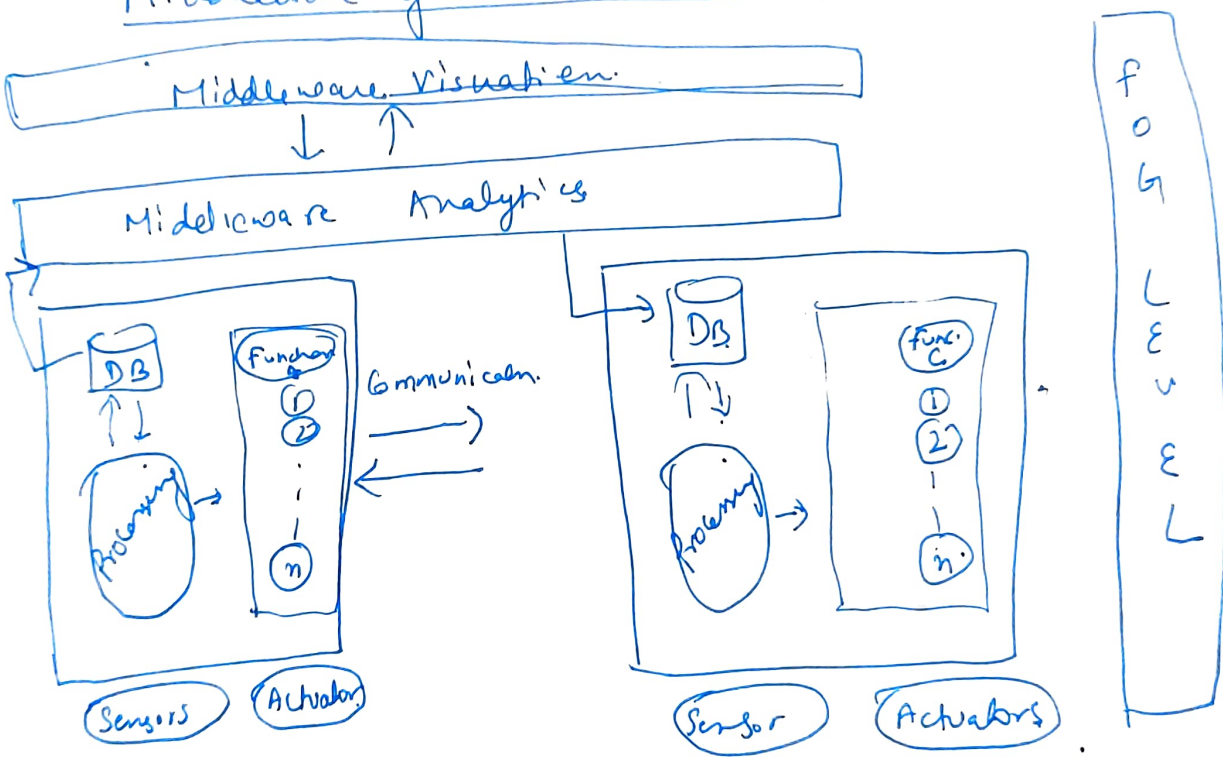
UNIT-5

⇒ MIDDLEWARE OF FOG & EDGE COMPUTING

The Existing IoT Systems gather data in Central place (cloud) for post processing & Situation awareness. Such a Collection & Computing method can not Sustain due to data growth; the Communication bandwidth is limited by Spectrum & Can't Catch up the data growth rate. The Middleware Enables a Mesh N/w Composed of many devices to share resources & Capabilities

Middleware general architecture.



Every Node has Small but powerful Computational Unit Nodes are able to Store raw & processed Info in the database. that is designed for Compressed data. Inside each processing unit, Nodes are able to manage I/I data generate real time Info.. Every Node is able to Communicate & cooperate with Neighbours using wireless. bluetooth

Middleware act as bridge b/w diverse technology, tools & DB So that you Can Integrate from Seamless into Single Systems

→ Middleware Enables the flow of real time info access within or among N/w

→ It helps streamline process & Improves efficiency

→ It is able to Maintain the Integrity of Info across Systems

→ It helps to create different types of N/w application.

→ It has the Capacity to Implement logic based on the request made by client.

