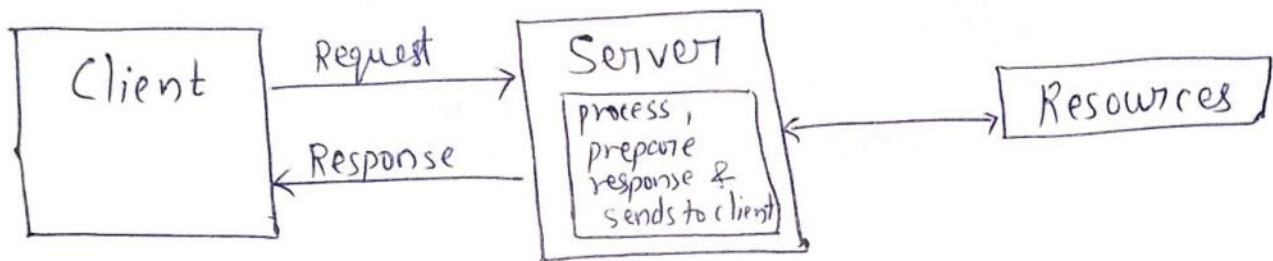


fig:- Physical Design of IOT

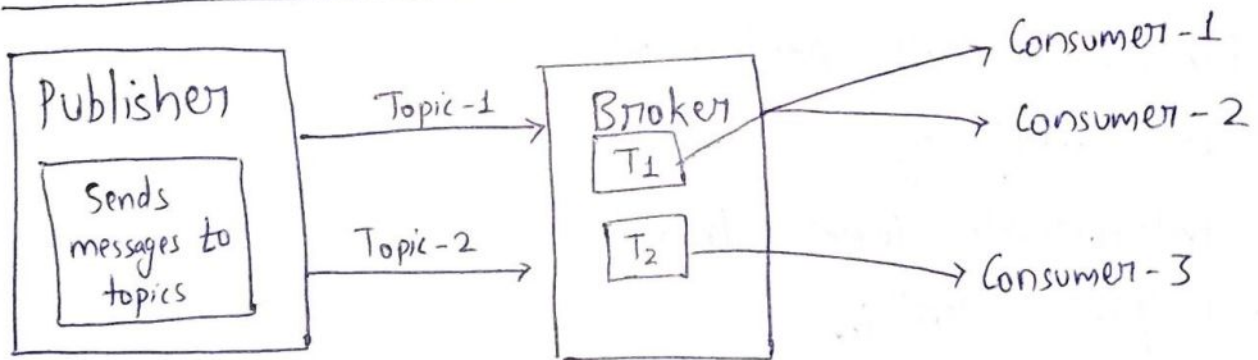
IOT Communication Model \Rightarrow

- (i) Request Response commⁿ Model
- (ii) Publish - Subscribe commⁿ model
- (iii) Push - Pull commⁿ model
- (iv) Exclusive Pair commⁿ model

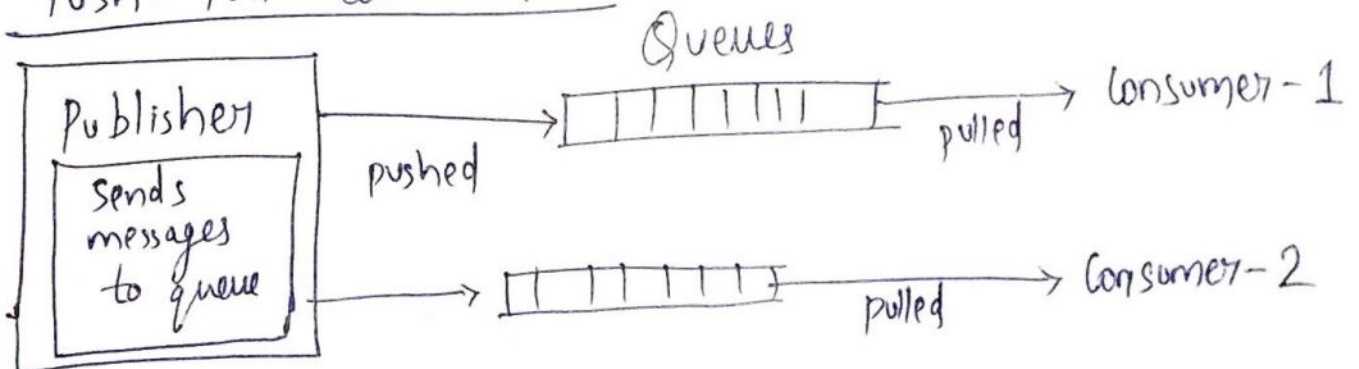
(i) Request Response commⁿ Model



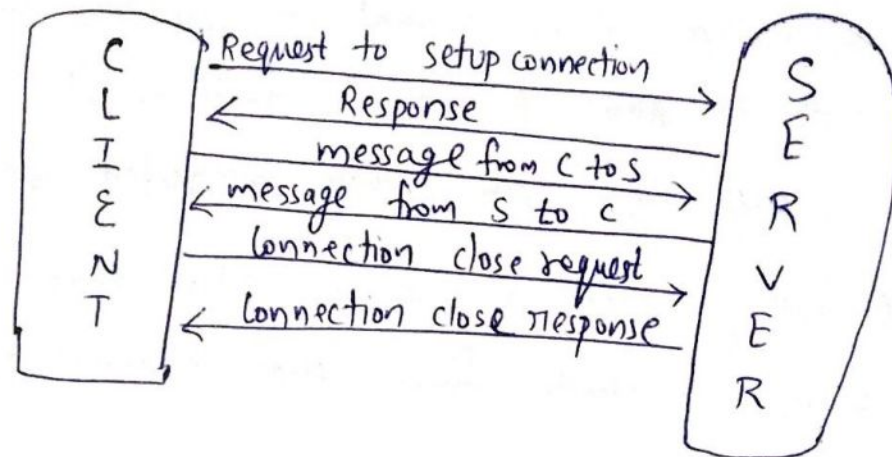
(ii) Publish - Subscribe commⁿ Model



(iii) Push - Pull commⁿ Model



(iv) Exclusive Pair Commⁿ Model :- Bidirectional, fully duplex commⁿ model




REST based commⁿ APIs

- Representational State Transfer (REST) is set of architectural principal by which you can design web services that focus on system's resources.
- Request-Response commⁿ model
- Each request involves setting up a new TCP connection
- Header overhead
- Not suitable for RTA

Web Socket based :-

- It allows bidirectional, full duplex commⁿ b/w C & S.
- Exclusive pair commⁿ model
- Single TCP connection
- No header overhead
- Suitable for RTA

M2M:- It is a direct communication system b/w the devices using wired or wireless commn channels without any human interaction. It collects the data and shares it with other connected devices. 

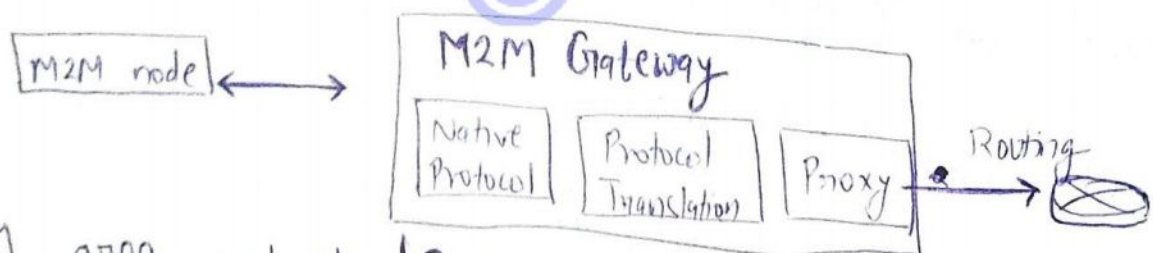
Eg- Controlling electrical devices like fans & bulbs using Bluetooth from the smartphone.

Differences b/w IOT & M2M:-

- (i) **IOT** is a subset of M2M.
- (ii) The point to point commn of M2M is the main difference b/w M2M & IOT.
- (iii) Another key difference b/w IOT & M2M is scalability.

M2M Gateways:-

- To enable the commn b/w remote M2M area networks, M2M gateways are used.



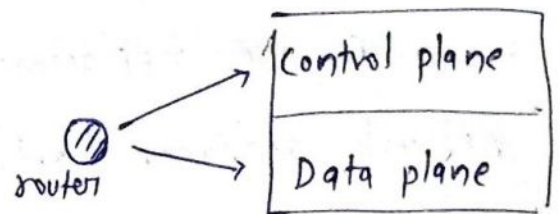
M2M area networks / Protocols:-

Bluetooth, ZigBee, 802.15.4, 6LoWPAN, M-BUS, ~~UWB~~ UWB, ModBus, Z-wave

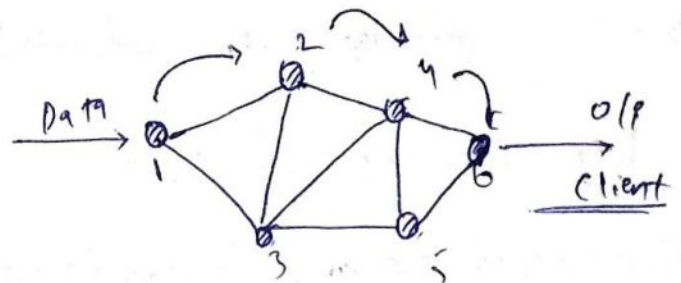
- Since non IP-based protocols are used within M2M area networks, the m2m nodes within one network can't commn with nodes in an external network.

