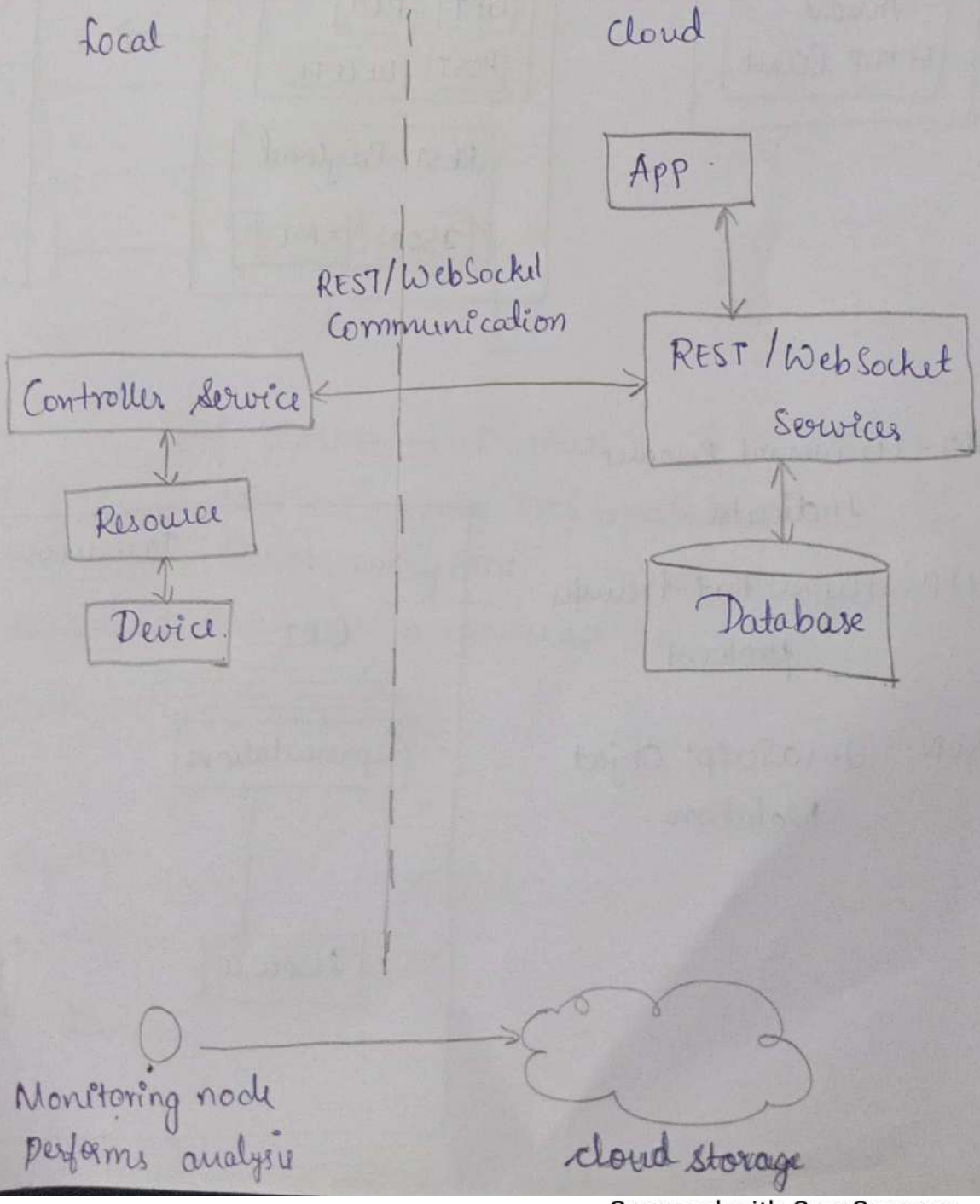
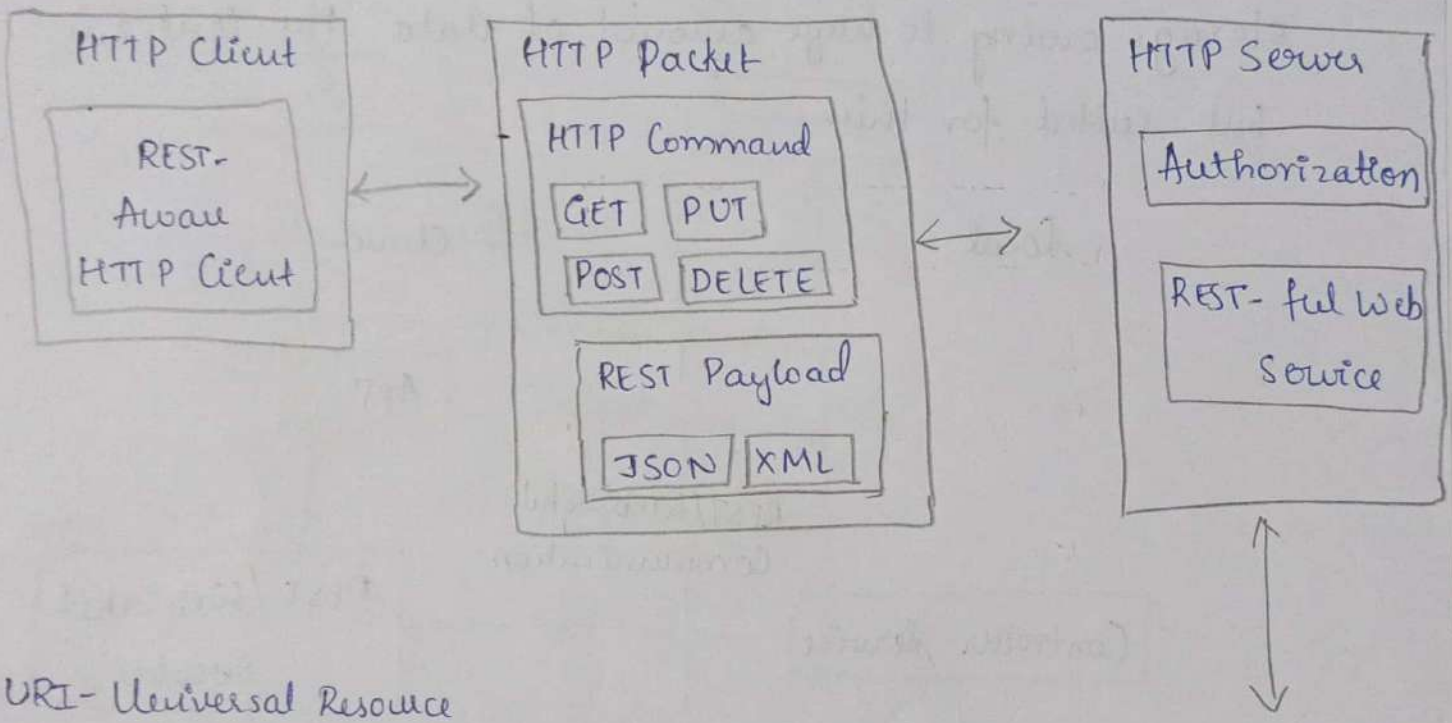
2b) A-level 2 IoT System is best suitable for the scenario of automating existing electrical appliances at home, as automation of electrical appliances is not computationally intensive and a single node at home is sufficient for collecting data. As cloud based application and data storage owing to huge amount of data the level-2 is best-suited for this.



