9) → The industrial Internet of Things (IToT) is the use of smart sensors and actuators to enhance manufacturing and industrial processes.

→ IIoT also known as the industry internet or Industry 4.0

→ IIoT uses the power of smart machines and real-time analytics to take advantage of the data that 'dumb machines" have produced in industrial setting for years

→ IIoT is better at communicating important information that can be used to drive buisness decisions faster and more accurately.

→ IIoT holds great potential for quality control, sustainable and green practices, supply chain traceability, and overall supply chain efficiency

→ IIoT is a network of intelligent devices connected to form systems that monitor, collect, exchange and analyze data. Each industrial IoT ecosystem consists of:
  • connected devices that can sense, communicate and store information about themselves.
  • public and / or private data communications infrastructure
  • analytics and applications that generate buisness information from raw data.

- storage for the raw data that is generated by the IIoT devices.
- people.

Drivers of IIoT:-

1) **Technology:** of smart sensors, Robotics & Automation, Augmented / Virtual Reality, Big Data Analytics, Cloud Integration, Software Applications, Mobiles, Low Power Hardware devices and scalability of IPv6 - 3.4×10^{38} IP addresses, etc is a major driver of the Industrial Internet.

2) **Customer Behavior:** The edge that IIot gives to the enterprises over their competitor helps them achieve better customer satisfaction and retention through value addition.

3) **Macro - Economic Drivers:** Government policies like Industry 4.0, Smart Factories, Make In India, Support of Green Initiatives, Rising Energy and crude oil prices etc. works totally in favor of the IIoT evolution.

Differences:

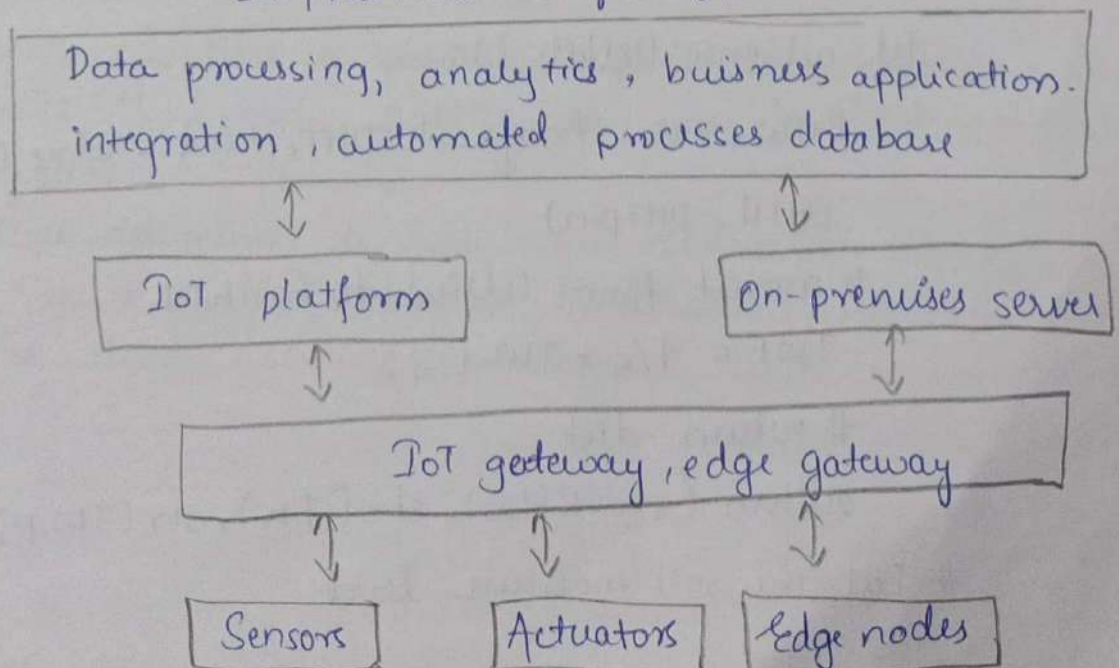| IoT	IIoT.
* It focuses on general applications ranging from wearable to robots and machines	*. It focuses on industrial applications such as manufacturing, power plants, oil and gas etc.