SDN (Software Defined Networking) \Rightarrow

\rightarrow SDN is a networking architecture that separates the control plane from the data plane and centralizes the network controller.

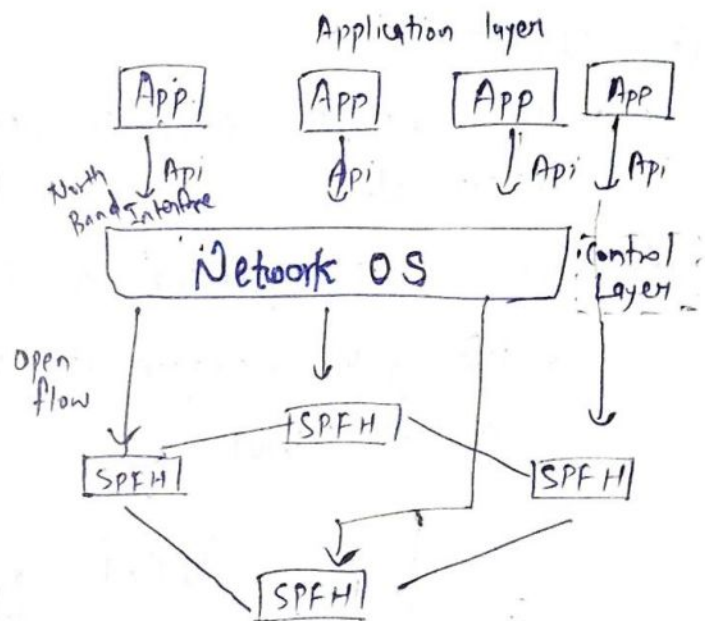


\rightarrow Software based SDN controllers maintain a unified view of network and make configuration, management & provisioning simpler.



\rightarrow SDN uses simple packet forwarding hardware as opposed to specialized h/w in conventional networks.

\rightarrow Increase the efficiency of network resource sharing & improve IoT service-level agreements.



Key Elements of SDN

① Centralized Network Controller

(Network administrators can rapidly configure the network)

② Programmable Open APIs

(Interface b/w SDN application & control layers) (North Bound)

③ Standard Commⁿ Interface (OpenFlow)

[Interface b/w control & Infrastructure Layers] (South Bound)

NFV (Network function Virtualization) \Rightarrow

- \rightarrow NFV is the replacement of network appliance hardware such as routers & firewalls with virtual machines, operating on standard servers.
- \rightarrow NFV provides the infrastructure on which SDN can run.

Key Elements of NFV:-

① Virtualized Network function (VNF)

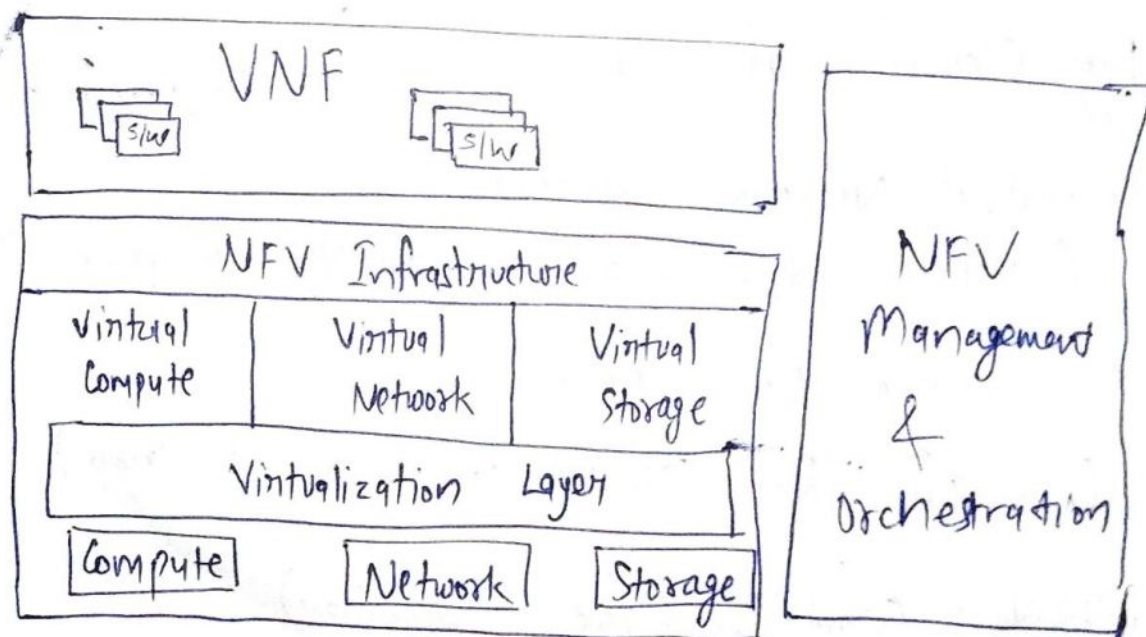
- \rightarrow It is a s/w which is capable of running over NFVI

② NFVI

- \rightarrow It includes compute, network & storage resources that are virtualized.

③ NFV Management & Orchestration

- \rightarrow life cycle management of physical & s/w resources that supports VNFs.



SDN

- SDN architecture mainly focuses on data centers.
- SDN separates control plane & data plane by centralizing control & programmability of network.
- SDN uses OpenFlow as a commⁿ protocol.
- SDN reduces cost of network because now there is no need of expensive switches & routers.
- Application of SDN:-
 - Networking
 - Cloud orchestration

Difference Similarity:-

Running on virtual machines, network controllers, gateways are deployed.

centralized control console to monitor routing.

Reduction of operational expenses through automation of network configuration.

Programmable interfaces enable provisioning of new network devices.

NFV

- NFV is targeted at service providers or operators.
- NFV helps operators to virtualize functions like load balancing, routing & policy management.
- There is no protocol determined yet for NFV.
- NFV increases scalability & agility as well as speed up time-to-market as it dynamically allot h/w a level of capacity to network functions.
- Application of NFV:-
 - Routers, firewalls
 - WAN accelerators
 - Content Delivery Networks (CDN)

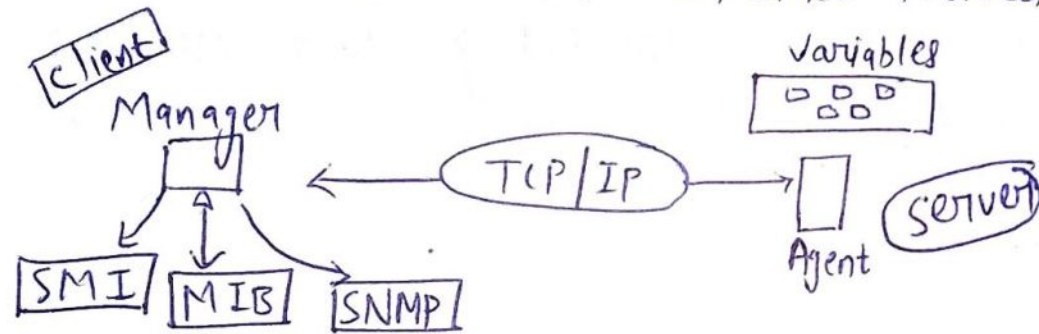
routers, firewalls are deployed as slw on top of virtualised infra.

VNF are managed & monitored regardless of where they are located

VNF eliminates need to procure h/w for individual network function.

Quickly deploy & decommission functions to support proof-of-concept trials.

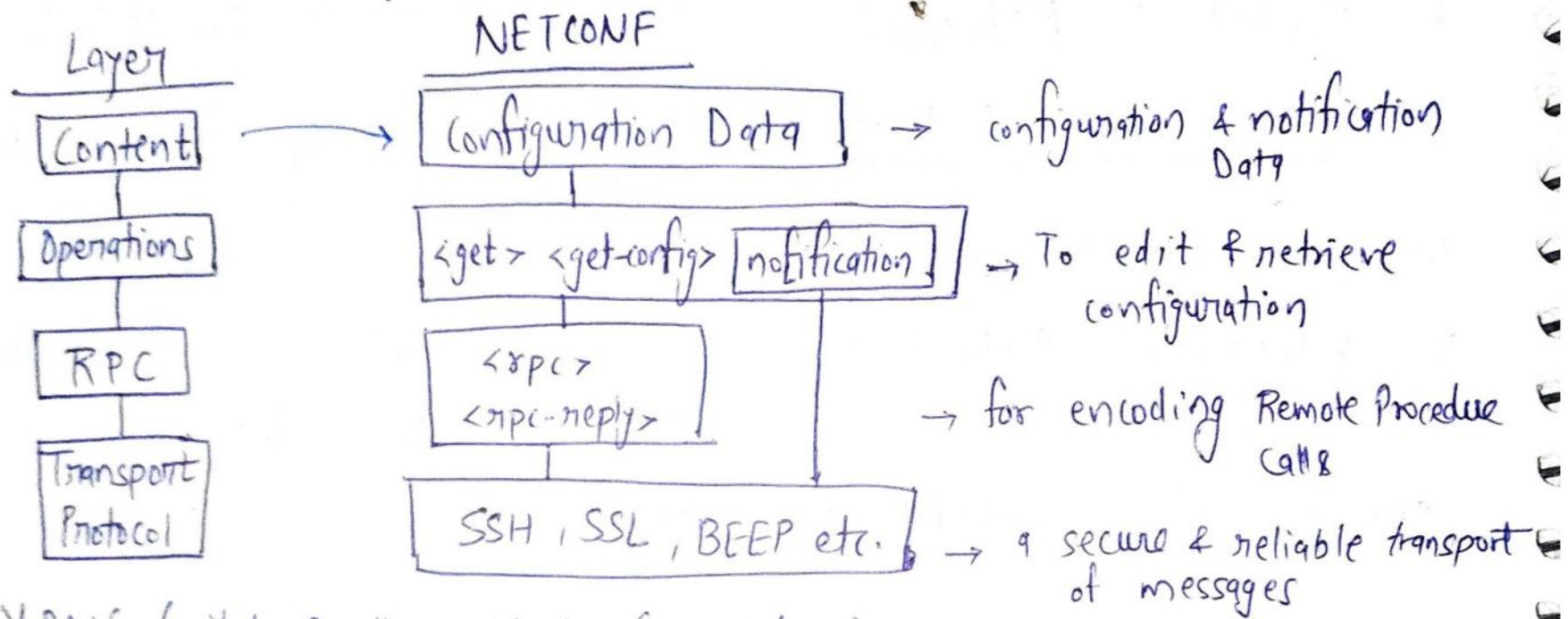
SNMP (Simple Network Management Protocol) :- It is a networking protocol used for management & monitoring of network-connected devices in Internet Protocol networks.



