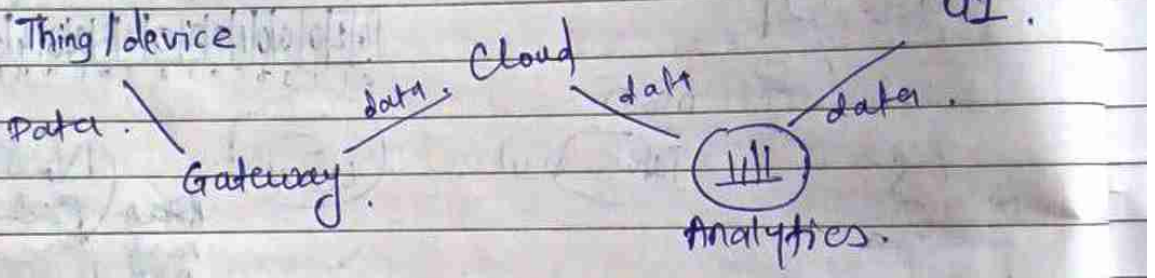
PAGE No.
DATE / /

Mod 4: Design & Dev of AI enabled IOT applications.

* **IOT interfacing**: IoT interfacing refers to how devices & systems in internet of things (IoT) communicate, interact & share data with each other or with outside world. (It's about connecting sensors, devices or controllers to each other or to cloud).
Key components:- H/w interfaces, S/w interfaces, Protocols, Cloud Services.

- Major components of IoT



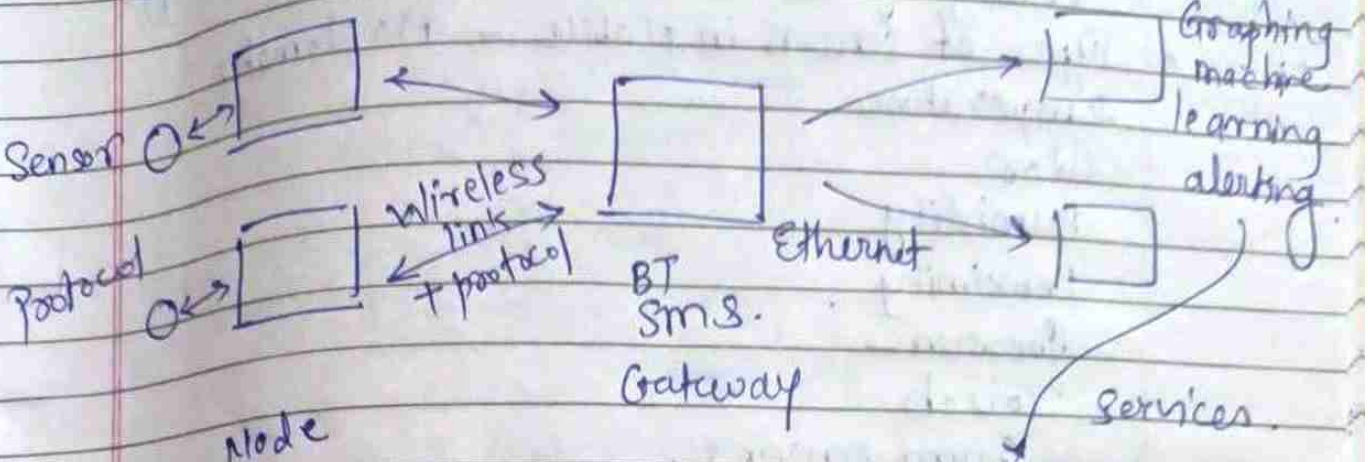
* Things:-
① Physical: exist in phy world, capable of being sensed actuated and connected. Eg: env, industrial robot electrical equipments, etc
② Virtual: exist in inf world & are capable of being stored, processed and accessed. Eg: appln s/w.

- IoT system design cycle:-

Connected devices (sensors) → HUB/Gateways → Cloud → Service (Statistics)
Companies ⇒ third party Companies.
information flux.