IoT	IIoT
* It deals with small scale networks.	* Deals with large scale networks.
* It offers easy off-site programming	* It can be programmed remotely i.e, offers remote on-site programming
* It requires identity and privacy.	* It requires robust security to protect the data.
* It needs moderate requirements.	* It needs stringent requirement.
* Short product life cycle	* long product life cycle
* less reliable	* highly-reliable

Infrastructure of IIoT

Data processing, analytics, buisness application. integration, automated procisces database

$\updownarrow$ $\updownarrow$

IoT platform   On-premises server

$\updownarrow$ $\updownarrow$

IoT gateway, edge gateway

$\updownarrow$ $\updownarrow$ $\updownarrow$

Sensors   Actuators   Edge nodes

7b) Web Socket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection. It enables the interaction between web server and web browser with lower overhead.

desktop
"Hello Server"
Mobile
"Hello Client"
Tablet
Server
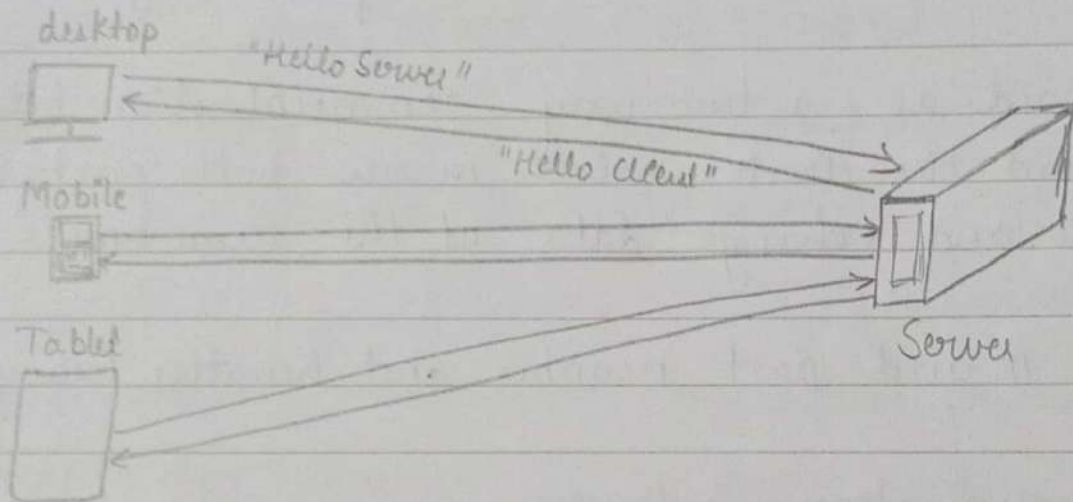
Server handshake with various clients

→ Web Socket uses HTTP as the initial transport mechanism, but keeps the TCP connection alive after the HTTP response is received so that it can be used for sending messages between the client and server

→ Web Sockets provides a persistent connection between a client and server that both parties can use to start sending data at any time

→ The client establishes a Web Socket connection through a process known as the WebSocket handshake. This process starts with the client sending a regular HTTP request to the server

Scanned with CamScanner

* Web Sockets is a stateful protocol, whereas REST is based on stateless protocol, i.e the client does not need to know about the server and the same holds true for server

* Web Socket connection can scale vertically on a single server.

* They are defined as a two-way communication between the servers and the clients, which means both parties communicate and exchange data at the same time

* It rely on IP and port number and handles heavy loads

* Cost of communication is lower

* This standard enables a new kind of applications. Buisness for real time web app can speed up with the help of this technology