NETCONF (Network Configuration Protocol) :- It is a network management protocol developed & standardized by IETF.

- NETCONF provides mechanisms to install, manipulate & delete the configuration of network devices.
- NETCONF protocol uses an XML based data encoding for the configuration data.

NETCONF Layering Model :-



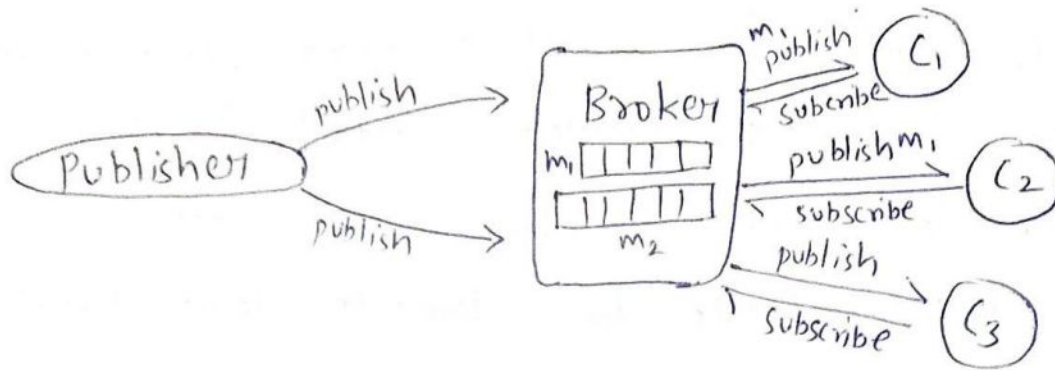
YANG (Yet Another Next Generation) :-

- YANG is a data modelling language for NETCONF.
- This can be used to model both configuration data as well as state data of network elements.
- YANG is a modular language representing data structures in an XML tree format.

SNMP → MIB

MQTT Protocols :-

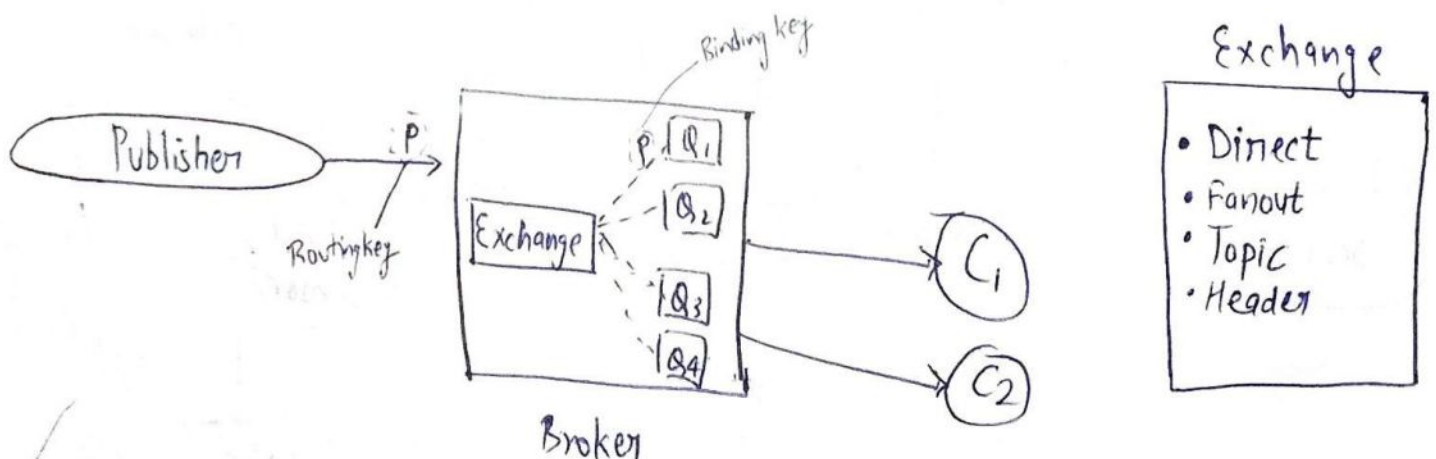
- Message Queue Telemetry Transport is an ISO standard publish-subscribe based light weight messaging protocol for use on top of the TCP/IP protocol.



- It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited.

AMQP Protocol :-

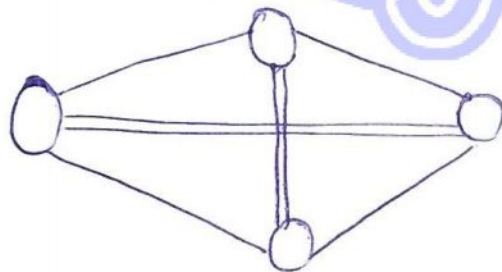
- Advanced Message Queuing Protocol is an open standard application layer protocol for message oriented middleware.
- The defining features of AMQP are message orientation, queuing, routing, reliability & security.



Zigbee :-

- Zigbee is a standards-based wireless technology developed to enable low-cost, low-power wireless machine to machine and IOT networks.
- Zigbee is built for control & sensor networks on the IEEE 802.15.4 wireless standard for WPNS.
- Zigbee provides IOT with features to support low-cost, highly reliable networks for device to device comm.
- Zigbee uses a mesh network architecture for communication.

mesh
Topology

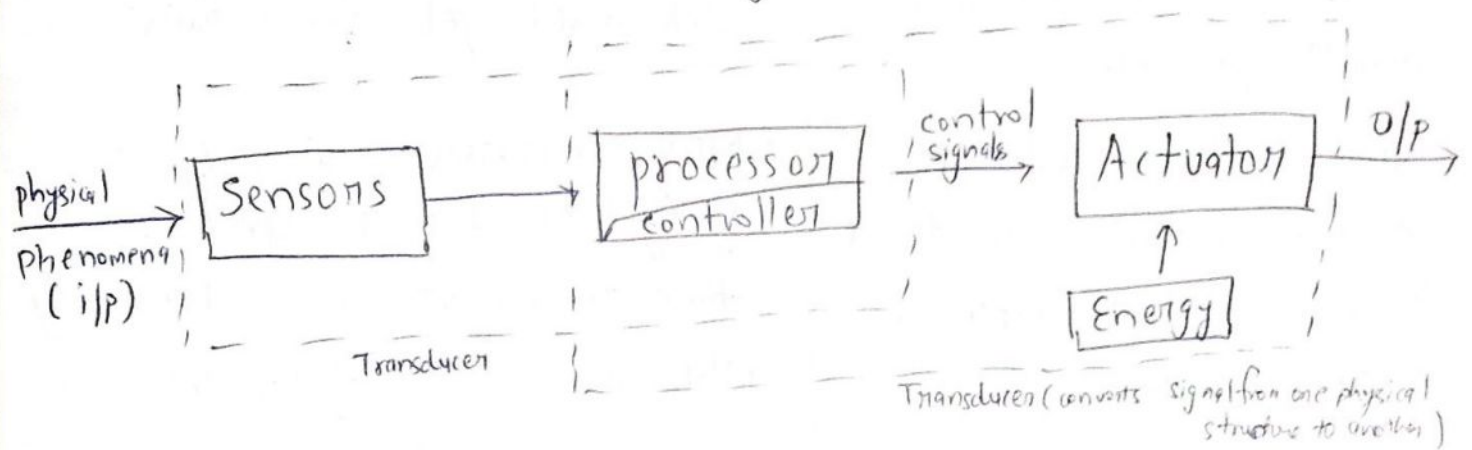


- Coordinator
- Routers
- End Devices

- Zigbee protocol defines 3 types of nodes : coordinators, routers and end devices.

Sensor :-

- Sensors are devices that detect external information, replacing it with a signals that humans & machines can distinguish.
- IOT sensors used to detect & measure various physical phenomena such as heat & pressure as well as 5 human senses: sight, hearing, touch, taste & smell.