PAGE No.
DATE / /

IoT Architecture :-



Node
(controller, memory & power management)

refers to system that combines graphing (visualizn) & machine learning to monitor data & generate alerts when unusual patterns anamolies are detected. predefined conditions

* Sensors :-

Sensor measure or identify a particular quantity. convert physical quantities to electrical signals.
Types :- Analog & Digital.
Analog :- Signals from analog sensors are smooth & continuous
Digital :- " digital i/ps have finite no. of possible value

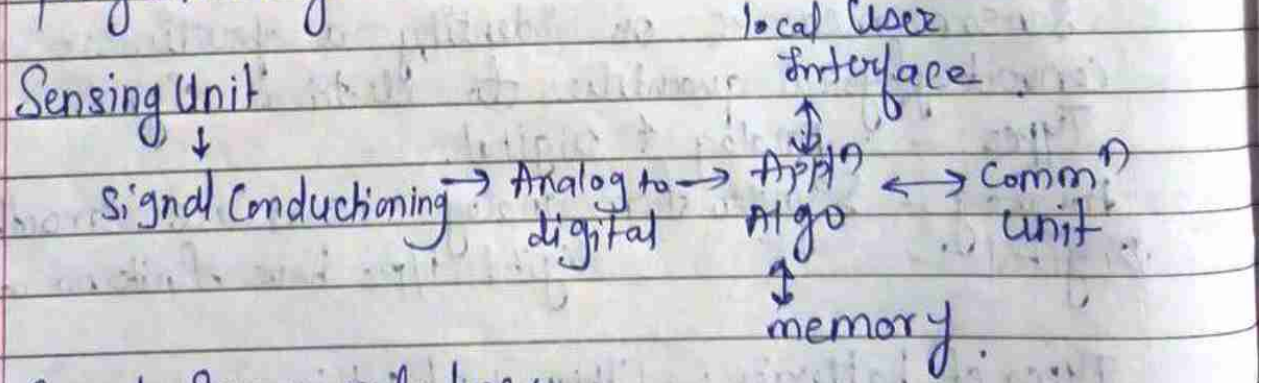
Types of batteries :- Lithium ion/ Lithium polymer, Pb-Acid (lead Acid) NiCd (Nickle Cadmium), NiMH (Nickle metal hydride).

Li-ion / Li-Poly : most popular for portable & wearable IoT
- Highest energy density, low maintenance, Ease of handling

PIR → passive infrared sensors.

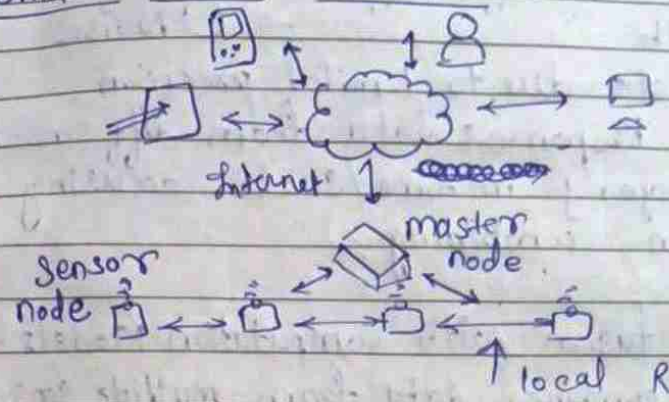
- Accelometer sensor → used for tilt screening appln (racing games in mobile)
- Types of sensors in mobile :- Accelometer, temperature
  Color
  Humidity
  Proximity
  Pressure
  Touch
  Gyro sensers.
  Magnetometer sensors.

* Smart Sensor :- Sensores with integrated electronics that can perform data conversion, Bidirectional commⁿ, take decisions & perform logical operⁿ.
  - A sensor with built-in IC (microcontroller & sensor) which provides physical parameter as o/p on ~~comp~~ connecting it to a supply voltage & programming it.



Smart Sensor node has :—
(1) Sensors (2) Data acquisition unit
(3) Embedded damage detection algo (4) Wireless Radio.

- Smart Sensor Network :—



- Wireless Sensor N/w ⇒ (WSN) Network of sensor nodes which connect wirelessly. Nodes have capability of computation, data compaction, aggregation & analysis, comm^n & n/w. Each node has independent computing power and capability to send & receive responses, data forward and routing capabilities.

