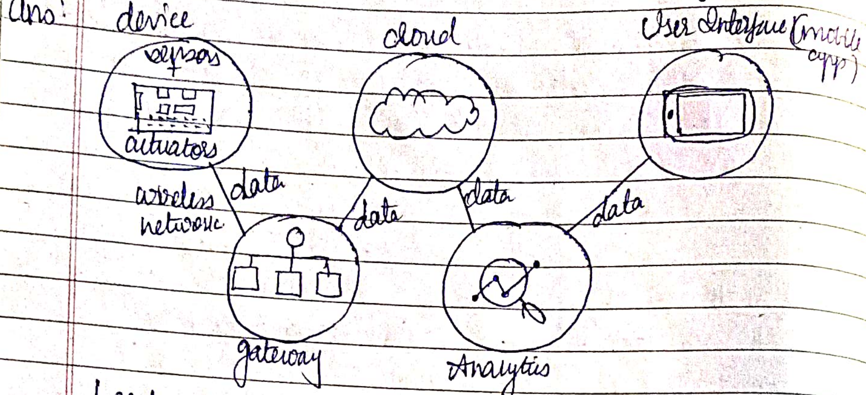
# IOT

## Unit 3.

1. Demonstrate IoT components with diagram.

Ans:




hardware used in IoT includes devices for:
1. remote dashboard,
2. devices for control
3. servers
4. routing/bridging devices.
5. sensors

They manage key tasks activities like:
activation, tech specifica^n, security, communica^n, detec^n to support specific goals & ac^ts.

Components:
1. Control units: small comp on single IC containing processor, memory & programmable I/O peripheral responsible for main operation
2. sensors: measure physical quantity & convert it into a signal

| | |
|---|---|
| temp sensors | accelerometers |
| image " | gyroscopes |
| light " | acoustic sensors |
| micro flow " | humidity " |
| gas RFID " | pressure " |

3. Communica" modules: responsible for communica" with rest of the IoT platform. provide wired / wireless connectivity. 2 ways of communica" betwn IoT sys & internet:

1. Internet enabled intermediate node acting as a gateway
2. IoT device has direct communica" with internet

- communica" betwn main control unit & communica" module uses serial protocol

4. Power sources:

small devices = current produced by batteries, solar cells / thermocouples

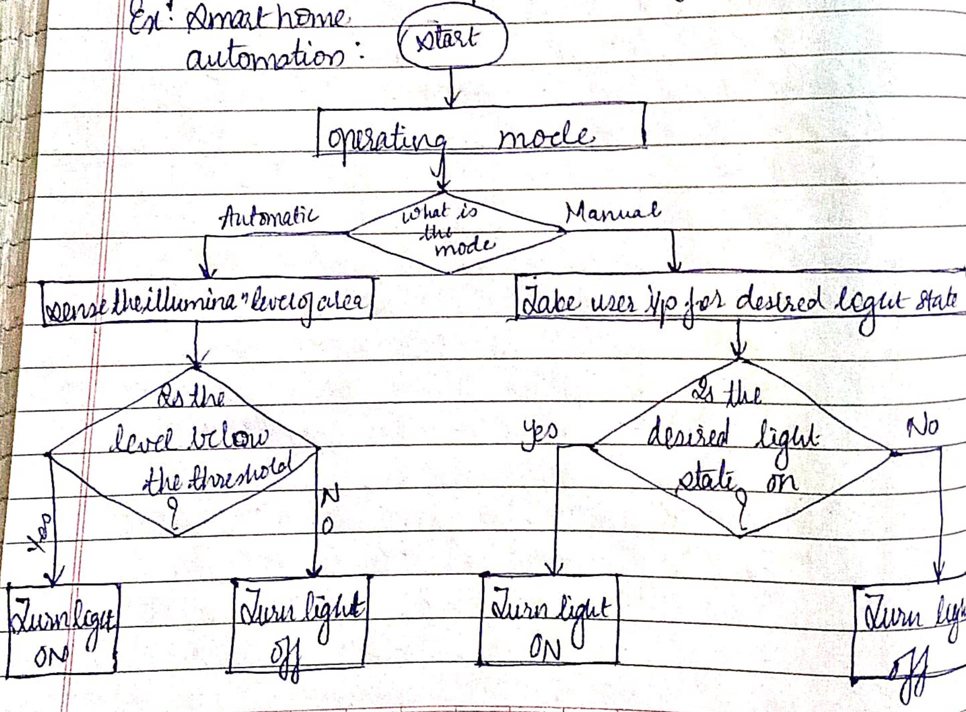
mobile devices = powered by lightweight batteries that can be recharged for longer life.