Types of Sensors :-

1. Temperature Sensors
2. Humidity Sensors
3. Pressure Sensors
4. Proximity Sensors
5. Level Sensors
6. Accelerometer
7. Gyroscope
8. Gas Sensors
9. Infrared Sensors
10. Optical Sensors

Actuators :- An actuator is a machine component or system that moves or controls the mechanism or the system.

- Sensors in the device sense the environment, then control signals are generated for the actuators according to the actions needed to perform.

Types of Actuators:-

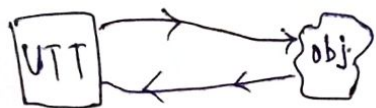
1. Hydraulic Actuators :- It uses hydraulic power.
2. Pneumatic Actuators :- It uses energy formed by vacuum at high pressure to convert into either linear or rotary motion.
3. Electrical Actuators :- It is actuated by a motor that converts electrical energy into mechanical torque.

~~DIP~~ ✓

UltraSonic Sensor

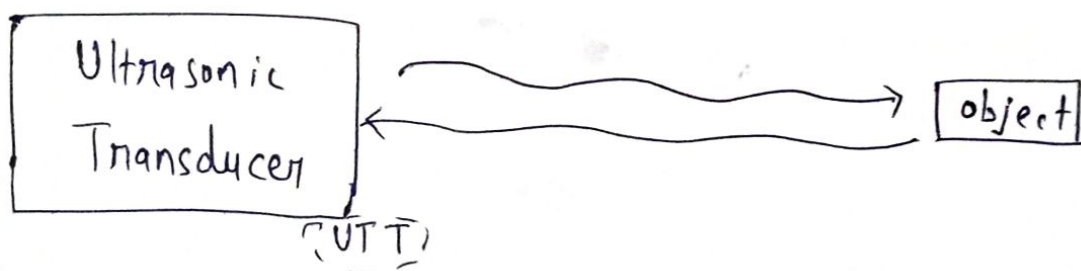
An ultrasonic sensor is a device that detects an object and measures the distance to it. It measures the distance by emitting ultrasound & receiving the wave that the object reflects.

$$\text{distance} = \frac{\text{speed of sound} \times \text{time taken}}{2}$$



IoT ultrasonic sensors are designed for non-contact detection of solid & liquid objects.

Basically used to measure fluid level, fluid identification / concentration, distance.



Ultrasonic sensors are typically used as a proximity sensor, setting a threshold distance which can determine whether an object is an obstacle.

Adv:-

- It produces ultrasonic frequencies.
- Sensors response is not dependent upon surface color or optical reflectivity.
- Does n't require much electricity.
- Sensors with digital o/p have excellent repeat sensing accuracy.

Disadv:-

- have a minimum sensing distance.
- Change in environment such as temp., pressure, humidity affect ultrasonic responses.



Temperature Sensor \Rightarrow Temperature sensors measure the amount of heat energy in a source, allowing them to detect temperature changes & convert these changes to data.

\rightarrow It requires a thermocouple or RTD.

\rightarrow The change in the temperature, corresponds to change in its physical property like resistance or voltage.

Working :-

The working of the sensor is the voltage that is read across the diode. If there is an increment in voltage, then temperature increases & there is a voltage decrement b/w the transistor's terminals of emitter & base. That data is saved by the sensor.

If the difference in voltage is amplified, then an analog signal is generated by the device & it's directly proportional to the temperature.

Application :-

- In rubber, plastic, biomedical industries.
- In mechanical engine for measuring engine oil temperature
- In electric cables for internal temperature.
- In electric motors for measuring motor winding temp, bearing temp, brushes temp.

Types of Temperature Sensor

1. Contact Type Temp. Sensor :-

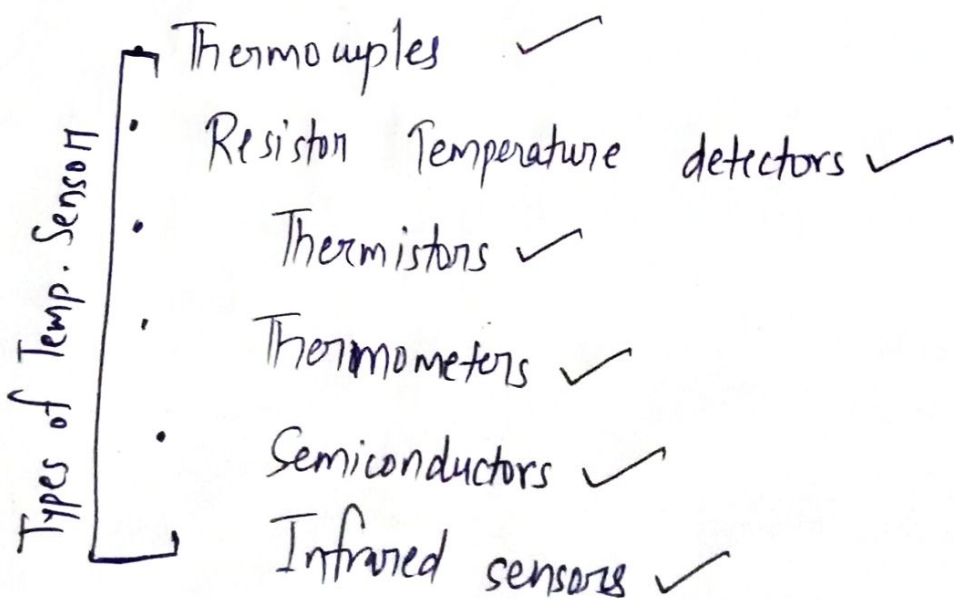
It measures the degree of hotness or coldness in an object by being in direct contact with it.

→ It's Used to detect solids, liquids or gases over a wide range of temp.

2. Non-Contact Type Temp. Sensor :-

They measure the degree of hotness or coolness through the radiation emitted by the heat source.

→ Different types of temperature sensors that have sensing capacity depending upon their range of application.



1. Thermocouples : — It is used for long distance .

A thermocouple is a voltage device that indicates temperature by measuring a change in the voltage.

- It consists of two different metals: opened & closed.

- Thermo-electric effect :- When two dissimilar metals produce a voltage, then a thermal difference exists b/w two metals. When the temp. goes up, the o/p voltage of thermocouple also increases.

2. Resistor Temperature Detector (RTD) :- It is one of the most accurate sensors.

$$\text{Resistance} \propto \text{temp. } (-270^{\circ}\text{C to } +850^{\circ}\text{C})$$

RTD requires an external current source to function properly.

However, the current produces heat in a resistive element causing an error in temp measurements.

$$\Delta T = \frac{P \times S}{\text{one squared power}} \leftarrow \text{degree C/milliwatt}$$

There are 2 wired, 3 wired, 4 wired methods to measure temp by using RTD.

3. Thermistors:- It changes its resistance when the temp changes like RTD sensor but it offers higher sensitivity.
Most of thermistors have a negative temp coefficient.
so $\text{temp} \propto \frac{1}{R}$

4. Thermometers:- It is device used to measure the temp of solids, liquids or gases.
(Heat) (measure)

Thermometer contains a liquid, which is mercury or alcohol in its glass tube.

volume of thermometer \propto temp.

→ When the liquid is heated it expands inside the narrow tube of the thermometer.

5. Semi conductor:- It comes in form of ICs.
(IC temp. sensor) (AD590 & LM35 temp sensors.)

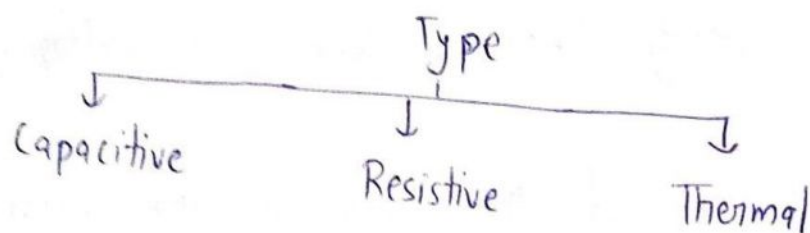
6. IR sensor:- It is used to sense certain characteristics of its surroundings by either emitting or detecting IR detection.
It is non-contacting sensors.



Humidity Sensor —

It is electronic devices that measures and report the moisture and air temperature of the environment (air, soil...).

It works by detecting changes that alter electrical current or temperature in air.



Capacitive :— A capacitive humidity sensor measures relative humidity by placing a thin strip of metal oxide b/w two electrodes. The metal oxide's electrical capacity changes with the atmosphere's relative humidity. It is linear & can measure relative humidity from 0% to 100%.

Resistive :— It utilizes ions in salts to measure the electrical impedance of atoms. As humidity changes, so do the resistance of electrodes on either side of salt medium.

Thermal :— Two thermal sensors conduct electricity based upon the humidity of surrounding air. One sensor is encased in dry nitrogen while other measures ambient air. The difference b/w the two measures the humidity.

Arduino \Rightarrow Arduino acts as the brain of the system and processes the data from the sensor.