- Sensing Circuit :— Circuit i/p receives o/p of sensor/transducer. Circuit receives energy in form of variation in currents, voltages, their phase angles & freq.

Sensor → Serial port interface & subcircuits → microcontroller

* Actuator :— A device that takes the actions as per the input command, pulse, state (1/0) or control signal. An attached motor, speaker, LED or an o/p device converts electrical energy into physical action.
- Piezoelectric vibrator: Pizoelec. crystals when applied varying electric voltages at i/p generate vibrations.
- Motor can be dc/ac. Relay switch: Electric Switch which can be controlled by 1/0 from port pin of microcontroller.

PAGE No.
DATE / /

* Working with low cost sensor :-
2 problems with " " " :- ① Noise
fluctuating data due to limited precision.
② Non-aligned Response: data from diff sensors may not align properly in time/scale causing difficulty in combining.

Soln: Sensor fusion with Computation Models →
Sensor fusion combines data from multiple sensors to improve accuracy & reliability. Computational models (like ML Algos) process raw sensor data to filter noise & align responses.

* IoT - Multi Sensor Systems :- uses multiple sensors
1) Classical Multi Sensor system: combine diff types of sensors such as pressure sensors, motion, sensor. Each sensor provides complementary data.
2) Multi-Sensor Joint System: A system with multiple IMUs (Inertial measurement Units) that are not rigidly connected. Useful in robotics, where sensors on diff limbs works together to understand robot's motion
3) Co-Located Multi Sensor Sys :- Setup with array of well-placed IMUs located in close proximity to each other. Coordinating positioning of these sensors allows for → better spatial resolution
use :- Wearable device where multiple IMUs placed on diff parts of body provide detailed motion analysis.

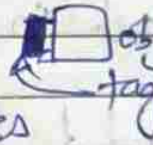
IoT Sys Component.
- IoT System compatibility
- digital Compatibility → Sensing element, Sensing circuit.

* Control Unit :—

- System on Chip → A circuit on a single chip, consisting of multiple processors, h/w units and embedded software
  Arduino, Raspberry Pi

- Node MCU : is a low cost open source IoT platform. It initially included firmware ↑ h/w based on ESP-12 module. Later support for ESP32 32bit MCU was added
  memory : 128 KB
  developer : ESP8266 opensource community

- (RFID) Radio freq. identification with IoT. for tracking ↑ comm°. Uses radio waves to transmit data betn a tag attached to an obj ↑ a reader. RFID tags → contain data about the object. RFID reader → reads data from tag. Antenna : comm° betn Reader ↑ tags.



* Interfacing Sensors to Micro controllers :—
Communication protocols :—

1. Parallel : multiline channel with each line capable of transmitting several bits of data simultaneously. (8 or 16 wires)
2. Serial : Stream their data, one single bit at a time. Most h/w interfaces are serial interfaces sacrificing potential speed in parallel. Serial interfaces generally use multiple wires to control flow ↑ timing of binary info along primary data wire.

PAGE No.
DATE / /

* IoT h/w platforms use diff common interfaces. Sensor & actuator can support one or more of these interfaces :—

(1) USB (Universal Serial Bus) : Connect elec. device to micro controller.

(2) GPIO (General purpose i/p o/p pins) : generic pin on IC or comp board whose behaviour is controlled by user at runtime. 2 States : High & low.

(3) Inter Integrated Circuit Serial Bus (I2C) : uses a protocol that enables multiple modules to be assigned a discrete address on bus. (protocol)

(4) SPI → Serial Peripheral Interface / Interchange : Bus devices employ a master-slave architecture (protocol)

(5) Universal Asyn Receiver / Transmitter (UART) : It is required when serial data must be laid out in memory in a parallel fashion

(6) Recommended Standard 232 (RS232) : is used for obtaining communication bet computer & circuit in order to transfer data.

Cloud Computing

- Cloud computing is a service provisioning technique where computing resources like hardware (servers, storage) + softwares + complete platform for developing appln are provided as a service by cloud providers to the customers.
- Pay as you use principle.
- customers don't have to invest money to purchase manage, scale infrastructures, s/w upgradations or licensing.