2. Explain steps to IoT sys. design methodology

Ans: Step 1: Purpose & requirement specification:
- define purpose & requirements of IoT sys.
- precise requirements:
    1. sensing "
    2. actuation "
    3. data collecⁿ "
    4. data analysis "
    5. sys management "
    6. data privacy & security "
    7. User interface "
    8. applicaⁿ "
    9. API reqs "

Step 2: Process model specificaⁿ: define users case

Ex: Smart home automation:



process models clearly identify what happens when & which entities are responsible for carrying out the process.

step3: Domain model specifica": define physical & virtual entities, devices, users, resources & services in the IoT sys.

- defines attributes of obj's & the relationships betw those.
- abstract representation of concepts, objs & entities in IoT domain.

key terms:

1. abstrac": provide notions/ concepts behind smtg w/o getting into precise detail.
2. physical entity: identifiable & discrete part of physical & the actual real environment.
3. virtual entity: virtual counterpart of physical entity of interest
4. Device: provides tech interface for interacting w/ or gaining info abt physical entity

   3 types:
   - Sensors: provide info/data about physical entity
   - Tags: identify physical entites to which the tags are usually physically attached.
   - Actuators: modify the physical state of physical entity.
5. Resources: s/w components that provide sensed & collected data from physical entities/ are used in actuation on physical entities

   2 types:
   - Ondevice: hosted, running & avaliable on device itself could further include executable code for assessing, processing & storing sensor info & code for controlling actuators as req.

- network resources: available over network & they req. network based communica^n protocol for accessing them.

6. Services: provides an open and standardised interface and offers all the necessary functionalities for interacting with the resources & the devices associated with physical entities

types:

- Resource level services: expose the functionality usually of a device by accessing its hosted resources.
- Virtual Entity-level services: provide access to info at virtual entity level.
  can be services associated to a single virtual entity that provide access for reading attribute info / for updating attributes.
- Integrated services: result of a service composition of resource level or virtual entity level services as well as any combos of both service abstraction.

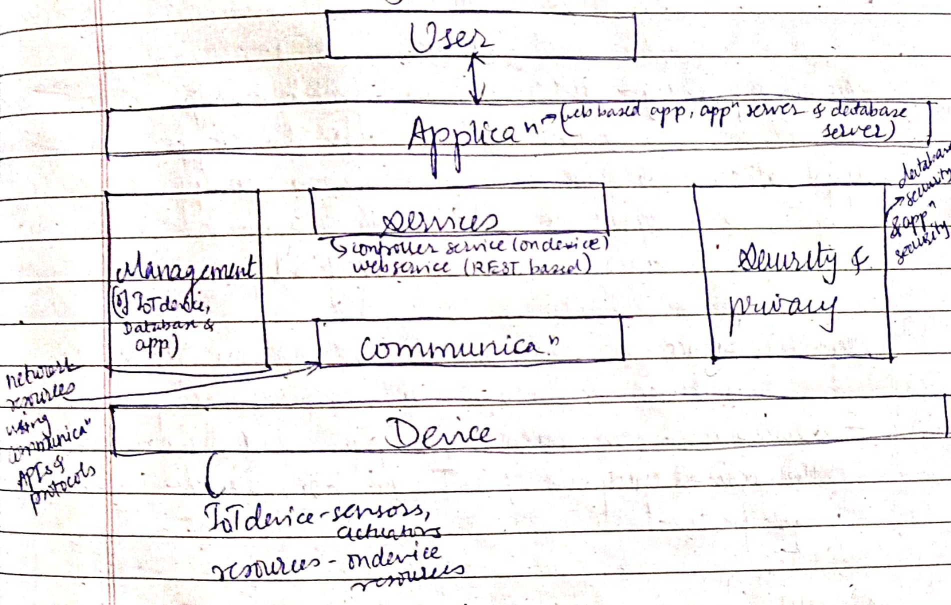
7. Users: human person / some kind of digital artefact, such as a service, an app^n / a s/w agent, that needs to interact w/ a physical entity.

step 4: Info model specification: define the struc (such as relations & attributes) of all info in the IoT sys.
- specifies how info is stored & represented

step 5: service specifica^n: map processes & info model to service & define service operations
- define services in IoT sys, service types, schedules, preconditions & effects.
- for deriving services, u look at process model specifica^n & info model specifican

Step 6: IoT level specifican: define IoT level for a sys.
- System is defined to have one of 6 levels of IoT.

