IoT Design Methodology.

1. purpose and requirement

1. This is the 1st step in IoT system design methodology.
2. In this step we decide ~~why~~ the purpose of building the IoT system.
3. And also what problem this IoT system will solve and ~~to~~ what is its requirement in real world.

2. process specification.

1. Here we will define the steps or flow of our IoT system, as how it will work in sequence.
2. As sensing things from environment
   → to taking actions.

Example;

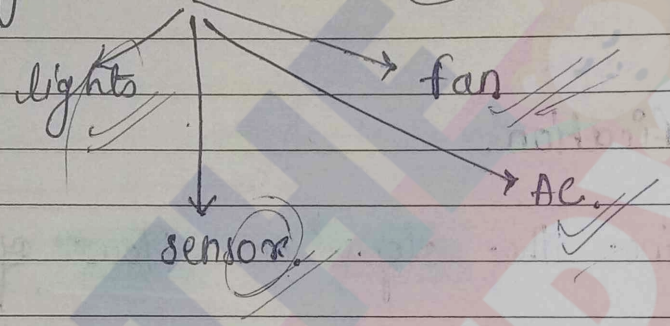
1. Sensor collects room temperature
2. Data is sent to the cloud.
3. cloud checks if temperature > 28°C
4. If yes, AC is turned ON automatically.

3. Domain Model specification.

1. The domain means the main areas where IoT system is used.

2. we Identify the main entities in that area and how they are related.

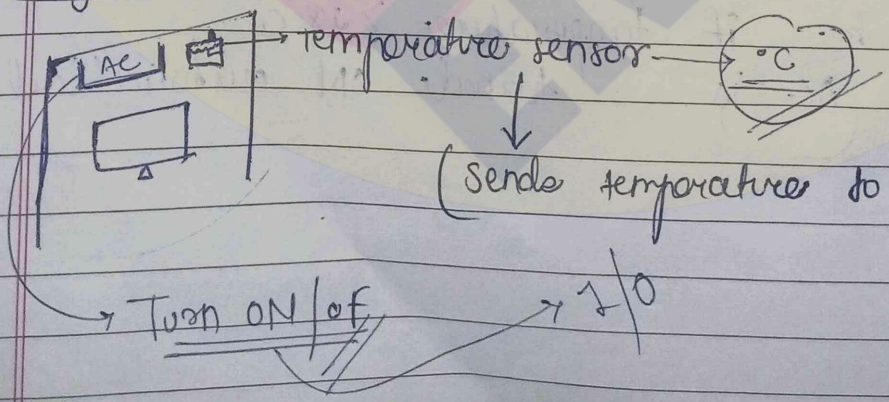
3. eg:- Entities (Home)



4. Information Model Specification.

1. In this step we decide what kind of data each device will collect, send or store.

eg:-



5. Service specification :- IoT system

1. It describes what services ~~or~~ ~~fe~~ the IoT system will provide to users.

eg: ① AC will get turn ON automatically.

② Turn ON/off lights remotely.

6. ~~Functional view specification~~ :-

6. IoT level specification.

1. IoT system can be designed in different levels (1 to 6) depending on size and complexity.

7. Functional view specification

This system shows how the system will actully work → the logic behind every service.

eg: service ⟹ "TURN ON AC"

→ When temperature > 28°C → Then AC ON".

8. ~~Device~~
8. Operational view specification :-

1. Here we explain how the system will run in real life.

2. Network setup,

IoT system ⟶ connect ⟶ wifi / Bluetooth

IoT system ⟶ power use.

3. This steps focuses on the practical working conditions.

eg :- Devices
↓
uses power supply / battery.

9. Device and component Integration.

Here we connect all hardware and software parts together.
(wires)

→ sensors, actuators and app.

All parts are made to work together smoothly.

10. Application Development.

1. Finally we build the user application
(website / App)

that allow users to monitor and control devices.

eg:- A mobile app that shows room temperature alerts and buttons to turn AC / lights ON or off.

SMART IRRIGATION SYSTEM.

Date
Page

1. purpose and Requirements

goal : Automatically water the crops based on soil moisture and weather conditions

To save water and ↑ yield.

Requirements : sensor
mobile app.
water pump.

2. process specification Model.

This step ~~system~~ will show how system will work.

process flow :

1. sensor → soil moisture → reads data
2. If soil is dry → send singnal → controller
3. controller → Activate water pumps.
4. When moisture reaches set level → pump stops.
5. Data displayed on farmer's App.

3. Domain Model specification.
Agriculture

Defines the main entities in the system and how they are related.

entities :

- Soil Moisture sensor
- controller
- water pump
- Mobile App
- cloud platform

4. Information Model specification

Describe what data is collected and in what format its shared.

- Soil Moisture (%) Sensor
- Temperature (°C) Sensor
- water pump status (ON/OFF).

5. Service specification :

Defines the services the IoT system offers.

eg :- Read Soil Moisture() → returns current soil moisture.

Turn pump on / Turn pump off → controls water flow.

6. Iot Level specification :

Defines which IoT level architecture fits best.

Smart irrigation is usually Level (3) Iot system :

- Smart devices
- cloud storage and analytics
- user app for monitoring and control.

7. functional View Specification

Shows functions performed by each component

sensor → Measures soil moisture
pump → Delivers water.

cloud → Stores and analyze data.
App → Displays data to user.

8. operational view specification

Describes how the system works in real time,

eg :- Mobile app recieves updates instantly
cloud processes data and send alerts.
power

9. Device and components Integration

Hardware :

- soil Moisture sensor
- Temperature sensor
- water pomp.

Here we connect all hardware and software parts.