* Cloud Service Models:—

1) IaaS (Infrastructure as a Service): Service provider provides own hardware. Client uses this and pays on per-use basis.
   Eg: EC2, S3 service (Simple Storage Service)

2) Platform as a Service (PaaS): complete resources needed to design, develop, test, deploy + host an appln are provided to customers. In addition to h/w, it also include s/w and configuration required to create applcn.
   Eg: Google App Engine.

3) SAAS: S/w as a service:- Directly provides software service over the internet to use, buy, install, etc.
   Eg: Google docs. (we can only use appln).

PAGE No.
DATE / /

- Cloud deployment Models:-

1) Public cloud: Services & infrastructure are hosted off-site by cloud provider & easily accessible to general public via internet.
2) Private cloud: Services & infra. are operated for a single operation accessible via private network, managed internally. greater level of security.
3) Community cloud: accessible by a grp of organizations.
4) Hybrid: mixture of public & private cloud. All critical and sensitive appl^n & data are stored in private cloud & non-critical & non sensitive appl^n and data are stored in public cloud

- features of Cloud computing:-

- It is elastic: (flexible, can scale up & down) i.e increase resources as well as decrease.
- Pay as per use
- Operation: Services are completely handled by the provider.
- Reduce capital cost: No need to invest money on purchasing & maintaining h/w
- Remote accessibility: Users can use appl^n & data stored on cloud from anywhere any time through internet
- Better use of IT Staff: Staff no need to worry on purchasing & maintaining of servers, softwares, etc. instead they can concentrate on work.

Cloud Services Examples:—

1] EC2 → Elastic compute cloud (IaaS from AWS)
- web service that provides a computing capacity in the form of virtual machines. EC2 allows users to launch instances on demand using simple web based interface. It provides high memory, high CPU resources, etc. Instances can be launched with variety of operating systems.
- pay as per use
- can select storage capacity as per need.

2] Google Compute Engine:- (IaaS from Google)
- it provides virtual machines of various computing capacities ranging from small instances to high memory machine types.

3] Windows Azure: (IaaS) → provides virtual machines (VM). of various computing capacities ranging from small instances to extensive machine types

4] Google App Engine: Platform as a Service
- Cloud based web service for hosting web appln and storing data. allows users to build scalable and reliable applications that run on same systems that power Google's own appln.
- provides SDK for dev web appln.
- Developers can develop & test their applications with GAE SDK on local machine & then upload it to GAE with a simple click of button.
- Scalable & auto load balancing
- Supports apps written in several prog. languages

- GAE provides free computing resources up to certain limit. Beyond that, users are billed based on amt of computing resources used such as amt of bandwidth consumed, data stored, etc.

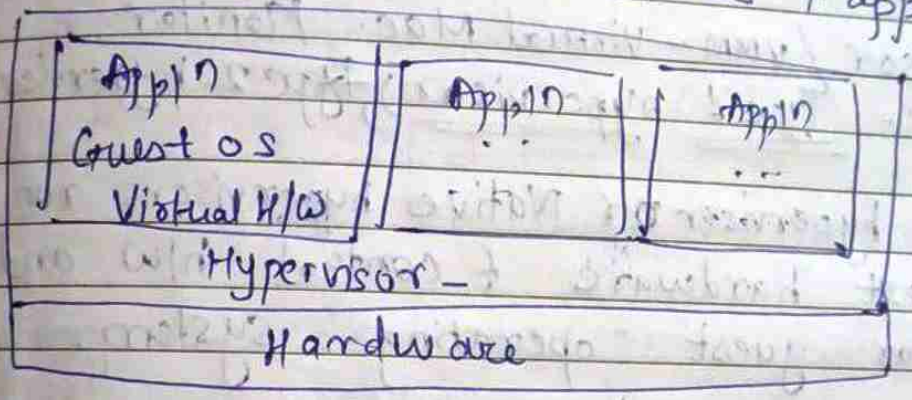
5] SaaS - Salesforce
- Salesforce Sales Cloud is a cloud based Customer Relationship Management (CRM) SaaS offering.
- Users can access CRM appln from anywhere thr. internet. Sales Cloud allows sales representatives to manage customer profiles, track opportunities optimize campaigns, etc.