Step 7: Funcnal view specifican: map IoT level to funcnal grps
- describe each of these funcnal blocks/grps in terms of what do they provide w.r.t. overall IoT sys being designed.




Step 8: Operational view specifican: define communican options, service hosting options, storage options & device options

Operational grp - operational view specifican
device - computing device = Raspberry Pi
sensor = LRR
actuator = Relay switch

mangement - device management = Raspberry Pi
database " = MySQL db
applica" " = Django
communica" - communica" API = REST APIS protocol
link layer = 802.11
Network layer = IPv4
Transport = TCP
applica" layer = HTTP
service - native service - controller service
web service (REST based)
mode "
state "
security & privacy - Authentication - for db & app"
Authorisation - for db & app
application - web based app" - Django web app
app" server - Django app server
database server - MySQL

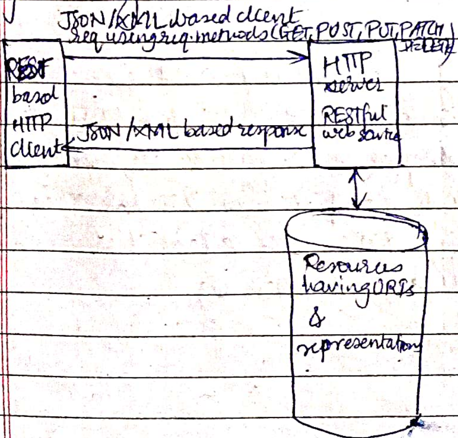
step 9: Device & component integration: Integrate devices, develop & integrate the components
- devices are connected & programmed to achieve the desired purpose of IoT sys from the h/w device perspective

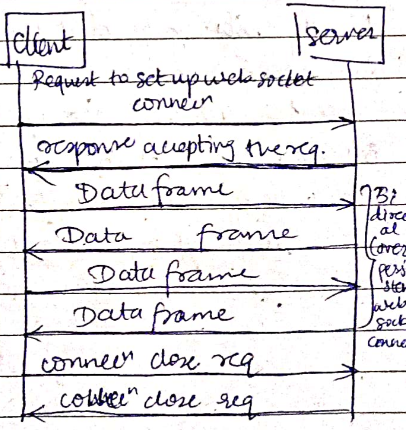
step 10: App" development: Develop an IoT app
- app provides the interfaces through which either ~~ext~~ a human / non-human user could interact with IoT sys.

Q3. Illustrate REST API and WebSocket API
Compare REST & Web Socket API

ans

| comparison attribute | REST based API | WebSocket based API |
|---|---|---|
| Communication model | Request - Response | Exclusive pair |
| 2. Data flow | server responds to the client on each request (uni directional) | Bidirectional |
| 3. State info | not preserved (stateless) | Preserved (stateful) |
| 4. Session maintained by | client | server & client |
| 5. Resource requirements | comparatively lower | comparatively higher |
| 6. Connection overhead | required | not required |
| 7. Uses | get or process data | realtime apps such as games |
| 8. scalability | High | low. |



Few extra points:

REST APIs (architectural constraints):

1. Client-server: requires both don't need to worry about each other's resources. This separation allows both to be independently developed, managed & updated.
2. Stateless: each req. from client to server must contain all of the info necessary to understand the req. & not take adv of stored contents (prev. req.s)
3. Cache: ensures that a client could reuse server response data later to improve network efficiency & performance
4. Uniform Interface: ensures that method of communicaⁿ between REST components must be uniform such that implementations are decoupled from the services they provide.
5. Layered sys: allows architecture to be composed of hierarchical layers by constraining component behaviour.
6. Code-on-Demand: allows the client funcⁿality to be extended by downloading & executing code in form of scripts from the server.

Web Socket APIs:

1. Supports full duplex communicaⁿ
2. reduce network traffic & latency as there is no overhead for conneⁿ setup & terminaⁿ req. for each message
3. Uses std. HTTP req.-response sequence to establish conneⁿ. When conneⁿ established - provides read & write interface for reading & writing data over established conneⁿ in an async full duplex manner.

compare betw diff IoT communica models with their advs & disadvs.

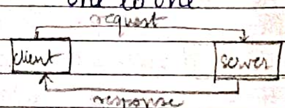
OR

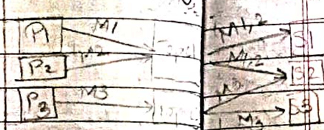
Q4. Demonstrate diff communica models.