Software :

- cloud platform
- Mobile App.

All companies are tested together to ensure smooth working.

10. Application Developement

The final step is to create a user-friendly app or dashboard.

- show soil moishure, temperature, homidity
- pump control
- Data charts of irrigation history
- Alert notifications.

FOREST FIRE DETECTION SYSTEM

1. purpose and Requirement

The main goal is to detect forest fire early and alert authorities to prevent large scale damage.

Requirement :

- Detect temperature, smoke and humidity
- Display data on a dashboard or mobile app.

2. process specification model :

This explains how the system will work step by step

process

1. sensor → temperature, smoke and humidity
2. When fire signs are detected → microcontroller process data.
3. Alert message sent to cloud.
4. Cloud triggers sms.

3. Domain Model specification :
— forest fire Detection

Define the main entities or objects and their relationships.

entities :
- Temperature sensor
- Smoke sensor.
- Humidity sensor.
- Microcontroller
- cloud
- App.

4. Information Model specification :

Defines what kind of data is collected and how it is represented.

Data examples :

- Temperature (°C), Humidity (%)
- smoke level (ppm)
-

format example :

{ "temp" : 55, "smoke" : 320, "status" : "fire Detected" }

5. Service specification :

It describes services provided by IoT system.

- collects all sensor readings
- Notifies forest officer
- show live data

6. IoT Level Specification : 1 — 6

Defines which IoT level architecture fits this project.

It uses IoT level 4 system, because it involves:

- Multiple sensors in different forest zones.
- Cloud storage and analytics
- central monitoring

7. Functional View specification :

Explain what each part of sytem does.

sensor → Measures soil moisture, humidity, smoke
controller → Take actions or give instructions.
cloud → Stores and analyze data
App → Displays data to users

8. operational view specification.

Describes how the system runs in real life.

- like how
  devices connects which each other
  how communication takes place
  via protols (http / MQTT)
  powere saving mode, powere supply, battery

9. Device and components Integration.

Connects all hardware and software components.

Hardware :

Temp sensor
Smoke sensor
Microcontroller

Software :

App
Iot platforms.

Integration ensures all devices communicate and perform together, smoothly.

10. Application Developement :-

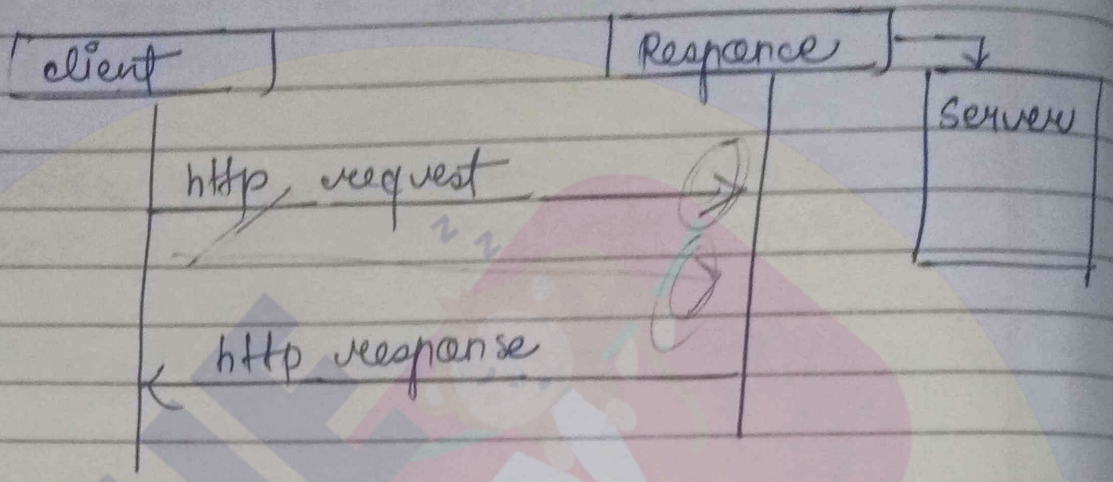
The final step is to create a user-friendly app or dashboards.

- Real time sensoing data
- Instant alerts.
- Fire location tracking using GPS.

IOT COMMUNICATION MODELS

To stateless

1. Request Response communication model

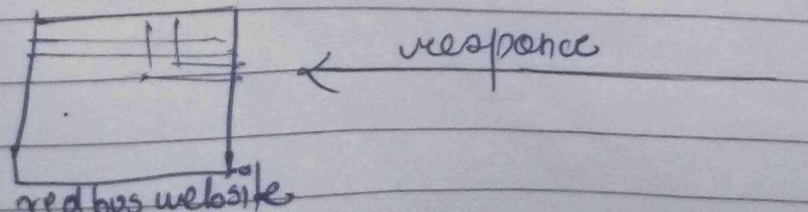


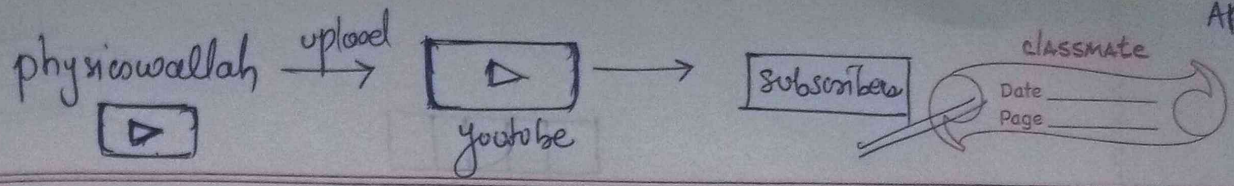
1. client request information from the server and wait till the response is served.