6] Salesforce Service Cloud (SaaS)
Cloud based Customer Service Management SaaS: Service cloud provides companies a call center like view & allows creating, tracking, routing & escalating cases.

4] Salesforce Marketing Cloud (SaaS) → based on social marketing. it identify sales leads from social media, discover advocates, identify most trending info on any topic. It allows companies to pro-actively engage with customers, manage social adv. campaigns & track performance

* Virtualization:- technology that makes it possible to run multiple appln & various operating systems on same server at same time.
- increases h/w utilization, saves energy & cost

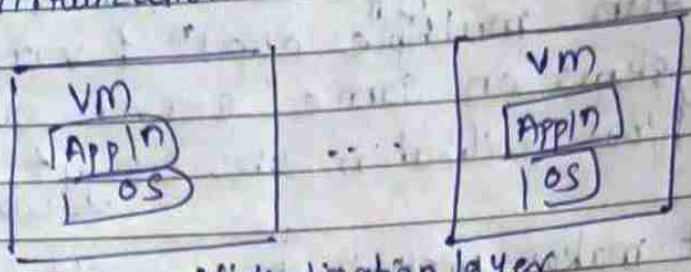
- Software that makes virtualization possible is known as a Hypervisor. It sits betn h/w and os and assigns amt of access that appln & os have with processor & other h/w resources. Hypervisor assigns the cpu & other h/w resources needed to run os & app.



* Types of Virtualization:-
- H/w or Server Virtualization (full, partial, para)
- N/w virtualizn: (internal & external n/w)
- Storage: Block, file virtn
- Memory: Appln level, os level
- Software: os level, Appln, Service
- Data: Database
- Desktop Virtn: virtual desktop infras., hosted virtual desktop.

In cloud computing, resources are pooled to serve multiple users using multi-tenancy. i.e it allows users to be served by same physical hardware

Virtulization in cloud :-



Virtualization layer
Physical layer → Processor, disks, memory, N/w

- Guest OS is an os that is installed in VM in addition to the host OS.

* Hypervisor / VMM - Virtual Mac. Monitor
2 types :- Type 1 hypervisor, Type 2 hypervisor

(1) Type 1 hypervisor or Native hypervisors run directly on host hardware & control h/w and monitor guest operating system.

(2) Type 2 hyper. or Hosted hyper :- run on top of conventional (main or Host) OS & monitor guest OS.

* Various forms of virtualization :-

(1) Full Virtualizn : Guest OS needs no modification & is not aware that it is being virtualized. It is enabled directly execution of user requests & binary translation of OS requests.

(2) Para Virtulzn :- guest os is modified to enable communication with hypervisor to improve performance & efficiency. Guest OS Kernel is modified to replace (non virtual) instructions with hypercalls that

can communicate with hypervisor.

(3) Hardware-Assisted Virtn: simplifies & improves virtn by using special h/w features built into processors like Intel VT-x & AMD-V
- these features allows hypervisor to directly interact with h/w, avoiding need for complex techniques like binary translation or para virtn
- Guest os can run instructions on physical processor more efficiently, even in virtualized env.

* Software defined Networking:-
- Traditional N/w:- Uses phy devices like routers & switches to control data flow. Changes require manual setup / new hardware. Hard to scale.

- Software defined N/w (SDN):- Seperates the control from the hardware. Managed by software from a central location, like remote brain controlling the N/w. makes n/w flexible and programmable, so changes are fast & easy using open APIs.