* Web Socket URLs use the ws scheme. They are also used for secure web Socket connections, which are equivalent to HTTP,
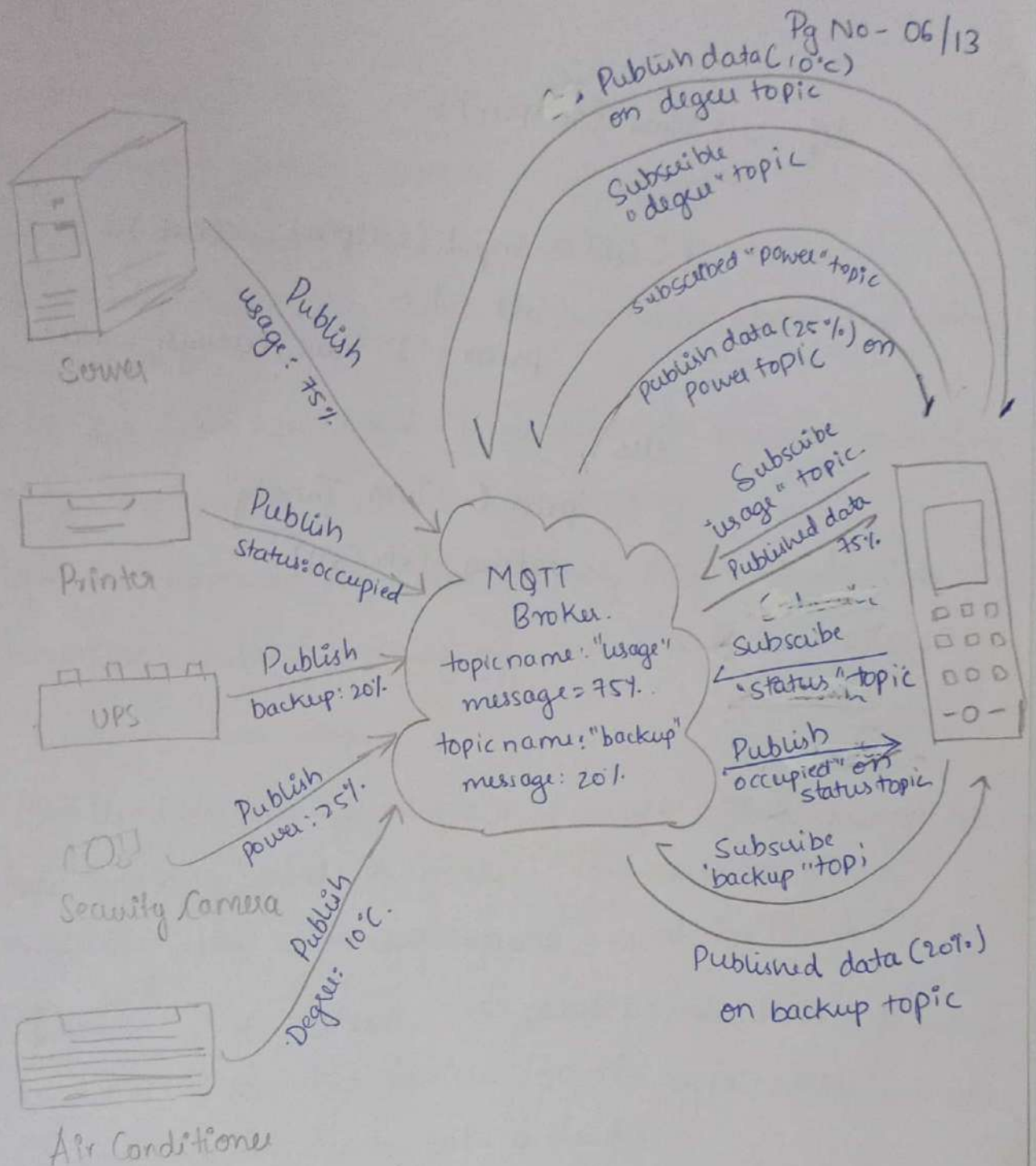
Ex:1) Multiplayer Games - Where heavy loads with lesser overhead is done

2) chat apps, Social feed where scalability is very much needed.

Thus, Web Sockets communication API is useful for Internet of Things.

8a)

Publish data (10°c) on deque topic

Subscribe "deque" topic

subscribed "power" topic

publish data (25%) on power topic

Server — Publish usage: 75%

Printer — Publish Status: occupied

UPS — Publish backup: 20%

Security Camera — Publish power: 25%

Air Conditioner — Publish Degree: 10°C.

MQTT Broker.
topic name: "usage"
message = 75%.
topic name: "backup"
message: 20%.

Subscribe "usage" topic
Published data 75%.