1a) REST- Based Communication API's and its constraints:

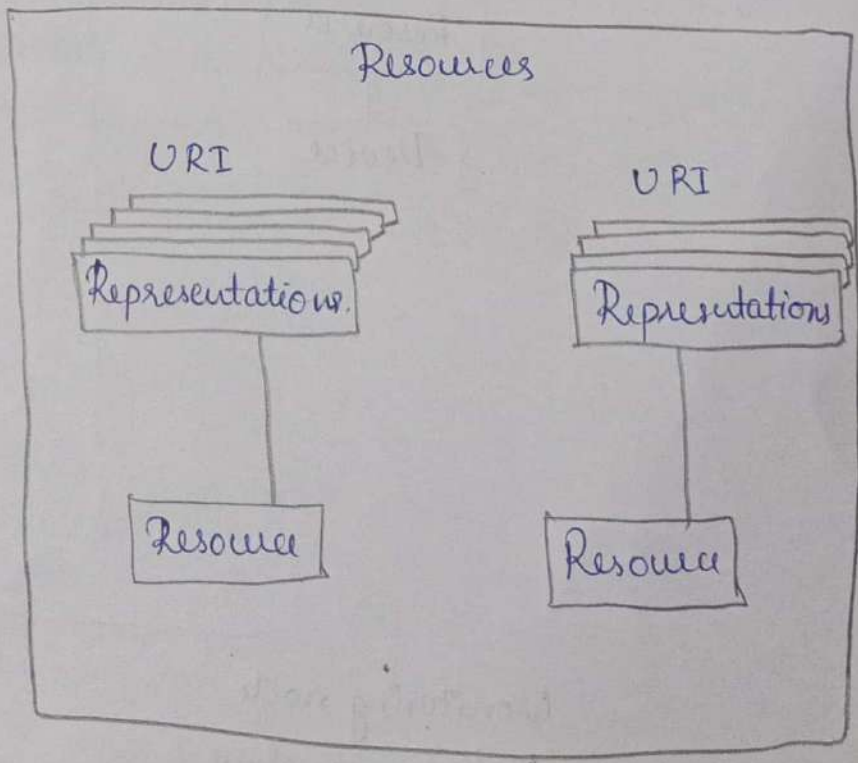
Representational State Transfer (REST) is a set of architectural principles by which we can design web services and web API's that focus on a system's resources and how resource states are addressed and transferred.



URI- Universal Resource Indicator

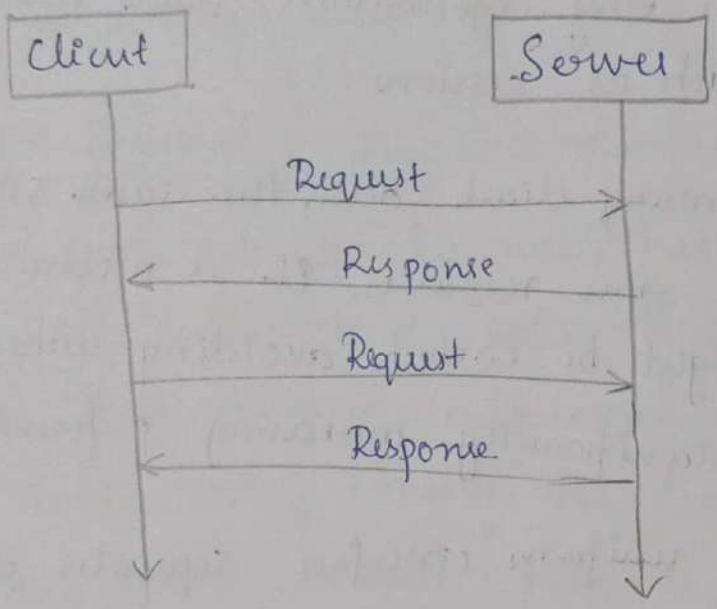
HTTP- Hyper text transfer protocol

JSON- JavaScript Object Notation.



REST APIs follow the request - response communication model. The REST architectural constraints apply to components, connectors, and data elements, within a distributed hypermedia system.

Constraints:



* It is based on client - server architecture.
* It is stateless it does not store information of requests, session or clients. Hence, everytime a new connection is to be established that is a overhead as well as time - consuming.
* Cacheable
* Layered System.
* Uniform Interface.
* Code on demand.