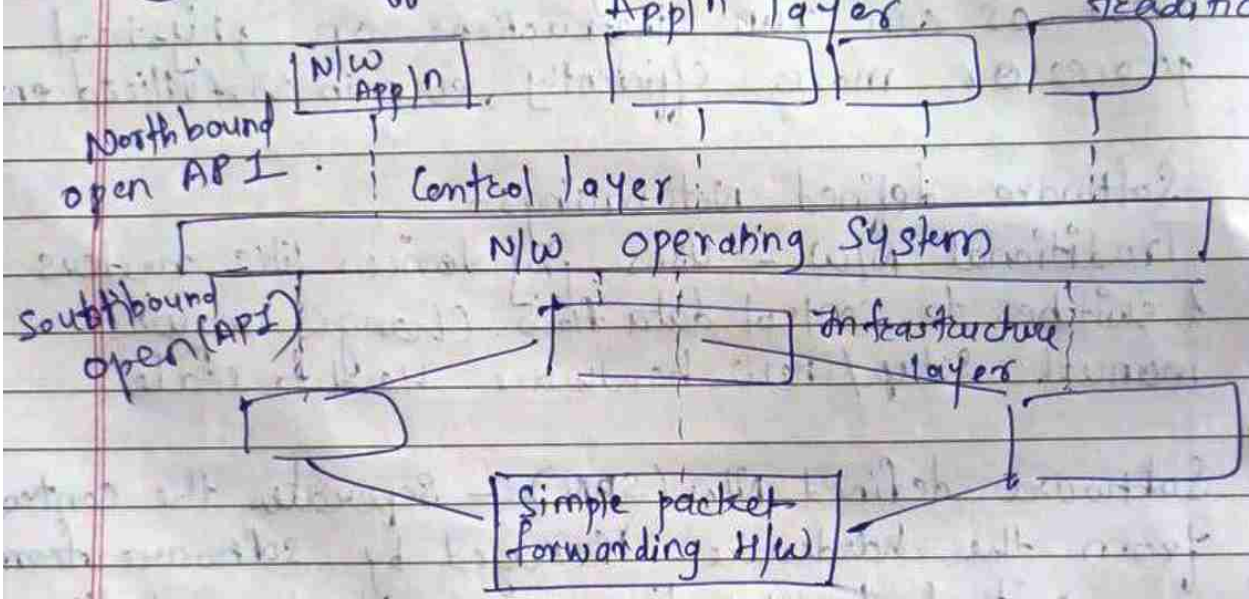
In traditional, Each device has two jobs:-
- Control Plane (Decides where data should go)
- Data Plane: forwards actual data packets to its destn.

Problems:-
(1) Complex devices: Packed with many protocol.
(2) Hard to Manage: Hard to update or change n/w. have to do manually
(3) Limited scalability: Adding new devices or handling massive appln is diff.

DATE / /

- SDN :- It decouples control from h/w.
  - Control Plane → Centralized in one place (Controller) making decisions for whole n/w.
  - Data Plane → devices like switches only forward packets based on instrⁿ from controller

Advantages :- ① Simpler hardware ② Centralized management ③ Scalable & flexible ④ Cost-Effective : uses inexpensive h/w compared to traditional n/w



SDN Architecture

* Map Reduce :-

Map Reduce is a method for processing and analyzing very large amounts of data, by breaking it into smaller tasks that can run at the same time (parallel processing)