2. The client and server can communicate
   - one to one
   - one to many

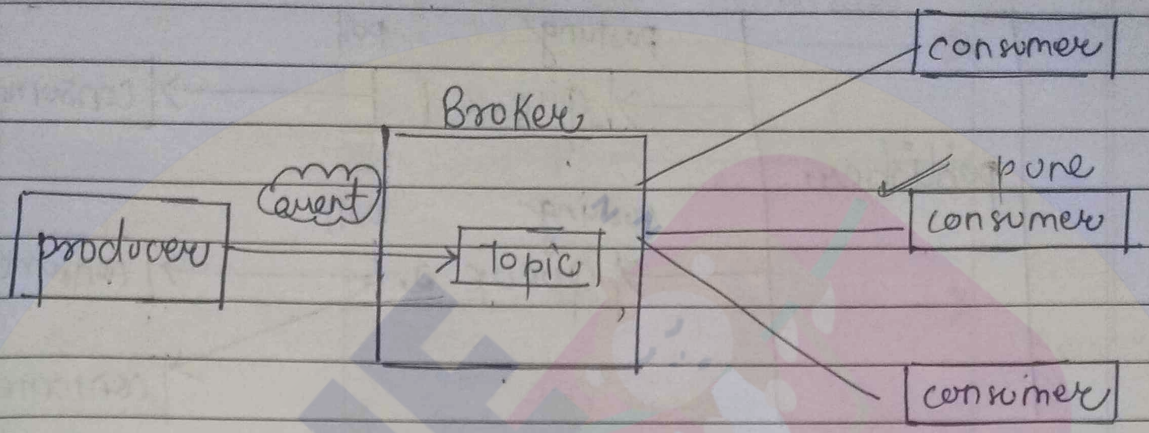
3. eg:-

client → webpage www.redbus —request→ server





2. publish / subscribe Model :



1. publishers : event generate and publish them.

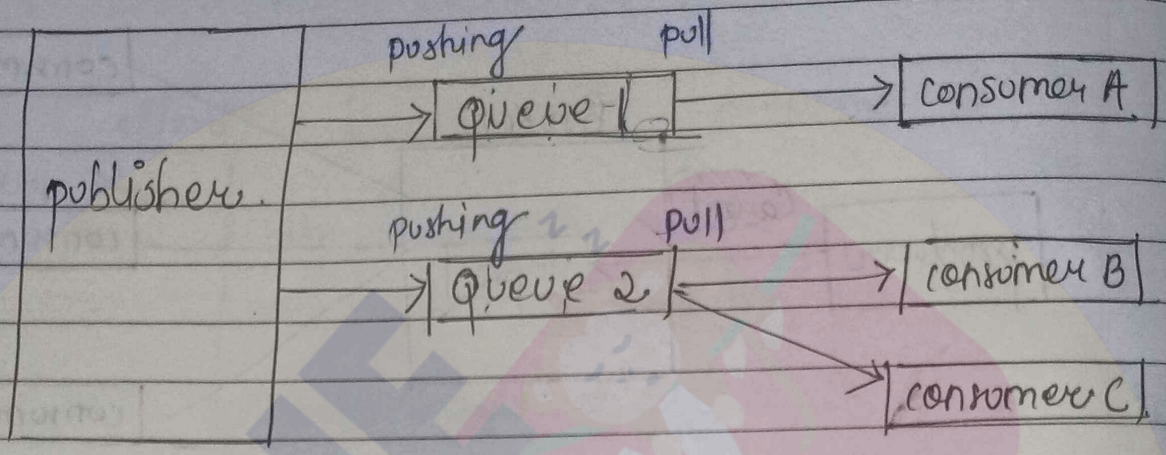
2. subscribers : process the events recieved.

3. Brokers : - It's a mediator
- It filters and routes event from publishers to intrested subscribers.

4. p / s model ⟹ is autonomous.

means that they do not need to know the presense of each other.