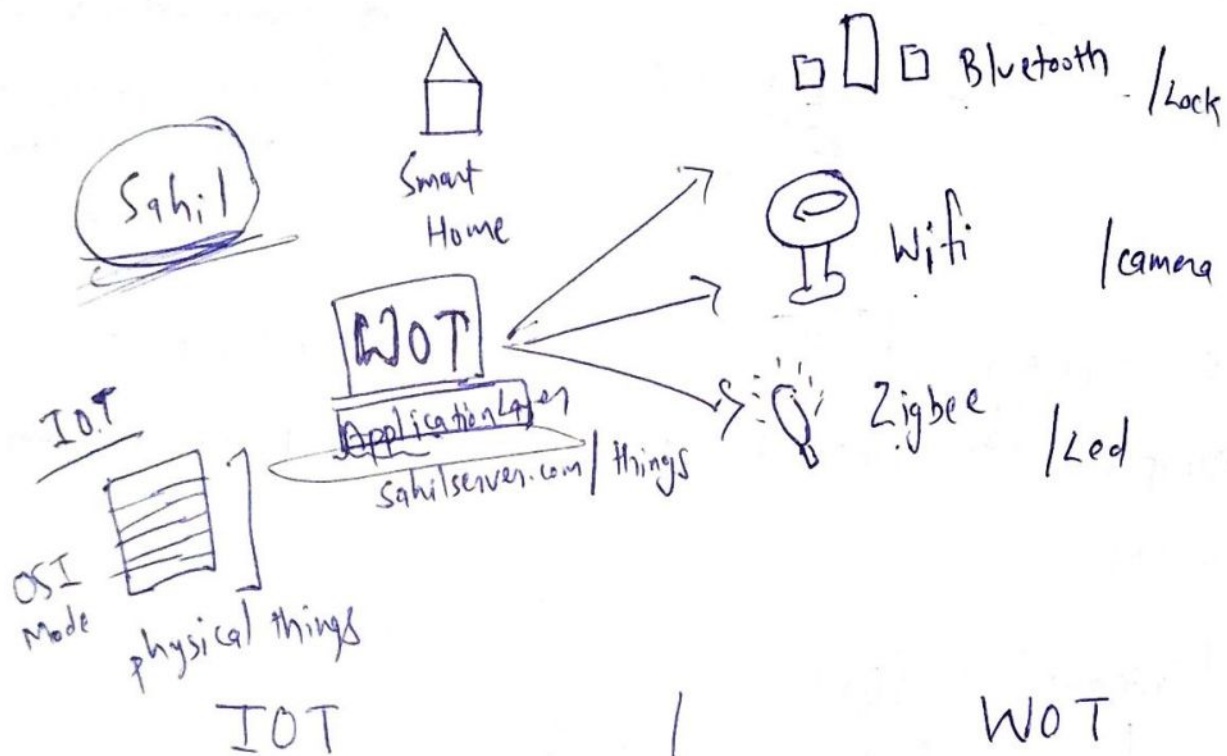
- \rightarrow Arduino is an open source h/w platform.
- \rightarrow It runs on Mac, Windows & Linux.
- \rightarrow Arduino is a microcontroller.

The Arduino board is connected to a computer via USB, where it connects with Arduino development Environment (IDE).

Raspberry Pi \Rightarrow

- Raspberry pi is a low cost mini computer with the physical size of credit card.
- Raspberry pi runs various flavors of Linux.
- Raspberry pi also allows interfacing sensors & actuators through the general purpose I/O pins.
- Since Raspberry pi runs Linux operating system, it supports Python "out of the box".

WOT → Web of things is all about making devices accessible over the web using web protocols like HTTP, WebSocket, JSON etc agnostic to anything below the Application Layer, so that any device can be part of universal • WOT regardless of what protocols it uses connect to the internet.



→ IOT is about creating a network of objects, things, system.

- No scalability security.
- Scope of IOT is broader

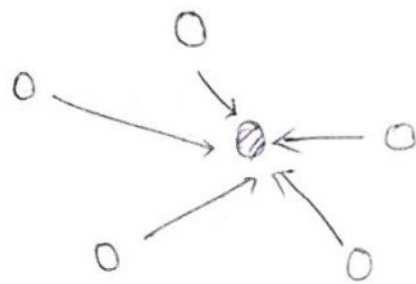
WOT

WOT tries to integrate them to web.

- Building WOT has various scalability security
- Scope of WOT includes web (WSN & RFID system)

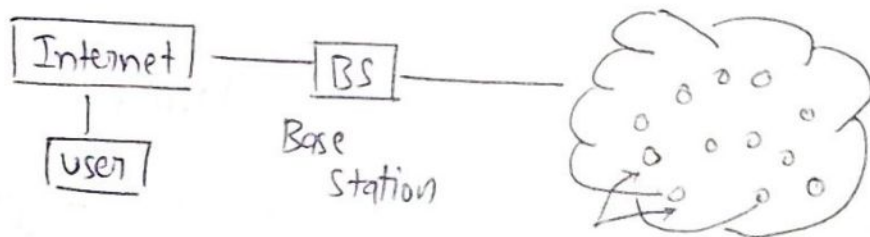
Wireless Sensor Network

- WSN are spatially distributed autonomous sensors to monitor physical or environmental conditions such as temperature, sound, pressure etc and to cooperatively pass their data through the network to a main location.
- WSN is built of "nodes" - from a few to several, where each node is connected to one sensors.



⊙ sensor network node
[Radio transceiver
Microcontroller
battery]

- The topology of the WSNs can vary from a simple star network to an advanced multi-hop WSN.



sensor node (for managing & monitoring the environment)

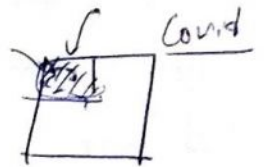
- WSN can be used for processing, analysis, storage & mining of the data.

RFID (Radio frequency Identification Technology) \Rightarrow

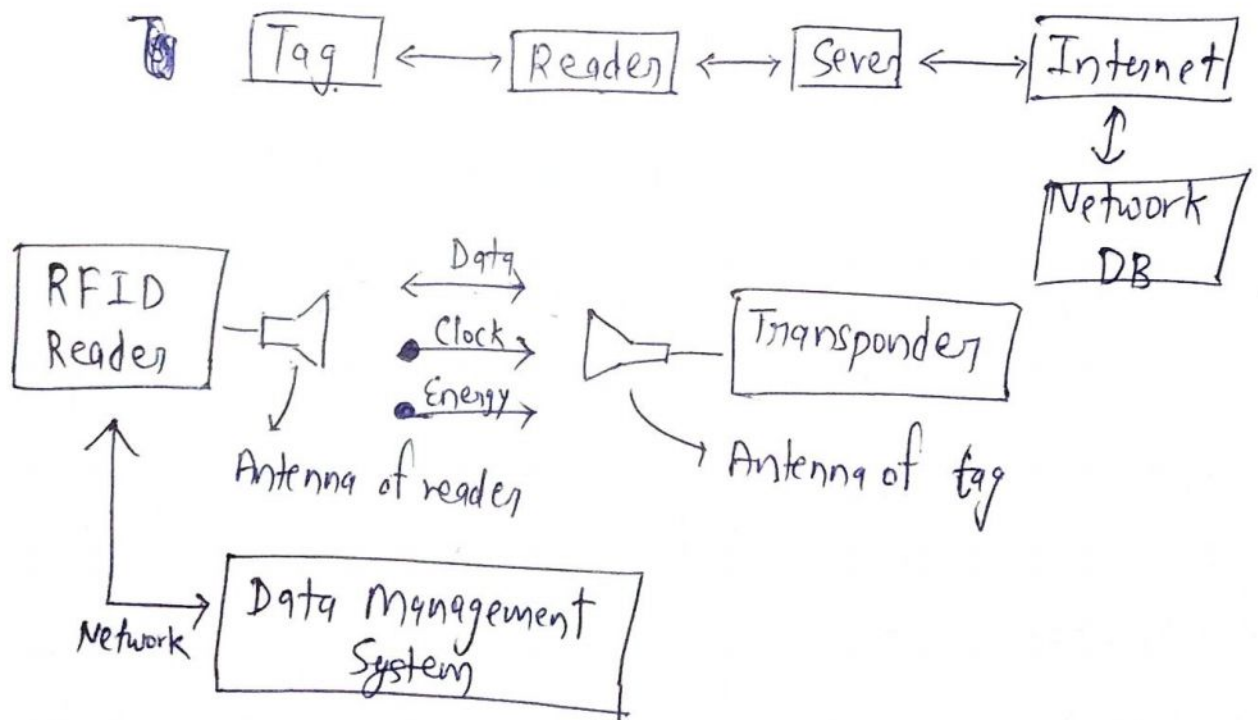
RFID tags are primarily used to make everyday objects commⁿ with each other & the main hub & report their status.

It is mainly used in Healthcare for

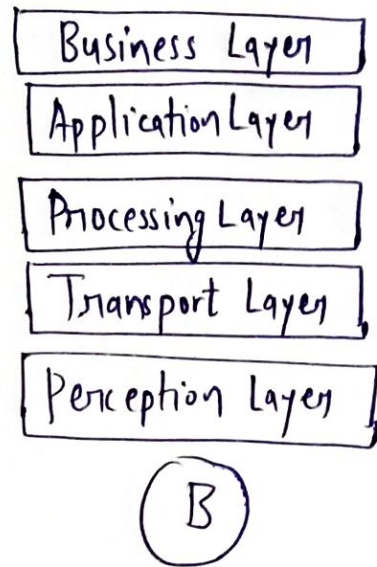
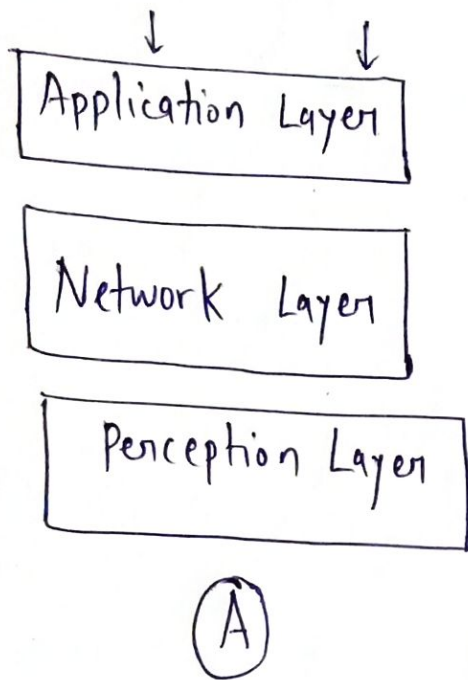
- Asset tracking & management
- Patient & staff tracking
- Infection control



\rightarrow RFID ~~tags~~ uses wireless non-contact radio frequency waves in which data is digitally encoded in RFID tags which can be read by reader through radio waves.