Stateless: One client can send multiple requests to the server; however each one of them must be independent, that is every request must contain all the necessary information so that server can understand it process it accordling. Server can't hold information of the client state. Any information status must stay on the client - such as sessions

Cacheable: Because many clients acess the same server, and often requesting the same resources, it is necessary that these responses might be cached, avoiding unnecsary processing and significantly increasing performance.

Client-Server: The uniform interface separates clients from servers. This separation of concerns means that, for example clients aren't concerned with data storage, which remains internal to server, so that the probability of client code is improved. Servers are not concerned with the uses interface or user state so that servers can be simpler and scalable. Servers and clients may also be replaced and developed indipendently, as long as the interface is not altered.

Layeid System: A client cannot ordinarily tell whether it is connected directly to the endserver, or to an intermediary along the way. Intermediary servers may improve the system scalability by enabling load-balancing and by providing shared caches. Layers may also enforce security policies.

Code-On-demand: This constraint allows the customer to run some code on demand, that is, extend part of server logic to the client, either through an applet or scripts. Thus, different customs may behave in specific ways even using exactly the same services provided by the server. As this item is not the part of architecture itself, it is considered optional. It can be used when performing some of the client side services which are more-efficient or faster.

Uniform Interface: It defines the interface between clients and servers. It simplifies decouples the architecture, which enables each part to evolve independently.

- HTTP verbs (GET, POST, PUT, DELETE)
- URIs (Resource names).
- HTTP Response (status & body.

Resource is represented to user as XML, JSON, HTML, TXT. Self-descriptive messages to describe how to process the message.

Eg: → Host: dxe.com.
→ Accept: application/JSON. // Data format
→ User-Agent: Chrome/version. // browser
→ GET #1/user/sneha HTTP 1.1.
request — protocol — version

1b) All of these are IoT enabling technologies that strengthen IoT applications.

WSN's or wireless sensor networks have multiple nodes, routers connected together that can monitor the status of the Parking System. It has sensors to detect the current allocations or free slots in parking system.

Big-Data Analytics: Suppose the parking-system is located at some mall or store. The data collected on occupancy could be related to customers and geographical data to get could some useful insights.

We could analyse the data by storing it on cloud (PaaS / IaaS) Eg: S3 buckets of Amazon, and use Buisness Intelligence tools to visualize it also encorporate the Big Data Analytics tool to make furthe predictions like when is the customer visit time high. After on an average how many hours customers leave the mall.

Cloud: IaaS could be used to store ~~and~~ the data. PaaS to deploy, build & manage the user-friendly application that is cloud-based SaaS - delivers this application to the users

Sensors could be IR sensors, bluetooth for communication etc.

1. Wireless Sensor Networks (WSN)

→ Wireless Sensor networks comprises of distributed devices with sensors which are used to monitor the environmental and physical conditions

WSN consists of
1) number of end-nodes
2) routers
3) coordinator

Ex: Weather monitoring system, Smart grids, health monitoring, soil moisture monitoring system etc

2. Big-Data Analytics : It is the process of collecting, organizing and analyzing large sets of data (called Big data) to discover patterns and other useful information

It involves several steps from data cleansing, data mining, data processing and visualization

All these data are collected and stored in warehouses

Ex: Web-data, e-commerce, purchases at department/grocery stores, Band/Credit card transactions, Social network

Characteristics of Big Data:

1) Variety: Data can be structured, unstructured, semi-structured or mixture of all

2) Velocity: It refers to speed at which data is processed

It may be in batches, real-time and streams

3) Volume: It refers to amount of data

It might be in terabytes, petabytes

3. Cloud Computing: It is practice of using a network of remote servers hosted on the Internet to store, manage and process data, rather than a local server or personal computer.

Cloud Service Models

→ 1) IaaS

Provides capability to the consumer to hire infrastructure components such as servers, storage, and network

Ex: AWS, Google Cloud

2) Paas

Capability provided to consumer to deploy consumer-created or acquired applications on the Cloud provider's infrastructure.