Subscribe "status" topic

Publish "occupied" on status topic

Subscribe "backup" topic

Published data (20%) on backup topic

MQTT uses publish subscribe messaging protocol allowing a message to be published once and multiple consumer (applications /devices) to recieve the message providing decoupling between consumers and producers
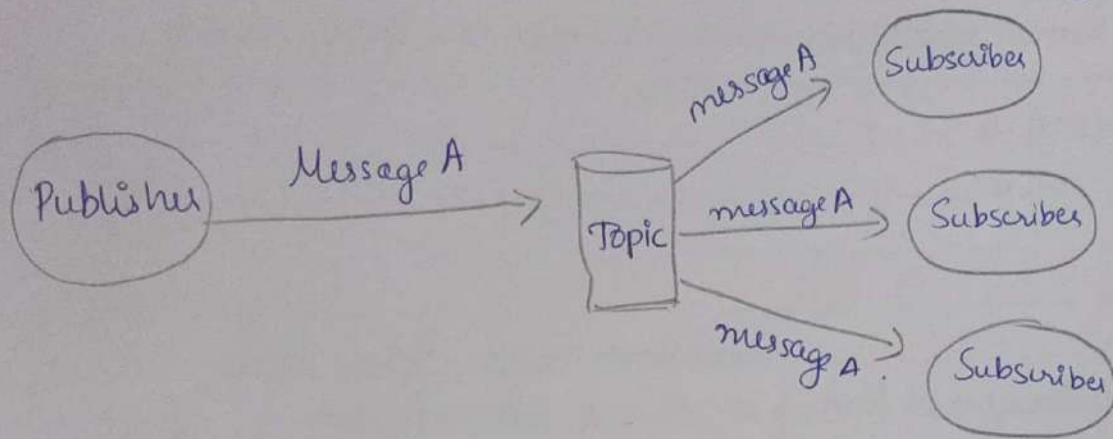
→ Publisher is the data source (eg. sensors)

→ They publish data on some topic and all the customers subscribed to it recieves the message.

→ Producer sends (publishes) a message (publication) on a topic (subject) A consumer subscribes (makes a subscription) for messages on a topic (subject)

→ If message sewer / broker matches publications to subscriptions

• If no matches messages are discarded

• If no or more matches the message is delivered to each matching subscriber / consumer.
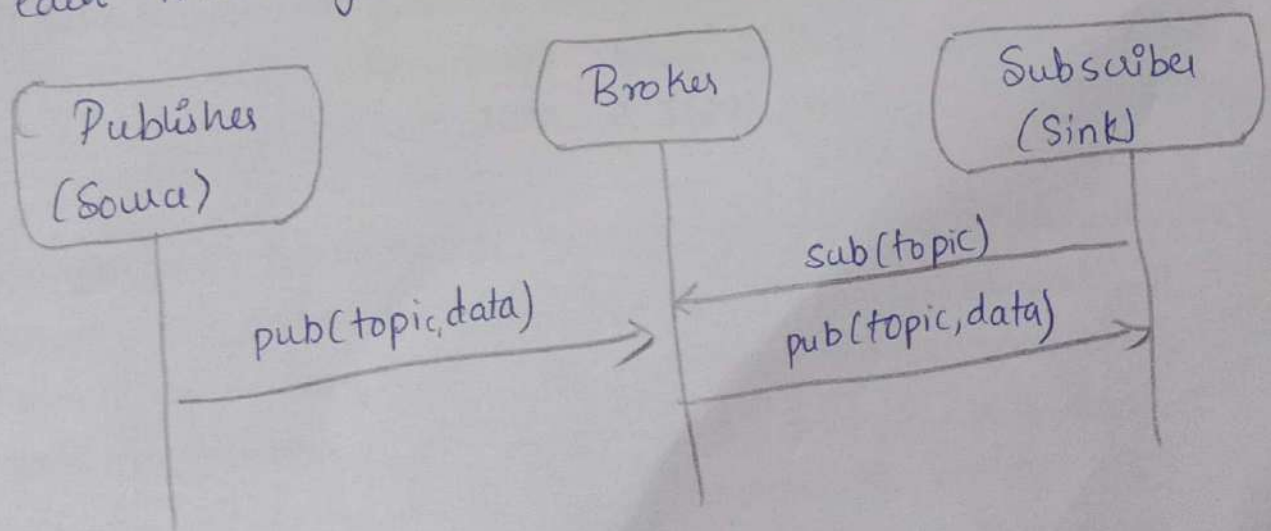
8b)