Three Layer and five layer architecture of IOT



- (i) The Perception Layer is the physical layer, which has sensors for sensing & gathering information about the environment. It senses some physical parameters.
 - (ii) The network layer is responsible for connecting to other smart things, network devices & servers.
 - (iii) The application layer is responsible for delivering application specific services to the users. Ex- smart homes, smart cities ...
- 3 Layer architecture is not sufficient for research on IOT because research often focuses on finer aspects of the IOT.

One is the five layer architecture, which additionally includes the processing & business layers.

→ The role of the perception & application layers is the same as the architecture with 3 layers.

The function of the remaining 3 layers are as follows:-

(i) The transport layer transfers the sensor data from perception layer to processing layer & vice versa through networks such as wireless, 4G, LAN, 3G, Bluetooth, RFID & NFC.

(ii) The processing layer is also known as the middleware layer. It stores, analyzes & processes huge amounts of data that comes from the transport layer. It can manage & provide a diverse set of services to lower layers.

It employs many technologies such as databases, cloud computing & big data processing modules.

(iii) The business layer manages the whole IoT system, including applications, business & profit models & users privacy.

Lite OS :- It is a lightweight, open source IOT device & smartphone OS.

- It is designed to have a low footprint, which saves space & reduces the load of the OS on the device.
- It supports smartphones, wearables, smart homes, Internet of vehicles.
- It simplifies IOT device development & connectivity while focusing on enhancing user experience.
- The smallest kernel (6KB) on market offers fast-start & low power consumption feature.

Riot OS :- It is an open source embedded OS.

- It is designed for networked & memory constrained systems.
- Based on microkernel architecture & written in ANSI C.
- features →
 - Modularity
 - Tickless Scheduler
 - Straight forward interrupt handler
 - Support various h/w vendors
 - Reliability & real time features

Contiki OS :- → It is an open source OS for IOT.

- It supports full standard IPv6 & IPv4 along wireless standards 6lowpan, CoAP.
- It can fit into 10KB of RAM & 100KB of ROM.
- Application →
 - small web browser
 - web server
 - calcu.
 - shell
 - email client

- Tiny OS - It is a free open source OS. Designed for WSN.
- Tiny OS began as a collaboration b/w universities.
 - It features a component based architecture.

features →

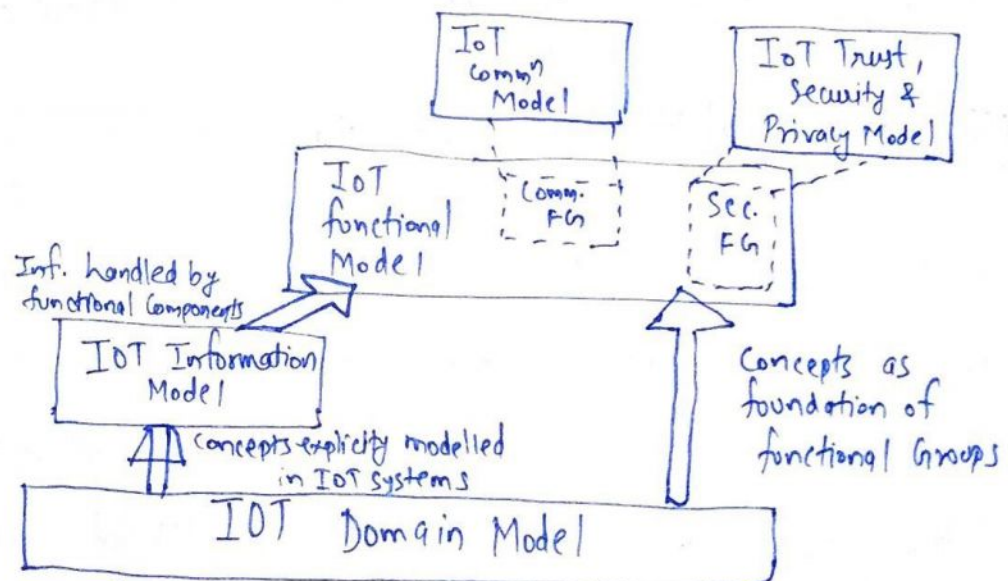
- Completely non blocking.
- Tasks are non preemptive & run in FIFO order
- Statically linked.
- Event based execution model means no user/kernel boundary & hence supports high concurrency.

Models →

- Data
- Thread
- Program

IOT Reference Model & Architecture →

- IOT Reference Model describes the domain using a no. of sub models.
- It aims at establishing a common grounding & a common language for IOT architectures & IOT systems.



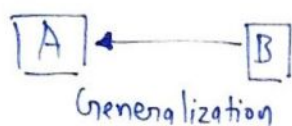
- The foundation of IOT reference model is the IOT domain Model, which introduces the main concepts of IOT devices, IOT services & Virtual Entities (VE).

IOT domain Model concepts are independent of specific technologies & use cases.

3 kinds of device types for IOT domain Model:-

1. Sensors
2. Actuators
3. Tags - In general identify, the Physical entity that they are attached to. It can be devices or physical entities but not both as domain model shows.

Model notation & Semantics:-

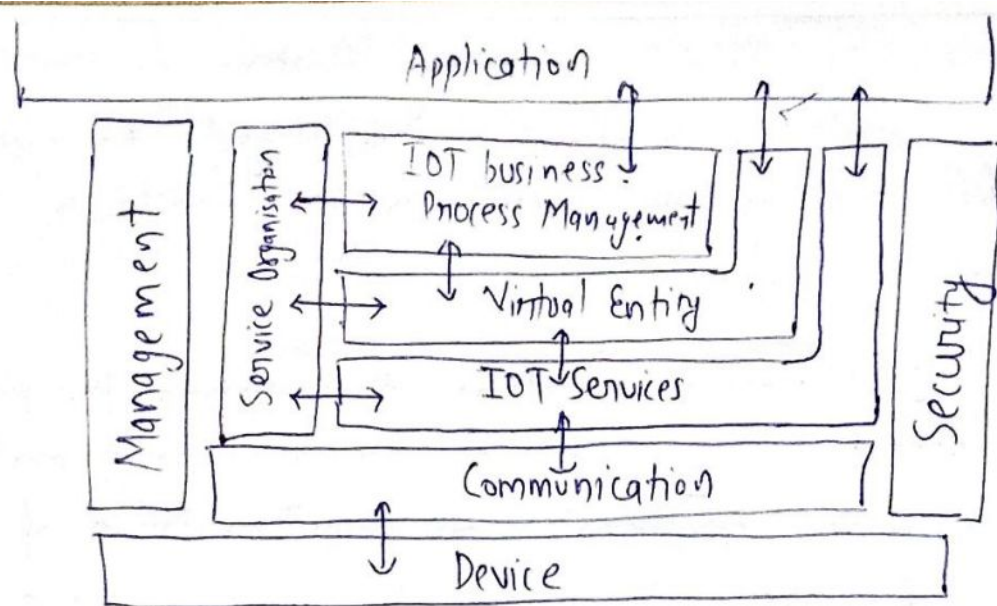


Association
Composition

IOT information Model:- It captures the details of a virtual entity centric model. Similar to the IOT domain Model, It is represented using unified Modelling Language (UML) diagrams.

IOT Functional Model:- It aims at describing mainly functional groups and their interaction with ARM, while functional view of a reference Architecture describes the functional components of FGs, interfaces & interactions b/w the components.

→ The functional view is typically derived from the functional model in conjunction with high level requirements.



Device Functional Group contains all possible functionality hosted by physical devices that are used for increment in physical entities. It includes sensing, actuation, processing, storage & identification components.

Commⁿ functional Group abstracts all possible commⁿ mechanisms used by relevant devices in an actual systems in order to transfer information to digital world components or other devices.

IOT Service F.G. mainly to service class from IOT domain Model & contains single IOT services exposed by resources hosted on device or in network.

Virtual Entity F.G. corresponds to virtual entity class in IOT DM. It contains necessary functionality to manage associations b/w virtual entities with themselves as well as b/w VE & related IOT services.

IOT service organisation F.G. is to host all functional components that support the composition of IOT & VE services.

IOT Process Management F.G. is a collection of functionality that allows smooth integration of IOT related services with business process.

Management FG:- It includes necessary functions for enabling faults & performance monitoring of the system, configuration for enabling the system to be flexible to changing user demands.

Security FG:- It contains functions that ensure the ~~secure~~ secure operation of system as well as management of privacy. It contains components for authentication of users, authorisation of access to services by users, secure commⁿ b/w entities of the system such as Devices, services app.