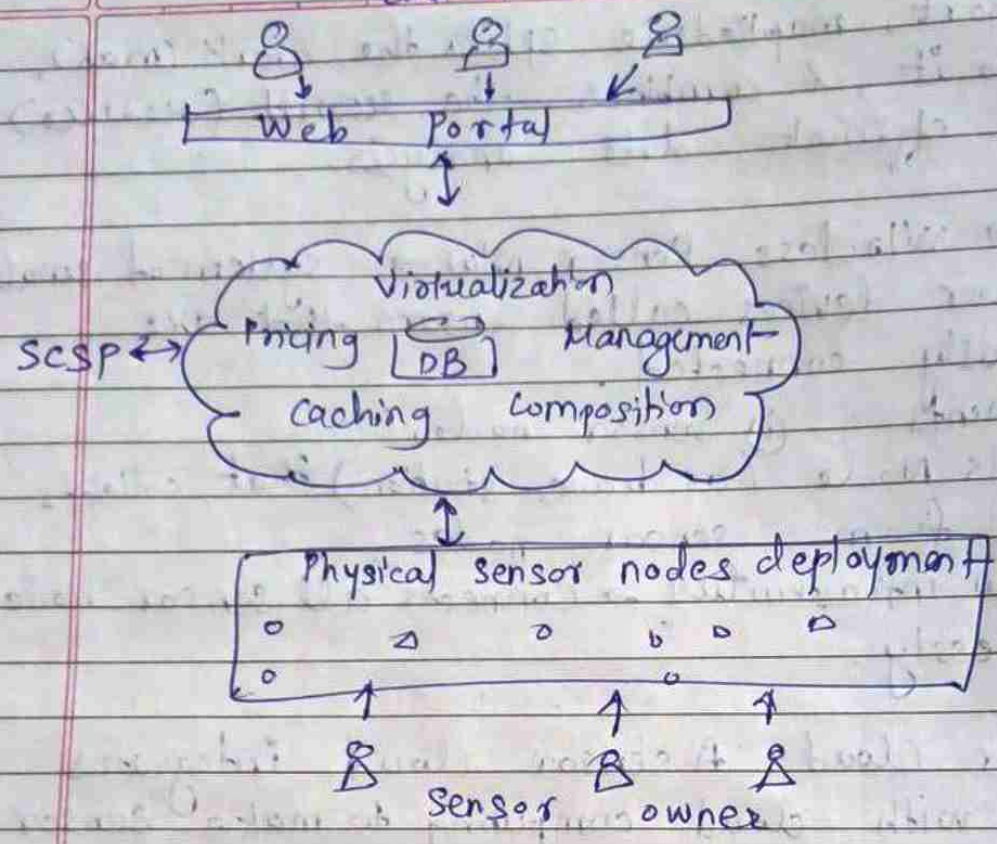
- Map functn → takes a set of data & converts it into another set of data, where individual elements are broken down into tuples (key/value) pairs
- Reduce functn → takes o/p from a map as an i/p and combines these data tuples into smaller set of tuples.

In short, mapReduce splits the work (map), processes it, & combines the results (reduce) for efficient data analysis.

* WSN → Wireless Sensor N/w → system of small, low power devices called sensors that are wirelessly connected.
Components :— ① Sensor nodes
② Sink Node (or base station) that collects data from sensor nodes
③ N/w infrastructure → Connects all sensor nodes wirelessly.

* Sensor Cloud : A sensor cloud integrates WSN with cloud computing to make sensor data accessible, manageable, & analyzable through cloud.
It works like → ① Data collection. Sensor nodes in WSN gather data & send it to cloud via sink node.
② Data Storage & Processing ③ Data Sharing → shared with users, or applications in real time via web or mobile interfaces.

Architecture :— • End Users → Registered themselves, selects templates and request for application.
• Sensor - owner : Purchase phy sensor devices, deploy over diff geog locations, & lend these devices to sensor cloud Service Provider (SCSP)
• SCSP → Plays managerial Role. Charges price from end users as per their usage.



- formation of virtual sensors →
  VS → logical representation of physical sensor in cloud.
  VSG → collection of VS.
  Optimal formation → Ensures sensors selected for VS or VSG are efficient, reliable & suitable for the task.

- Challenges in Sensor - Cloud →
  ① Efficient Virtualization of Sensors
  ② optimal composition of VSG
  ③ Geographical constraints → Ensuring sensors from same or overlapping areas are grouped logically.