3. Push / Pull Model :



1. The publishers, publishes the data and push into the queues.

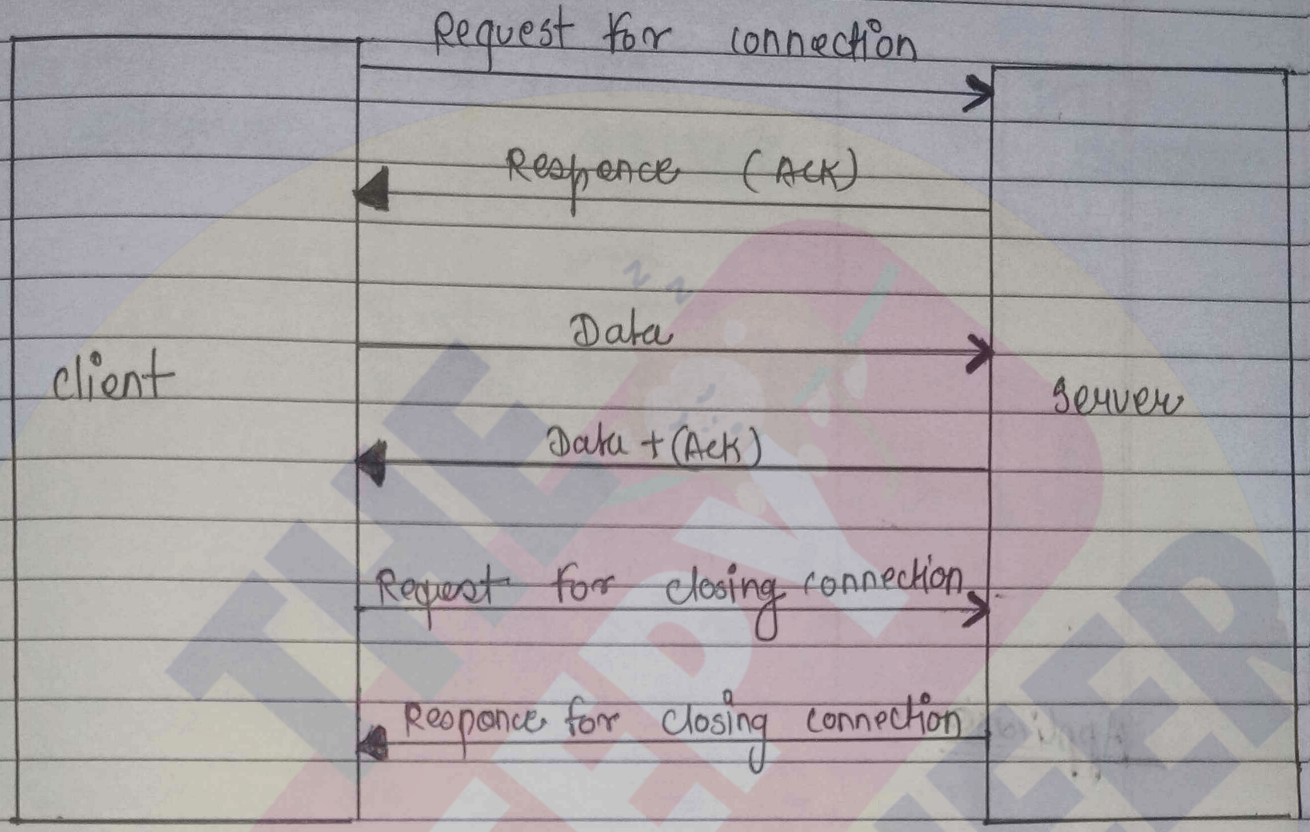
and

consumer pulls the data from queue.

2. To avoid traffic from publishers the queue sends the data in proper manner

3. It is autonomous.

4. Exclusive pair Model :-



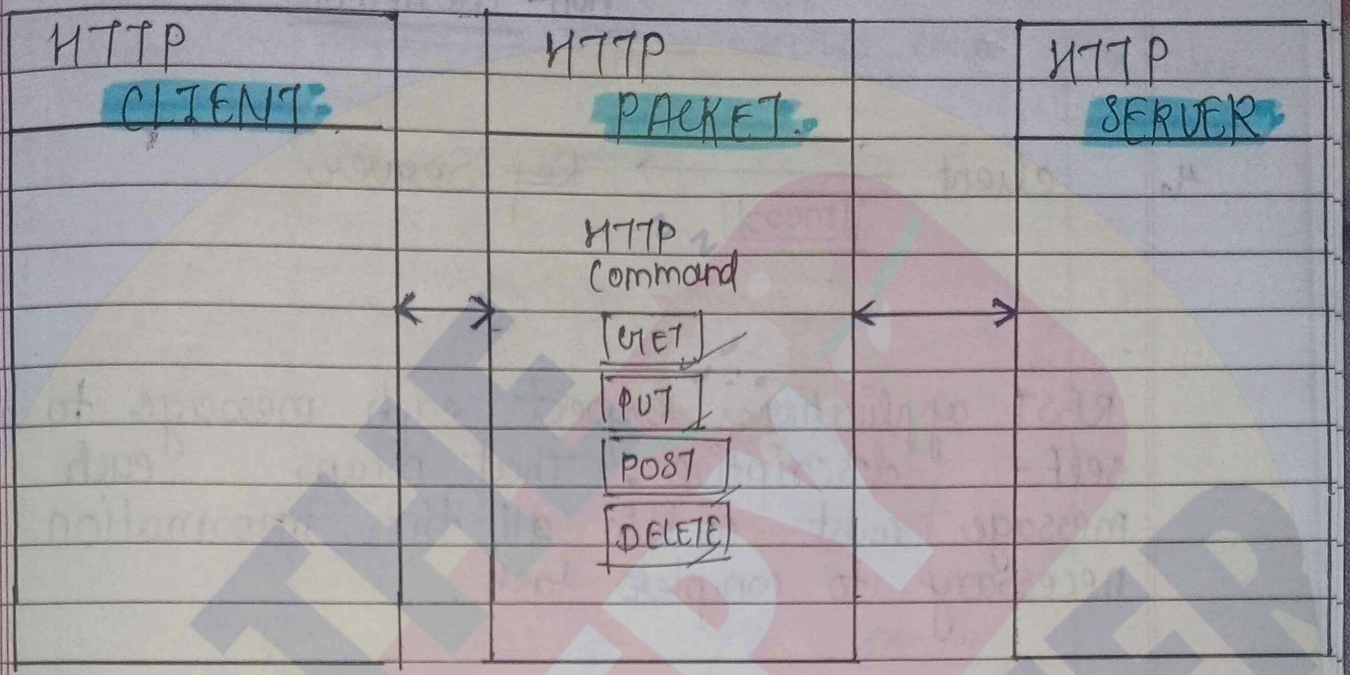
1. Exclusive line
   → full duplex, bi-directional.

request and responce can be done at same time.

2. client send request to server for opening the connection. This connection is open till the client send request for closing the connection.

IoT COMMUNICATION API's

1. REST BASED



1. Representational state Transfer (REST).
2. REST API's follow the request - response communication model.
3. These systems communicate over HTTP with the methods GET, PUT, POST, DELETE.