Application FG:- It is just a placeholder that represents all the needed logic for creating an IoT application. It contains logic tailored at specific domain such as smart grid.

IoT Communication Model & Trust, security, Privacy Model

Safety:- IoT reference Model can only provide IoT related guidelines for ensuring a safe system to extent possible.

Privacy:- Protecting the user privacy is of utmost importance for an IoT system.

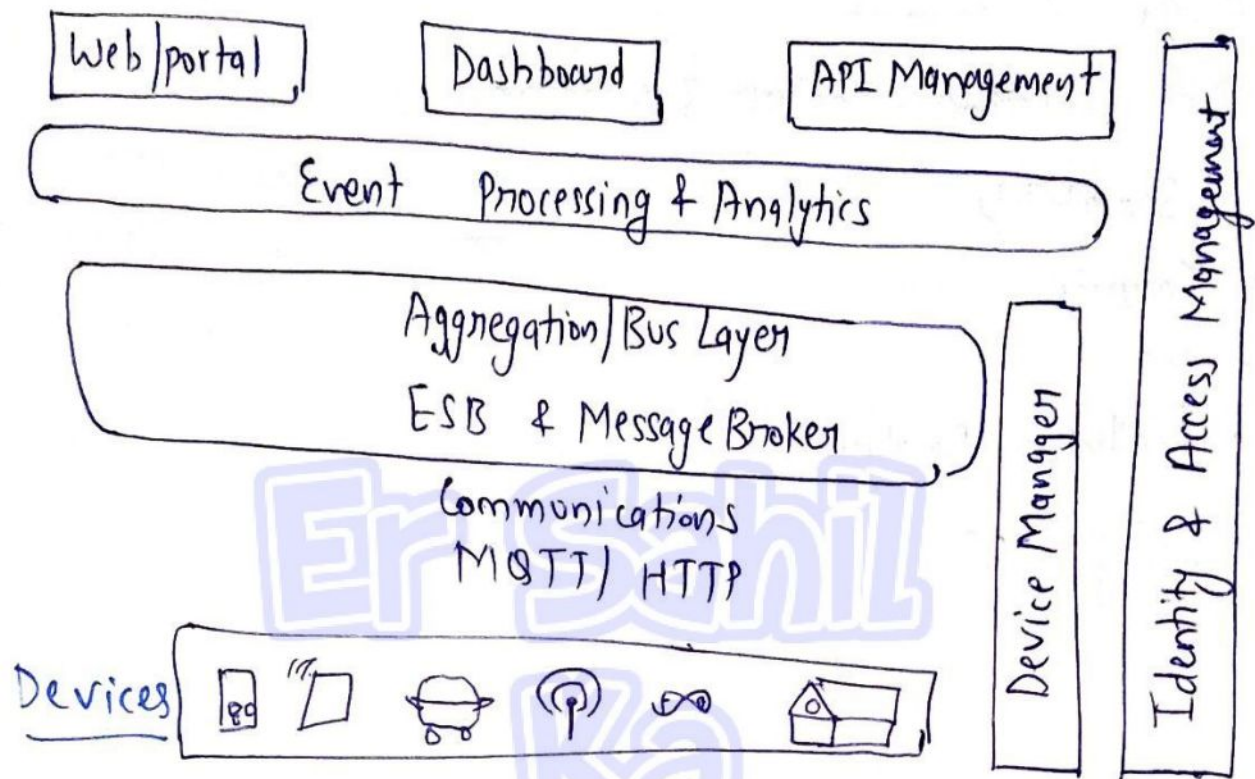
It contains following functional components : Identity Management, Authentication, Authorisation & Trust & Reputation.

Trust:- An entity is said to 'trust' a second entity when first entity makes the assumption that second entity will behave exactly as a first entity expects.

Security:- It focuses on confidentiality & integrity protection of interacting entities & functional components.

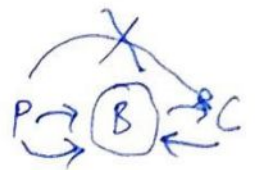
IoT Reference Architecture:-

The reference architecture consists of a set of components. Layers can be realized by means of specific technologies.



The Layers are:-

- Client commⁿ - Web/portal, Dashboard, APIs
- Event Processing - (Data Storage) include
- Aggregation/Bus - ESB & Message Broker
- Relevant transports - MQTT/HTTP/XMPP/CoAP ✓
- Devices



Cross cutting Layers are:-

- Device Manager
- Identity & Access Management

Requirements for a Reference Architecture:-

- Connectivity & communications
- Device Management
- Data collection, analysis & activation
- Scalability
- Security
- HA
- Predictive Analysis
- Integration

**Er Sahil
Ka
Gyan**

Some topics are in other pdfs so kindly have a look into it also...

IoT. Operating Systems : -

IoT OSs run on IoT devices and connects to a large network of devices.

It communicate with IoT device management s/w.

• Some of best IoT operating systems as follows: -

1. LiteOS

2. Tiny OS

3. RIOT. OS.

4. Contiki OS

5. Ubuntu Core

6. Window 10 IoT.
and many more.

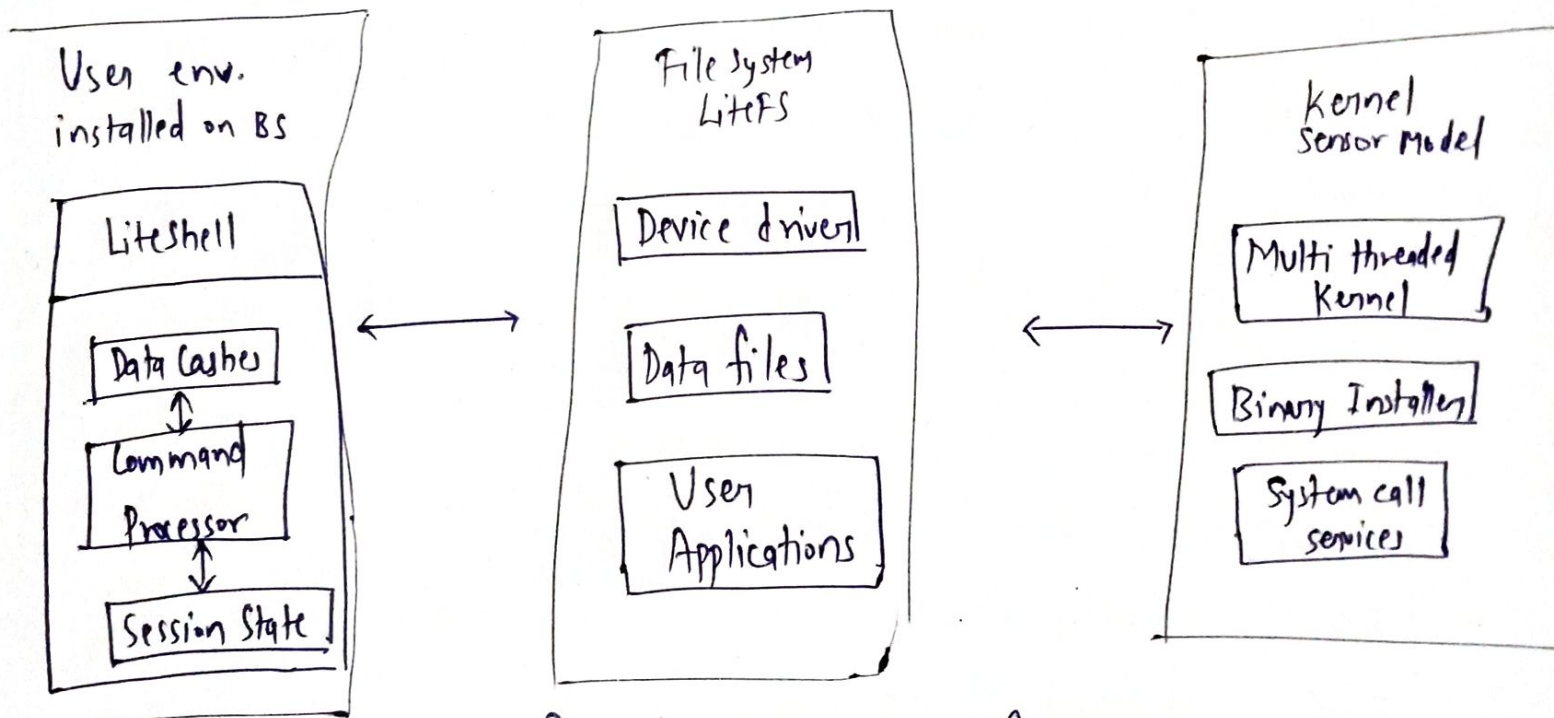


fig:- Architecture of LiteOS

Lite OS :- It is a lightweight, open source IOT device & smartphone OS.

- It is designed to have a low footprint, which saves space & reduces the load of the OS on the device.
- It supports smartphones, wearables, smart homes, Internet of vehicles.
- It simplifies IOT device development & connectivity while focusing on enhancing user experience.
- The smallest kernel (6KB) on market offers fast-start & low power consumption feature.

Riot OS :- It is an open source embedded OS. (friendly OS of IoT)

- It is designed for networked & memory constrained systems.
- Based on microkernel architecture & written in ANSI C.
- features →
 - Modularity ✓
 - Tickless Scheduler ✓
 - Straight forward interrupt handler ✓
 - Support various h/w vendors ✓
 - Reliability & real time features ✓

Contiki OS :- → It is an open source OS for IoT.