* FOG Computing :-

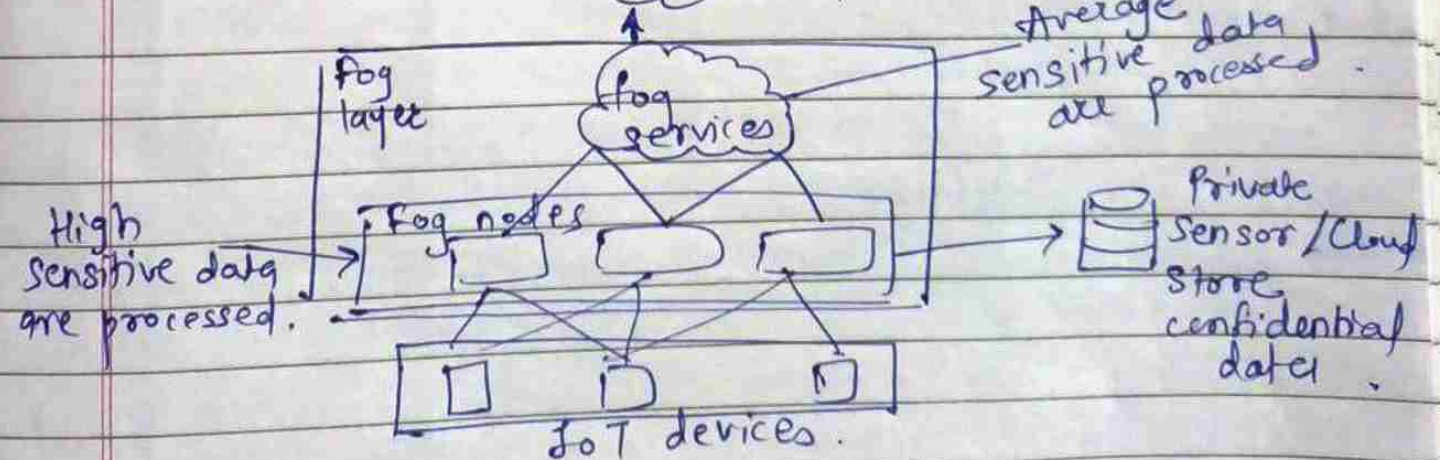
- fog computing is like a middle layer between your devices and cloud. Instead of sending all data to cloud for processing, fog computing processes data locally near the source.

Why its needed?

→ Issues are - volume - latency - Bandwidth
Sending data to cloud from every device causes delay (latency) ↑ requires lot of bandwidth.

→ With fog computing, data is processed near device, making it faster & reducing load on cloud

- Architecture :-
  - Cloud services are extended to IoT devices thr fog
  - In fog layer, many fog nodes are present
  - Sensor data are processed in fog before it is sent to cloud



In this way it maintains data security too.
- most time-sensitive data → analysed in fog nodes.
- less time-sensitive data → aggregated nodes analyze this
- non-time sensitive data → sent to cloud.

PAGE No.
DATE / /

- Advantages
  - Latency is reduced
  - Quick decision making
  - Better privacy. (Store confidential data in local servers)
  - Deployable in remote places
  - Better data handling
    - Can operate with less b/w
    - Data can be analyzed locally.
    - Reduce risk of latency.
  - low operation cost. (Reduces bandwidth consumption)

- Applications
  - Real time health analysis.
    - Patients with chronic illness.
    - Stroke patients. Analyze realtime data.
    - During emergency, alerts respective doctors immediately.
    - Historical data analysis can predict future dangers of patient.

Comparision betn diff IoT Cloud Platforms (8 mrks)

Parameter	AWS IoT	Cisco IoT	Google Cloud IoT	IBM Watson IoT	Microsoft Azure IoT
Primary use cases	Smart cities, Connected homes, healthcare.	Connected vehicles, manufacturing utilities.	Smart cities, logistics, energy	Agriculture, manufac, logistics	Health care, retail, manufacturing
Core features	Device shadow, real time analytics edge computing	Fog comp, edge processing 5G readiness	Predictive, maintainance, real time tracking	Predictive, maintainance, block chain support	Digital twins, anamoly detection, IoT Hub
Edge computing	FreeRTOS + Greengrass for processing at Edge	Fog comp with edge intelligence	TPU chip for AI/ML edge	Edge appln manager	IoT edge for dev + real time processing
Data Management	Secure, scalable device + data management	Reliable data control + managem.	Cloud BigQuery, data flow for analytics	Advanced lifecycle management	Time Series insights, Azure databricks
Integration support	Seamless with AWS services (Lambda, S3)	Best with Cisco devices.	Best with Google Services (BigQuery)	Blockchain + AI for industrial IoT	Works with enterprise tools (SAP, Salesforce)
Scalability	Extremely scalable for large IoT deployments.	Optimized for Edge + fog env.	Can handle millions of devices.	Scalable industrial IoT solutions	Supports billions of devices.

	AWS	Cisco	Google	IBM	Microsoft
Security features	Authentication, Defender for device protection	eSIM service for secure connectivity	Built-in device security	High Standards with MQTT protocol	Multi-layered Security with Azure Sphere
Pricing	High cost pay as you go: free trial available	Pricing on Request	Affordable free tier available	Pricing on request	Cost-effective with free trial option.