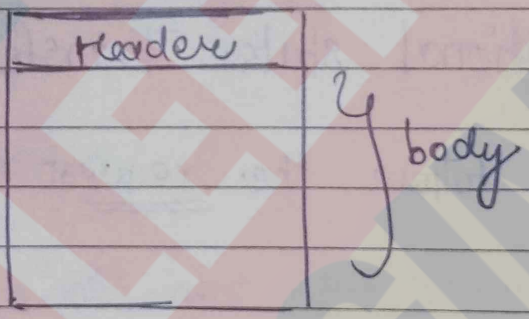
Key features

1. client - server
2. stateless :- A system where each request is treated independently and the server does not store any information about past interactions.

3. Cache : requires responses to be clearly labeled as cacheable or non-cacheable

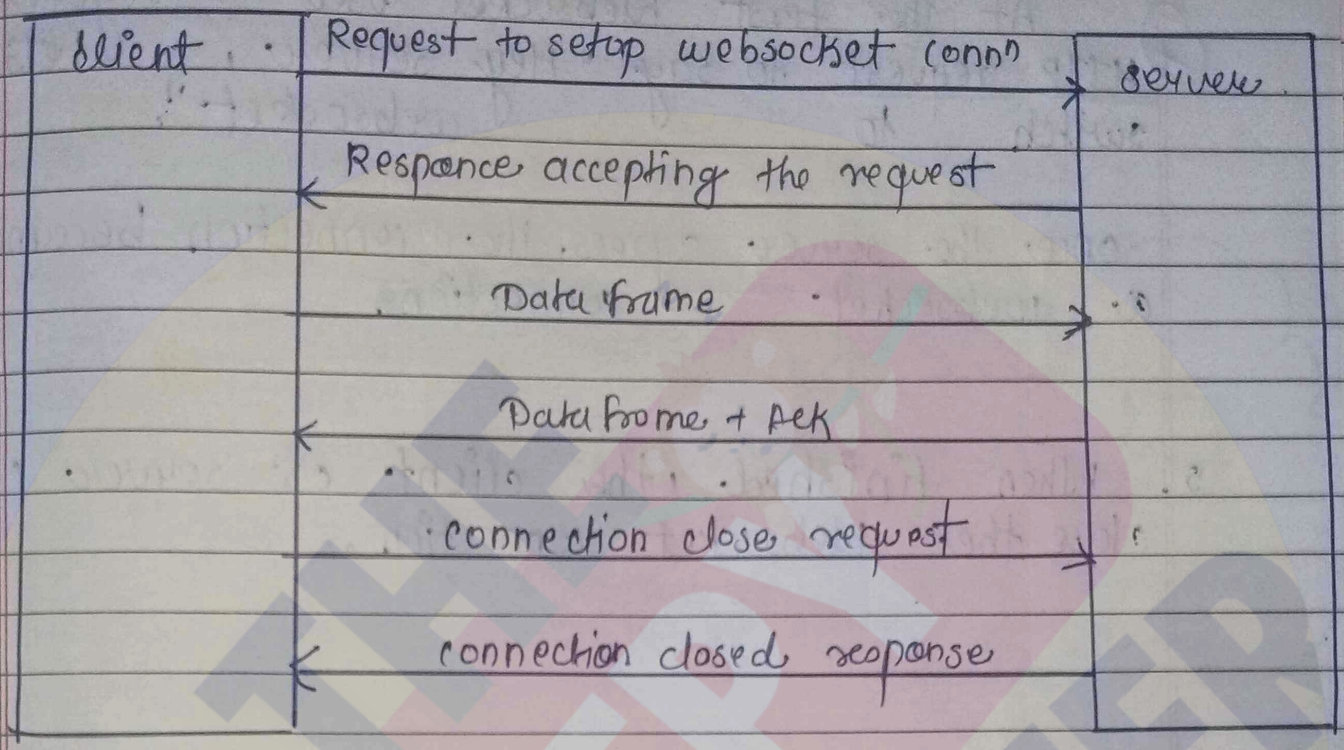
4. client → ~~Ret~~ Server
[mess]

REST applications expect each message to be self-descriptive. That means each message must contain all the information necessary to complete task.



mess.

WEB SOCKET BASED



1. Websocket allows Bidirectional full duplex comm^n bhw client and server.
2. Exclusive pair comm^n model
3. Normally, everytime you send message on the internet a new connection must be formed and then closed.

But, with websocket you make one connection and keep talking over it

4. Uses HTTP to start the connection :

At the first Websocket uses a regular HTTP request to say "Hey server, lets switch to websocket".

once the server agrees, the connection becomes a websocket connection.