Ex: Cloud9

3) Saas

Capability provided to the consumer to use provider's applications running in Cloud infrastructure

Ex: Google sheets/forms/doc/drive etc.

2a) The adapter layer fragment & reassembles IPV6 packets. The implementation of adaption layer enhances the routing / forwarding decision of packets both network and adaption layer.

Depending on what layer the routing mechanism is located, two categories of routing is defined:

1) Mesh-Under Technique

2) Route-Over Technique.

It dramatically reduces the IP transmission overhead by using header compression and fragments IPV6 packets to support minimum MTU (maximum transmission unit) of IPV6 packets. The routing protocol in 6LoWPAN is sensitive ∵ capabilities of nodes are limited in terms of energy, transmission range etc

1) Mesh under uses the layer-2 link layer addresses (IEEE 802.15.4 MAC) to forward data packets

2) Route-over uses layer three (network layer) addresses (IP address)

1) Route-over (layer three forwarding)

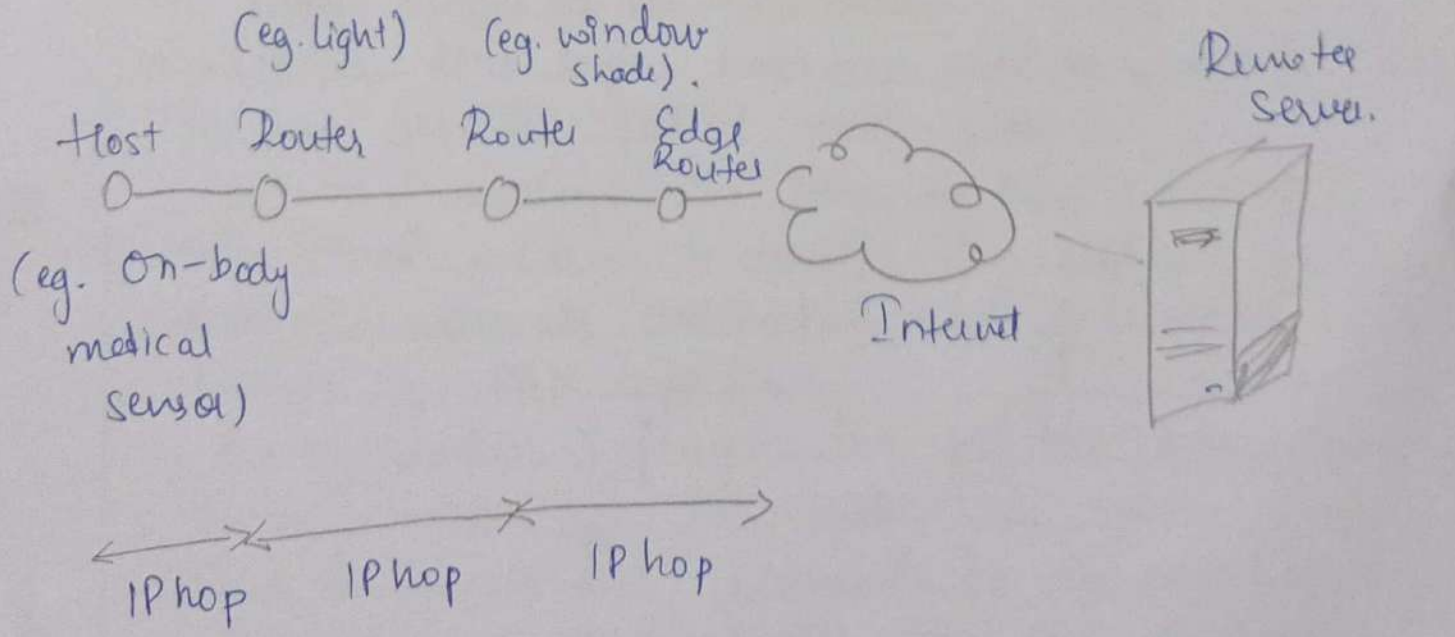
→ In route-over networks the routing takes place at the IP level as mentioned before, thus each hop in such networks represents one IP router. The usage of IP routing provides the foundation of large and more powerful and scalable networks.