JSON (Java Script Object Notation)
- → It is a text format for storing and transporting data
- → JSON is "self describing" and easy to understand
- → JSON is text, written with Java Script Object Notation
- → JSON is lightweight data-interchange format
- → JSON is used to send data b/w computers
- → JSON is language independent

JSON format is syntactically similar to the code for creating Java Script objects. Because of this similarity, a Java Script program can easily convert JSON data into Java Script objects

JSON format is text only, it can be sent to and from a server, and used as a data format by any programming language. There is a built-in function for converting JSON strings into native Java Script objects : JSON. parse()

If data is recieved from a server, in JSON format, we can use it like any other Java Script object. It has built-in function for converting an object into a JSON string : JSON. stringify()

Exchanging Data :- It takes place between a browser and server, the data can only be text. JSON is a text, so you can convert any Java Script object into JSON and send JSON text to a server. We can also convert any JSON text recieved from the server into Java Script objects in the programe. No complicated parsing and translations

when storing data, the data has to be in a certain format, and regardless of where you choose to store it, text is always one of the legal formats.

JSON also makes it possible to store JavaScript object as Text

Ex: { " employees ": [

    { "firstName ": "John ", "lastName ": "Y", "Age":9},
    { "firstName": "bbb", "lastName": "D", "Age": 21},

    ] }

It is sent as a dictionary of lists of dictionary consisting key value pairs. This JSON has an array of employees.

JSON doesn't use end tag. It can use arrays.

HTML	XML
* Markup language for displaying web pages in browser. Designed to display data with focus on how data looks	* Markup language defines set of rules for encoding documents that can be read by both humans and machines
* Invented in 1990	* Invented in 1996
* Static	* Dynamic.
* Extended from SGML	* extended from SGML.
* Used to display web page	* Transport data between the application and the database. To develop other markup languages
* No strict rules. Browser will still generate data to the best of it's ability	* Strict rules must be followed or processor will terminate processing the file
* Presentation type	* Neither presentation, nor programming
* Tags are predefined	* Custom tags can be defined by the author.
* Cannot preserve white space	* Preserves white space

7a)

- RIOT is a small OS for networked, memory-constrained systems with a focus on low-power wireless IOT devices
- It is open-source software, released under the LGPL
- Due to this unclonable license and its large independent community RIOT is often referred to as the Linux of the Internet of Things

Features of RIOT are:
- There are no new programming environments. C or C++ can be used directly with existing tools like gcc, gdb, etc
- Less hardware dependent code
- Supports 8-, 16- and 32-bit microcontroller platforms
- Energy efficiency is maintained
- Less interrupt latency, so real-time capability is ensured
- Multithreading is enabled
- Supports the entire network stack of IoT (802.15.4 Zigbee, 6LoWPAN, ICMP6, IPV6, RPL, CoAP, etc)
- Both static and dynamic memory allocation
- POSIX compliant (partial)
- All output can be seen in the terminal if hardware is not available; however, there is a visualization tool called RIOT-TV that is provided
- flexible memory management
- RPL (storing mode, P2P mode)
- high resolution, long-term timers
- a preemptive, tickless scheduler

1. Code for interfacing soil moisture sensor with Raspberry Pi.

```python
import sys.
import    RPi. GPIO as GPIO.
import    os
import    time import sleep.
import    Adafruit - DHT.
import    urllib 2

DEBUG = 1.
 # setup the pins we are connect to.
  Soilpin = 14
  DHTpin  = 23
  # set up our API and delay.
  myDelay = 15  # how many seconds between posting data.
GPIO. setmode (GPIO. BCM)

GPIO. setup (soilpin , GPIO. IN , pull - up- down =
                    GPIO. PUD-DOWN)

def getSensorData ():
   RHW, TW = Adafruit_DHT. read - retry (Adafruit_DHT,
    DHT11, DHTpin)
   # convert from Celsius to Farhett.
    TWF = 9/5 * TW + 32.
    # return dict
    return (str (RHW), str (TW), str (TWF))
  # Test the soil moisture level.
```

```
def soil time (soilpin):
        ST = 0
        if ( GPIO. input (soilpin) == True ):
                ST+ = 1
                print 'I have enough water'

    else:
                print: 'Iam Thirsty'
                return (str (ST))
```