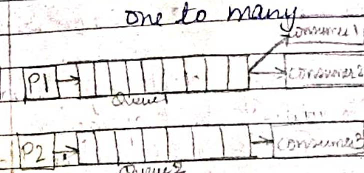
Ans:

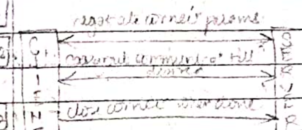
M = message

| Comparison attribute | Request - response | Publish (P) - Subscribe (S) | Push pull | Exclusive pair |
|---|---|---|---|---|
| 1. State | stateless | stateless | stateless | stateful |
| 2. Client req./action | required | not required | required | optional |
| 3. Type of communica | synchronus | asynchronus | asynchronus | synchronus |
| 4. resource req. | low | low | low | high |
| 5. Connecn overhead | High | Low | Low | low |
| 6. Scalability | medium | High | High | low |
| 7. direct connectivity between server & client | required | not required | not required | required |
| 8. message access & distribution | one to one | one to many | one to many | one to one |



publisher: entity that generates the messages & pushes em to topic

Topic: logical entity that holds the messages sent by the publisher

Subscribers: consumers that are interested to get messages on a particular topic w/o directly communicating w publisher

Broker: optional 3rd party betwn publisher & subscriber.

Q.5. What are 4 pillars of IoT, explain their working

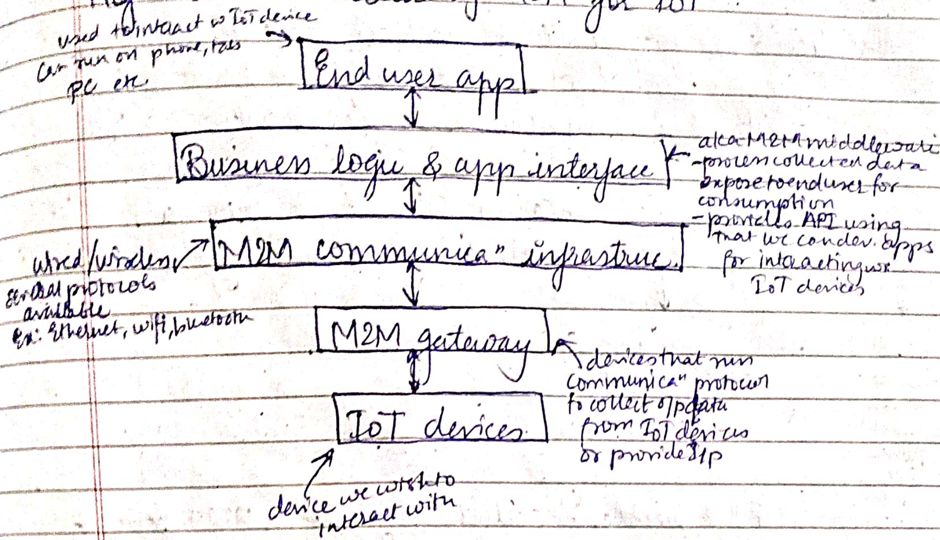
ans: Pillars of IoT are
M2M, RFID, SCADA, WSN

1. Machine-to-Machine (M2M):
techs, stds & protocols that enable the machines to communicate & interact w e-o & carry out useful tasks.

Applications:
1. Robotics: used to place inventory in warehouses & autofill shipments
2. logistics & fleet management: track movement of vehicles
3. Utility: smart utility meters for electricity, water gas etc
4. Vending machines: providing current operational status & stock of items & send someone to restock as necessary.

General High level architecture of M2M (defined by ETSI
1. M2M device: device capable of replying to requested data/ transmitting data on its own.
2. M2M area network (Device domain): provides connectivity in M2M devices & M2M gateways
3. M2M gateway: interconnects M2M device to the communic
4. M2M Communica" network (Network domain): communica" betw M2M gateway & M2M app. Ex: LTE, WLAN.
5. M2M apps: middleware layer where data goes through various app services & used by business processing prog

# High level architecture of M2M for IoT :




| attribute | M2M | IoT |
|---|---|---|
| tech | M2M | connected things (sensors & actuators) |
| apps used | vertical apps | horizontal apps |
| IP protocol | not used | used. |
| logic embedded in | H/w | H/w & S/w |
| interoperability | low | high |
| scalability | low | high |
| internet connectivity | rare | often. |
| network | | |

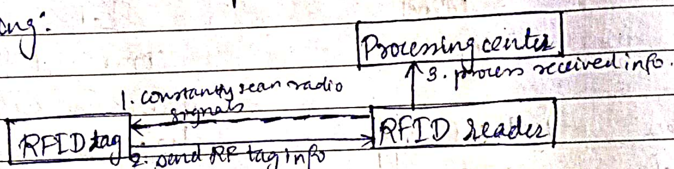
2. Radio Frequency Identification (RFID):
tech using which an obj can be identified, tracked & monitored using radio waves.