→ Since every router must implement all features supported by a normal IP router. The most widely used routing protocol for route-over 6LoWPAN network today is RPL as defined by IETF in RFC 6550.

→ In this scheme, all routing decisions are taken in the network-layer where each node acts as an IP router. In route-over each link-layer hop is an IP hop. the IP routing supports the forwarding of packets b/w these links

→ In forwarding process IP routing tables and IPV6 hop-by-hop options are used. For routing & forwarding processes the network layer takes decision using the additional encapsulated IP header. The adaption layer map b/w the frames & IP headers.

→ When an IP packet is fragmentd by the adaption layer, fragments are sent to the next hop based on routing table information. ~~when~~ If the packet is destined for itself, the network layer sends the IP packet to the transport layer, otherwise forwards the packet to the next hop based on the routing table information. If there are one or more fragments missing, then all fragments are retransmitted to one hop distance. After recieving all fragments sucessfully the adaption layer creates an IP packet from these fragments and pass it to the network layer.



1. Mesh under - technique.

→ In mesh-under scheme, the network layer does not perform any IP routing inside a 6LoWPAN.

→ The adaption layer performs the mesh routing and forwards packets to the destination over multiple radio hops

→ In mesh-under scheme, routing and forwarding are performed at link layer based 802.15.4 frame or the 6LOWPAN header

→ To send a packet to a particular destination, the EUI 64 bit address or the 16 bit short address is used and sent it to a neighbor node to move the packet closer to the destination

→ Multiple link layer hops are used to complete a single IP hop soit is called mesh-under

→ 6LoWPAN employs the idea of the originator and the final address to describe the original source and the ultimate destination node of a single IP hop within a PAN respectively

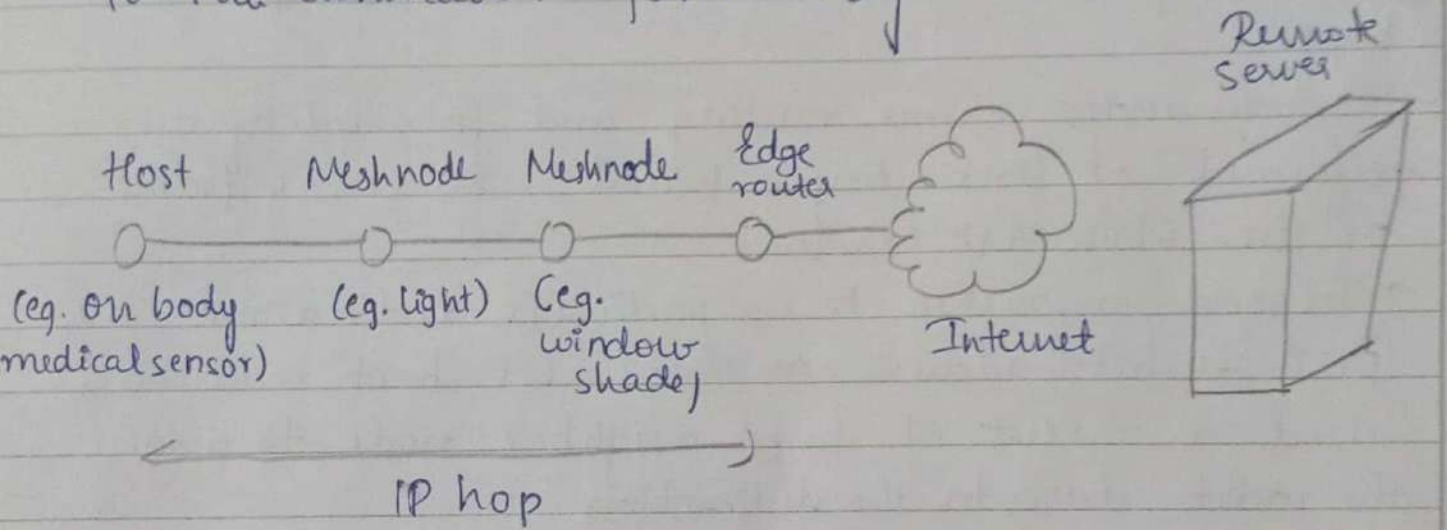
→ An IP packet is fragmented by the adaption layer to a number of fragments