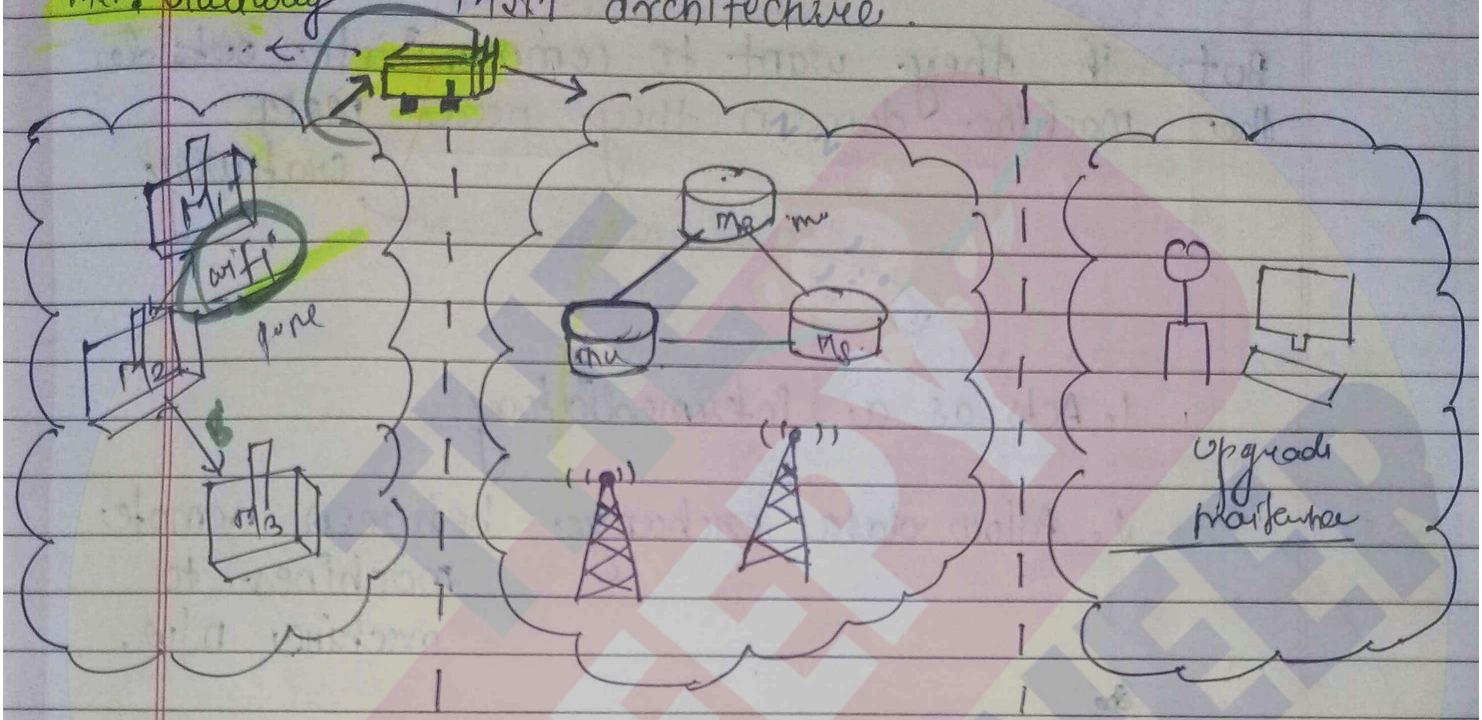
5. When finished, the client or server can close the Websocket connection.

4 PILLARS OF IOT...

M2M { Machine to Machine }

M2M Gateway

M2M architecture.



M2M domain

M2M Network domain

M2M Application Domain

1. IoT – IoT allows large # or lot of devices to connect with each other.
   [sensors, actuators].

2. M2M – M2M allows limited device to communicate with each other.

   It is designed to communicate between Machine and Machine.

3. In Machine domain, The machines communicate each other by wi-fi and Bluetooth.

But if they want to communicate outside the machine domain they need M2M Gateway.

1. Acts as a intermediatory
2. Allow data exchange between remote machines to machine n/w.

4. M2M Network domain :-

This network can be wired or wireless

5. M2M application domain :-

Here there will be maintance, service and upgradation for all the device in M2M.

Features of M2M :

1. Low mobility :

M2M devices do not move and if moves only within a certain area.

2. Small data Transmission :

Devices send or recieve small amounts of data.