- It supports full standard IPv6 & IPv4 along wireless standards 6lowpan, CoAP.
- It can fit into 10KB of RAM & 100KB of ROM.
- Application →
 - small web browser ✓
 - web server ✓
 - calcu. ✓
 - shell ✓
 - email client ✓

Tiny OS: - It is a free open source OS. Designed for WSN.

→ Tiny OS began as a collaboration b/w universities.

→ It features a component based architecture.

features →

- Completely non blocking.
- Tasks are non preemptive & run in FIFO order.
- Statically linked.
- Event based execution model means no user/kernel boundary & hence supports high concurrency.

Models →

- Data ✓
- Thread ✓
- Program ✓

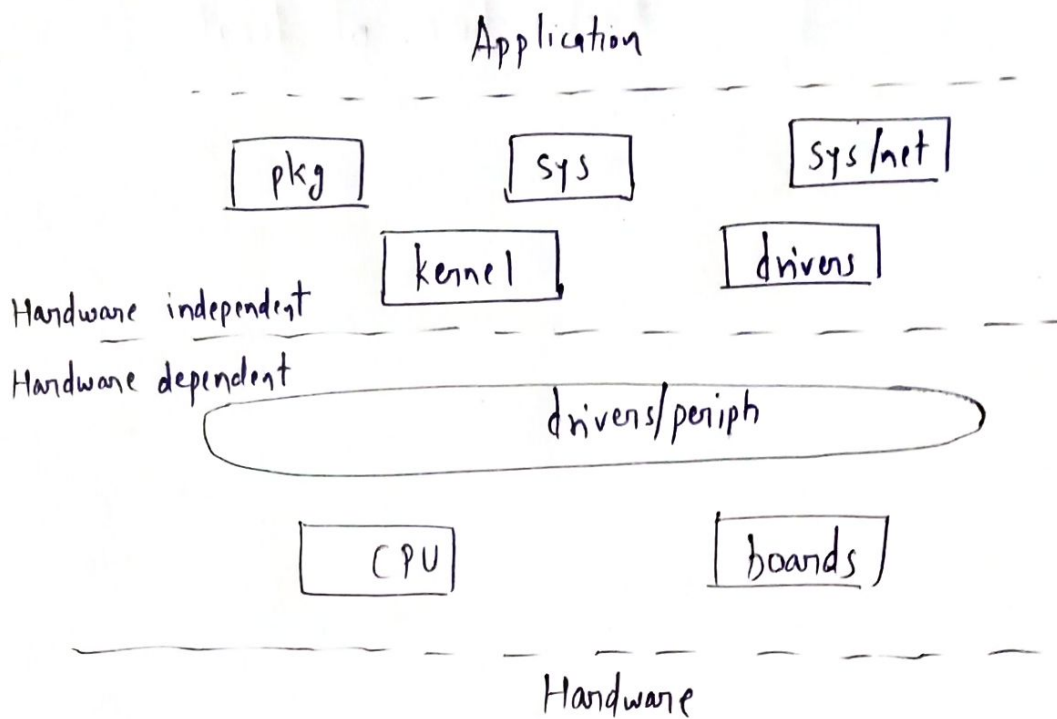


fig:- Architecture of RIOT OS

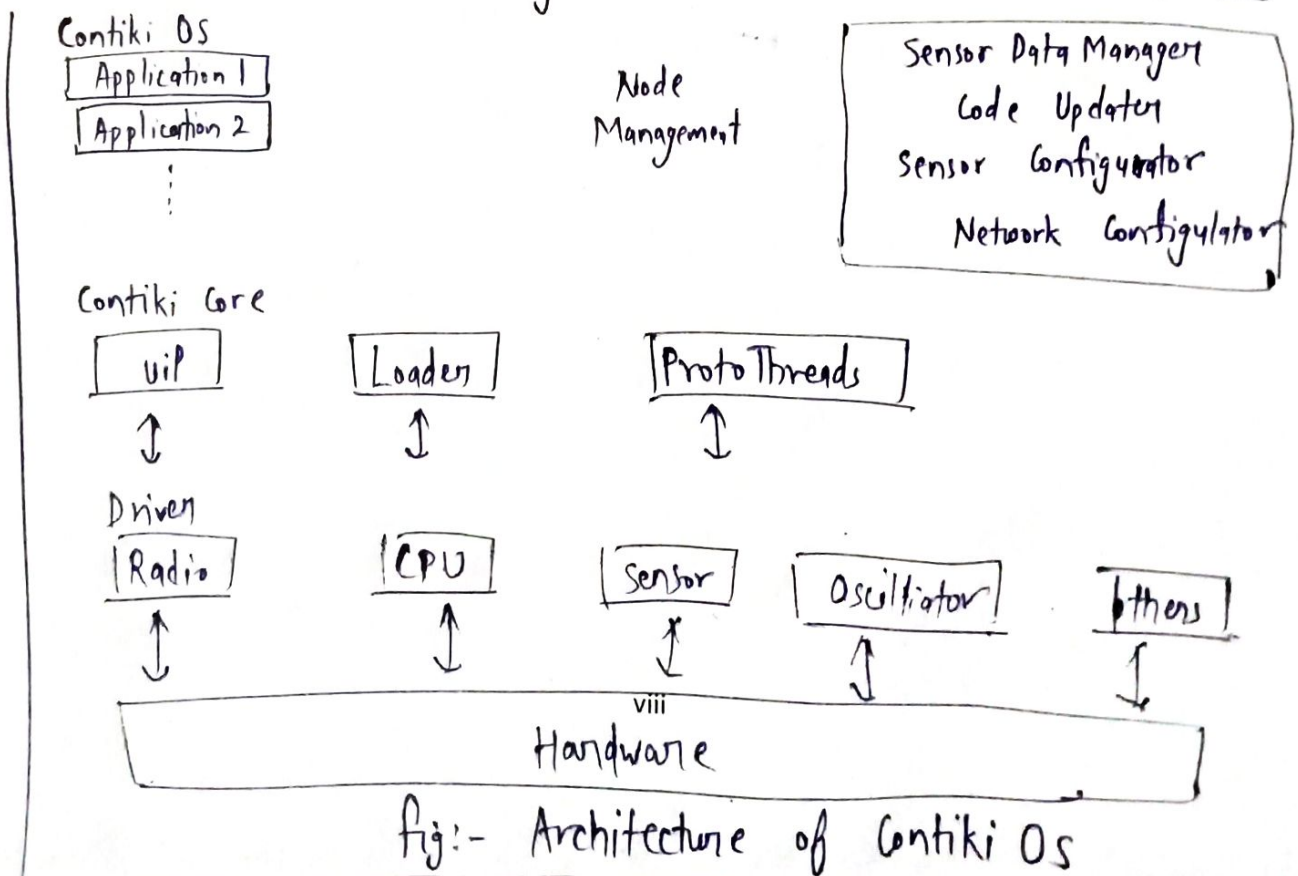


fig:- Architecture of Contiki OS

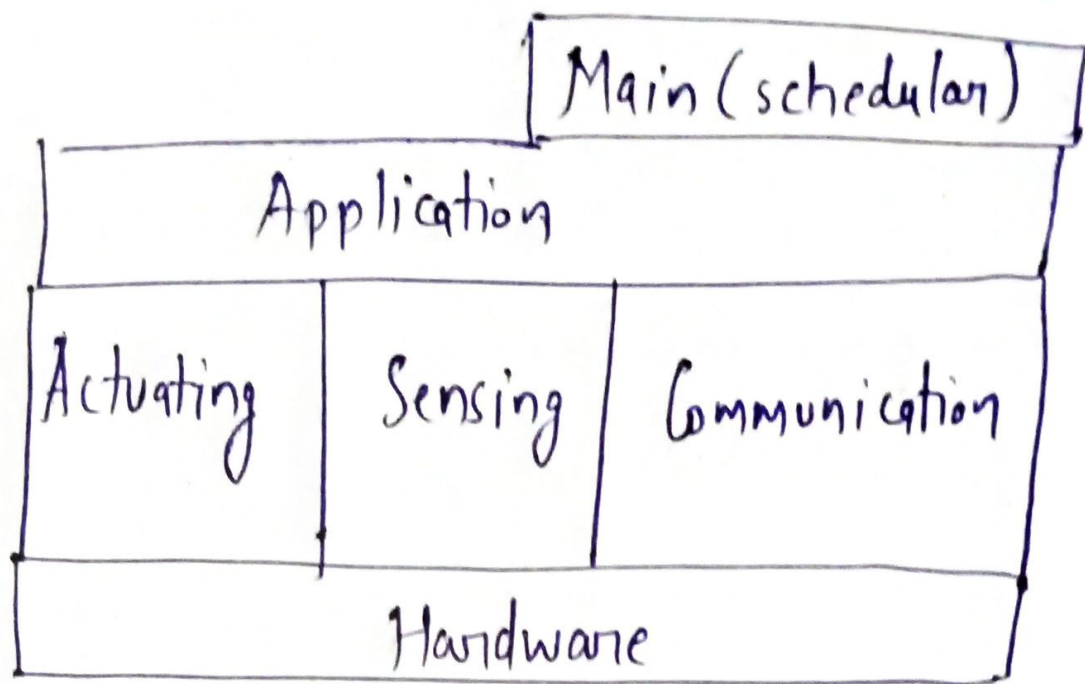


fig:- Architecture of TinyOS

some topics in other pdfs so kindly have a look on those too.