→ These fragments are delivered to the next hop by mesh routing and eventually reach the destination.

→ Different fragments of an IP packet can go through different paths and they are gathered at the destination.

→ If all fragments are reached successfully, adaption layer of the destination node reassembles all fragments and creates an IP packet

→ In case of any fragment missing in the forwarding process the entire IP packet is retransmitted to the destination for recovery



Source	Intermediate node	Intermediate node	destination
5. Application layer	5 AL	5 AL	5 Application layer
4. Transport layer	4 TL	4 TL	4. Transport layer
3. Network layer	3 NL	3 NL	3. Network layer
2 Data link layer	2 DLL	2 DLL	2. Data link layer
1 Physical layer	1 PL	PL	1 Physical layer

AL - Application layer
TL - transport layer
NL - network layer
DLL - data link layer
PL - Physical layer

In intermediate it passes only through data link layer

4) Types:

1) Business models

2) Revenue - Sharing.

3) Cost - savings Sharing.

4) Product - Sharing

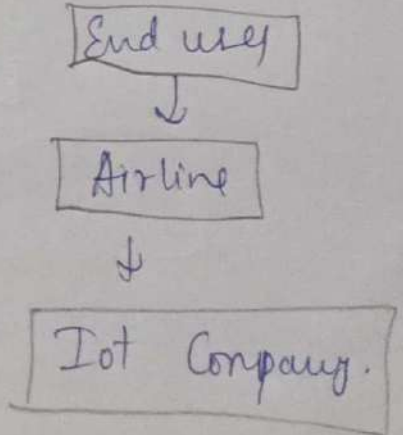
5) Product-as-a-Service.

6) Performance - as - a - product.

7) Transactional.

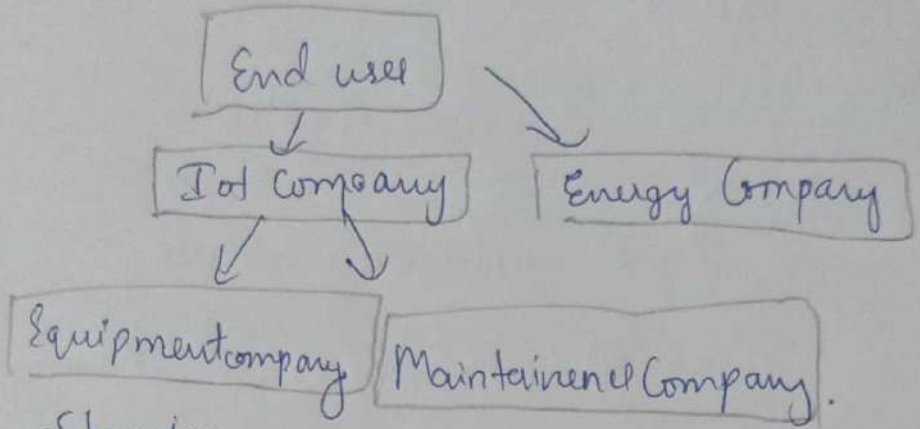
1) Revenue - Sharing: Luaga lost in air transit.

IoT Soln: A tracking device is placed inside the luggage and transmits its location using 2G. The user can track his luggage using a smartphone app



2) Costs saving sharing

Home / building energy consumption.



3) Product - Sharing

→ Relatively high investment & maintainence.

→ The end user buys the car upfront and pays for its ongoing maintenance, fuel & insurance.