pune → mumbai

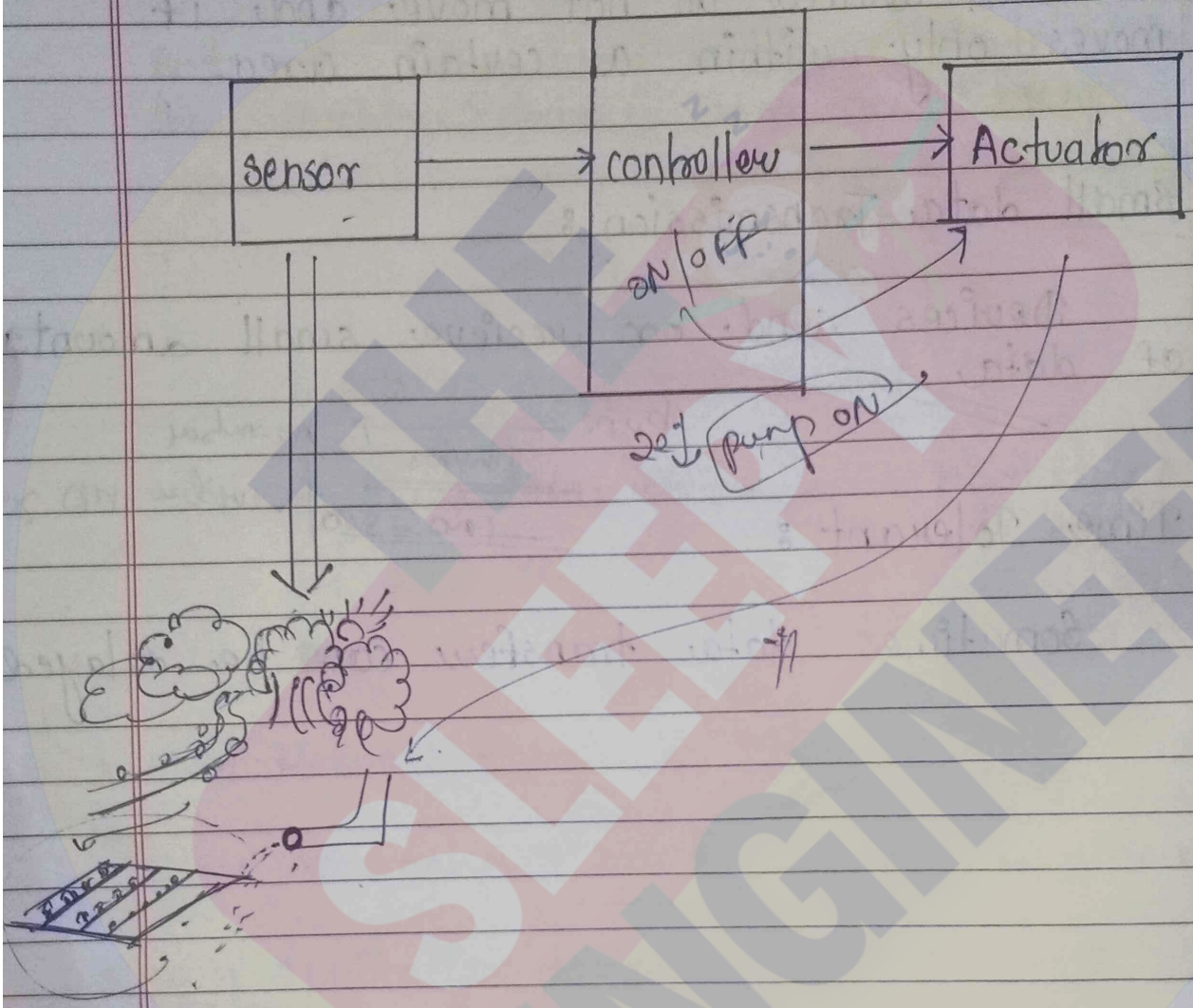
photos 100 - 500 | video no ✗

3. Time Tolerant :

Sometimes data transfer can be delayed.

SCADA

SUPERVISORY CONTROL AND DATA ACQUISTION



SCADA.

Supervisory

SCADA is a software which is installed in a computer or machine for supervising or Monitoring.

Controlling

- control the process of machine with help of SCADA.
- We can turn on or off Any device with help of SCADA.
- We can change or modify parameter value using SCADA.

Data Acquisition

SCADA has legal rights to collect and store data recieved from sensors and actuator.

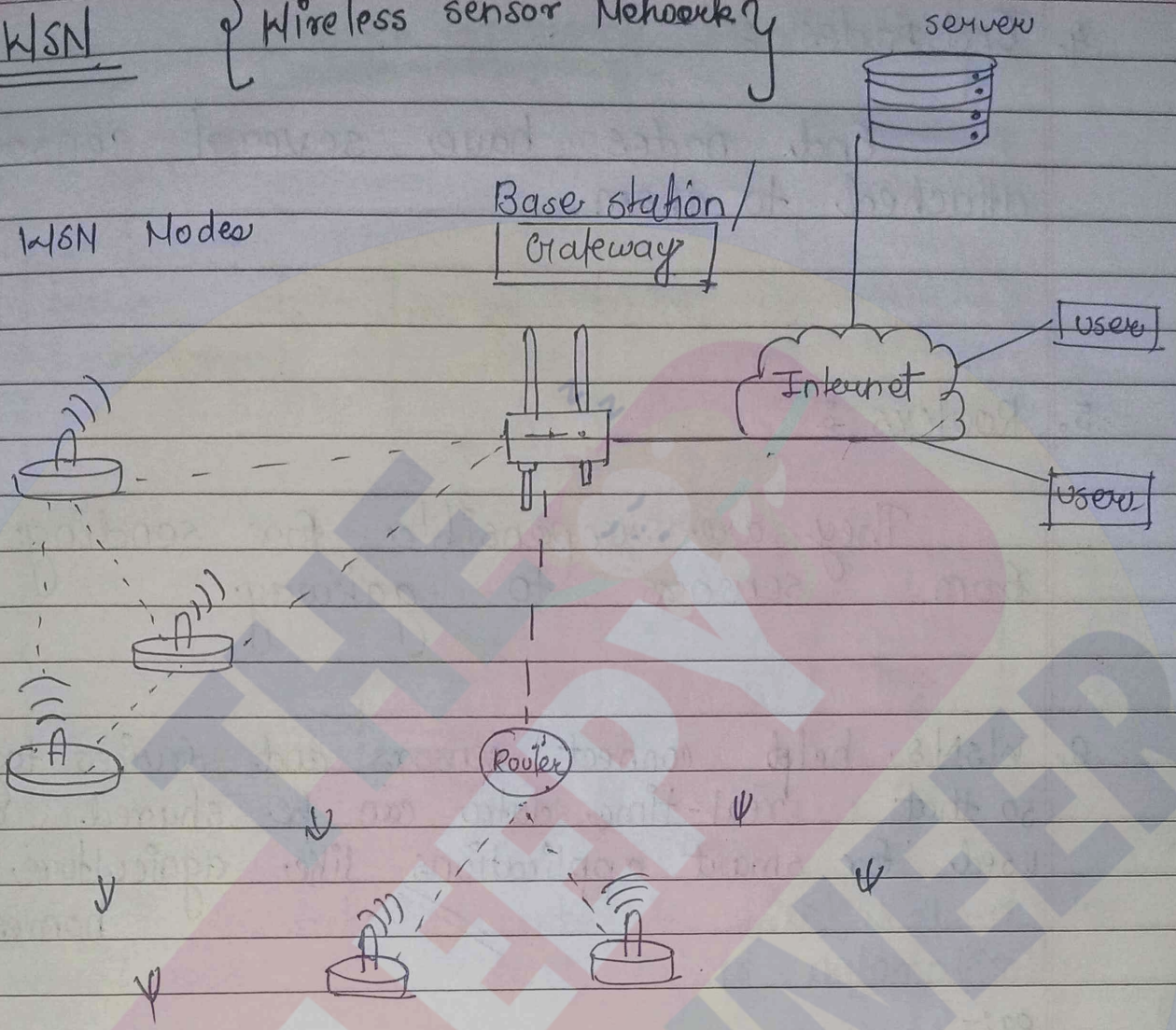
By using this data SCADA can enerate Warnings or Alarms.