RFID tag: contains circuit & antenna. contains info to identify & track an obj

- active
  - have their own power src
- passive
  - activated using external power src.

RFID reader: collects info from RFID tags & process it as desired.

working:




1. RFID reader continuously scans RFID tag for radio signals
2. as soon as RF enabled tag comes near reader, it activates the RF circuitry in RF tag & extracts info from it
3. Extracted info is sent to processing center & carry out desired actions

Applican:
1. Read tracking data in buildings
2. Transport industry for no. plate monitoring for traffic management
3. Travel industry for tracking baggage, ~~paspor~~ passports & other safety equipments.
4. Hotel industry for keys, tracking baggage, tracking staff.

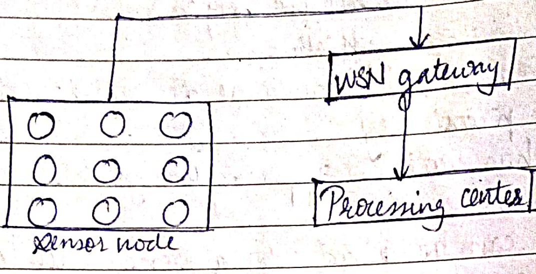
Date
Page

advs:
1. tags can store good amt of info
2. " are re-writable
3. RFID tech is robust & proven.
4. " is cost effective
5. " can work from some dist. & does not req. extreme proximity.
6. " has various apps & usage & is interoperable

disadvs:
1. tags can be read by anyone with a RFID reader.
2. might be labour intensive to prog. RFID tags & attach to each obj.
3. Any electro magnetic interference can interrupt the funcn ing of RFID.

gement

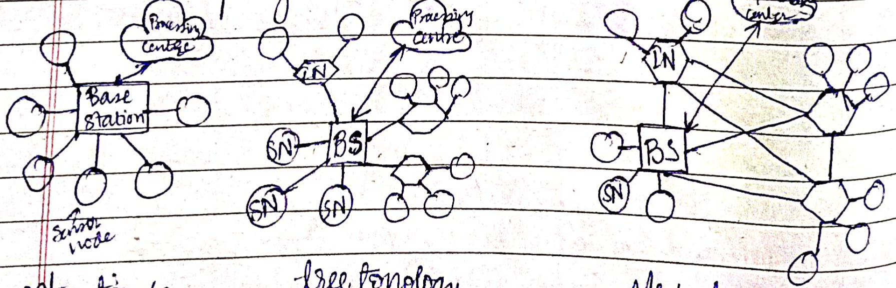
3. Wireless Sensor Network (WSN): large collec$^n$ of sensor devices that can monitor several physical condi$^n$.




1. sensor device = sensor node = monitor several physi[cal] condi$^n$s such as temp, air pressure, illumina$^n$ of light, movement of people, wind speed, humidity, e[tc]
2. collected info is sent to processing center via WSN gate[way] (aka base station / sink node)
3. processing centre evaluates info received from vario[us] sensor nodes & sends instruc$^n$s to the connected devices to act suitably
4. WSNs use IEEE 802.15.4 standards & ex is Zigbee

Applica$^n$s:
5. commonly used for area monitoring, weather predic$^n$, security & industrial operations