Features

1. Real Time monitoring.
2. Alarm Management
3. Data Analysis and Reporting.
4. Scalability and flexibility.

SCADA system can expand easily by adding more sensor, collector etc.

WSN {Wireless sensor Network}



1. WSN comprises/consist of devices which are distributed over the area.

2. These devices are sensor, routers and gateway.

3. These devices all to-gether work to monitor the enviromental and physical conditions.

4. End nodes :

End nodes have several sensors attached to them.

5. Routers :

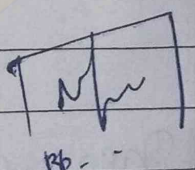
They are responsible for sending data from sensors to gateway

6. WSNs help connect sensors and devices together so that real-time data can be shared and used for smart applications like agriculture, homes.

eg:-

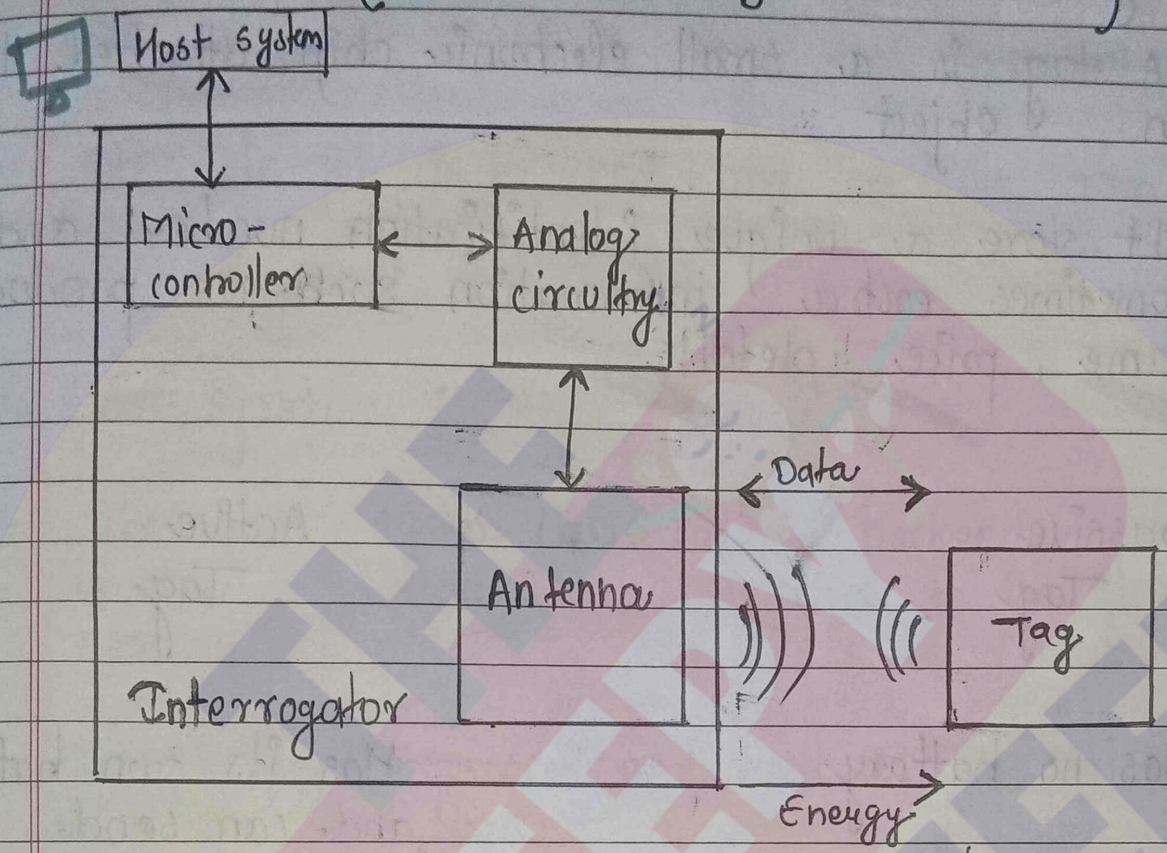
1. BSN (Body Sensor Network)

used in healthcare to monitor the human body (like heart rate , BP).



BP.

RFID {Radio frequency Identification}



1. RFID is a technology which is wireless used to identify and track objects automatically
2. RFID uses radio waves for tracking.
3. RFID is faster and more accurate object Identification.

Components

1. Tag :-

1. Tag :-

A tag is a small electronic chip attached to an object

It store a unique identification number and sometimes extra information such as product name, price details.

Passive Tag	Active Tag
- Has no battery	- Has its own battery and can send signals.
- Cheaper and most commonly used.	- used for long distance

2. Interrogator (Reader) :-

- It sends out an electromagnetic wave to activate ~~her~~ tag.
- once activated, the tag transmits its ID back to the reader.
- Reader can also be connected to a computer to ~~collect~~ transfer collected data.

3. Antenna :-

It is used in both tag and reader

It's main job is to send and recieve radio frequency signals.

4. Host system.

- The host system (also called backend system) is the computer connected to the RFID reader.

- It stores, processes and manage all the data collected from tags.