WSN topologies: B.S = Base station, I.N. = intermediate node, S.N. = sensor node



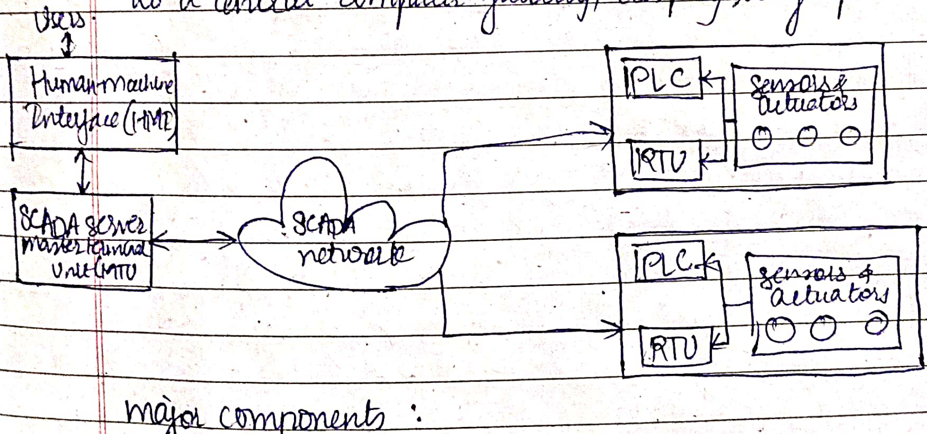

Star topology | Tree topology | Mesh topology

4. Supervisory Control And Data Acquisition (SCADA) is a control system architecture comprising comps., networked data communica^n & GUI for high-level process supervision, control & data acquisition in manufacturing plants & other industrial settings.

Applica^n:
used in distribution sys. such as water distribu^n & waste water collec^n sys, oil & nature gas pipelines, electrical utility transmission.

Working: collect info from various IoT sensors & provide control via actuators.

Architecture: collect plant & facility info, transfer it to a central computer facility, display it graphically




major components:

1. Facility & equipment sensors & actuators: no. of em across plant site, monitor & control temp, pressure, voltage current etc.

2. Programmable Logic Controllers (PLC) - connected to sensors & actuators - collect data from sensors & provide data to actuators. (implemented where local control needed & multiple i/o)

3. Remote Terminal Units (RTUs): connected to sensors & actuators collect data from sensors & may optionally provide data to actuators. (used for remote plant sites, could have several RTUs at plant sites)

Date
Page

4. SCADA network : utilise radio, telephone lines, fibre, cable satellites or other communica" mechanisms as appropriate. allows transfer of info & data back & forth betwn the SCADA server & ~~RTUs~~ RTUs or PLCs
5. SCADA server (or Master Terminal Unit) : controls the overall plant operations. aka control server collects data from RTUs/PLCs. Server S/W is programmed to tell sys what n when to monitor, what parameter ranges are acceptable & what response to initiate when parameters change outside acceptable values.
6. Human Machine Interface (HMI) : single monitor/worker or could be several depending on plant size & monitor & controlling requirements. Users can visually monitor overall operation in real time & type commands to send to the RTUs / PLCs. via server.

Date ________
Page ________

Q.6. Differentiate betw Vertical & Horizontal IoT apps.

ans:

| Aspect | Vertial IoT apps | Horizontal IoT apps. |
|---|---|---|
| 1. Defi^n | focused on a specific industry or domain | Design to work across multiple industries / domains |
| 2. Scope | Narrow, tailored to meet the needs of a specific usecase | Broad, addressing common generalities usable across selectors. |
| 3. Ex: | smart healthcare, industrial, automation, smart agriculture | cloud platforms, analytics tools, device management solu^ns. |
| 4. Customizations | Highly customized for industry-specific reqs. | Generalized to be adaptable across various use cases. |
| 5. Technology stack | often proprietary / industry specific techs. | Open & reusable frameworks or platforms |
| 6. Interoperability | Limited, works within the specific domain's ecosys. | moderate, as it focuses on reusable, scalable components. |
| 7. Ex.s of companies | Philips, John Deere - Healthcare IoT, AgriIoT | AWS IoT, Azure IoT Hub, Google cloud IoT. |

Q7. Categorize / Classify diff. connectivity technologies req^d for IoT app. dev^n & explain one of em in brief.

ans:

| Category | Technology | Explanations |
| --- | --- | --- |
| 1. Short range Low power | Bluetooth | - low power<br>- used for wearables, audio devices, health monitoring |
| | Wifi | - High speed.<br>- used for smart homes, offices, video streaming |
| | Zigbee | - mesh network<br>- smart lights, thermostats & security |
| | NFC (near field communica^n) | - short dist<br>- contactless payments, authentica^n |
| | Z wave | - mesh network using low energy radio waves<br>- communicate appliance to appliance |
| Low power, Wide area | 4G LTE | - High capacity, low latency<br>- great for real time updates |
| | 5G | - faster download speeds.<br>- connectivity to much more d... |
| | Cat-0 | - LTE based network<br>- lowest cost |
| | Cat-1. | - std for cellular IoT<br>- easy to set up<br>- great solu^n for apps requiring voice / browsing interface |
| | LoRaWAN | - Low Range Wide Area Network<br>- connect mobile<br>- secure<br>- bidirectional battery-operated devices. |