→ It plays Imp. role in Concurrent processing, load balancing & Transaction Management

→ It helps to Secure access to back End resources

⇒ DESIGN GOALS

→ Middleware Identify the Issues design in wireless & Mobile Environments

→ Detailed design for an Image transcoding proxy & appl^n Session

→ Resource sharing, Shorter & Service response time. & Collaborative Service.

→ Lower data transmission load, faster & more accurate analysis.

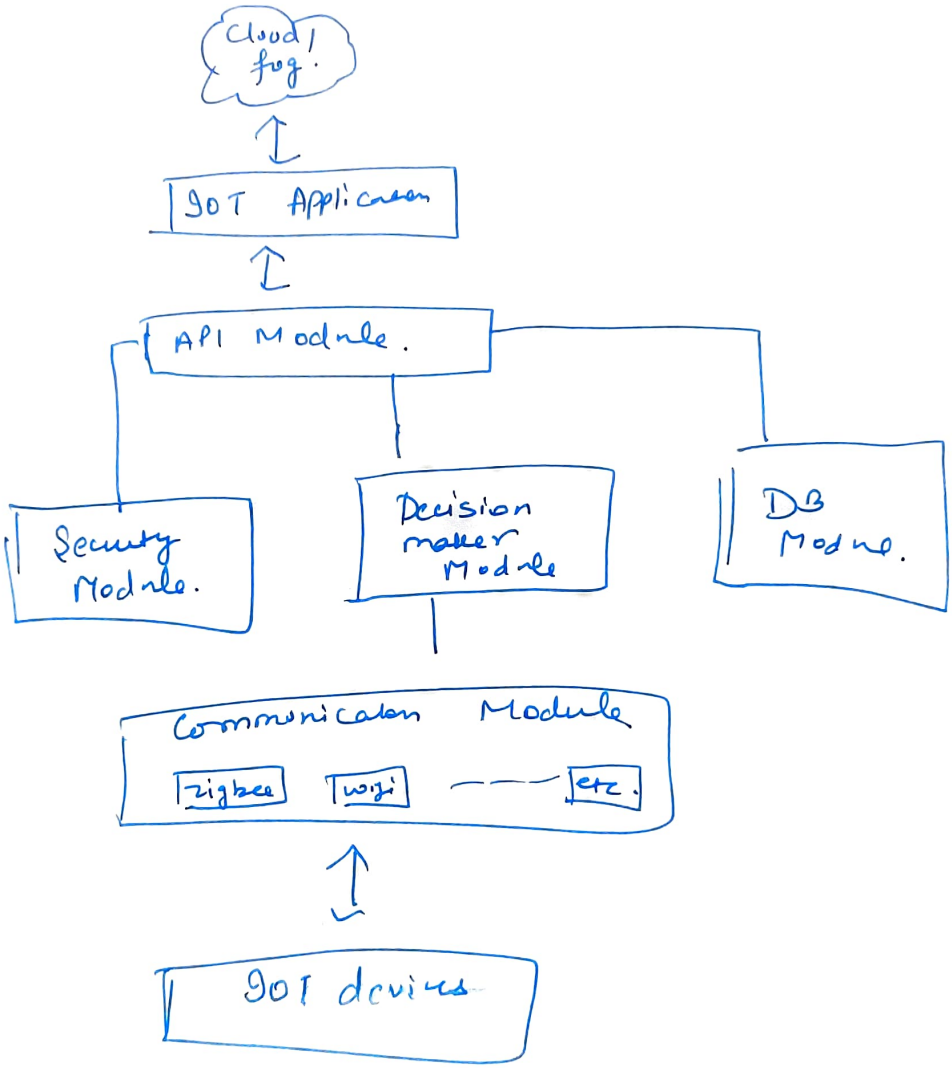
→ Latency - Low latency should be.

→ Efficiency - Efficient utilisation of resources & Energy

→ Generality - Due to heterogeneity of fog Node & client, we need provide same abstract to top layer & Appl^n Services for fog client

⇒ MIDDLEWARE SYSTEM ARCHITECTURE

Architecture consist of 4 distinct components IoT devices, Middleware, fog & cloud.
IoT devices collect data from surroundings & take basic security decisions. while Middleware take more advanced decision on whether data should be processed on fog or cloud.

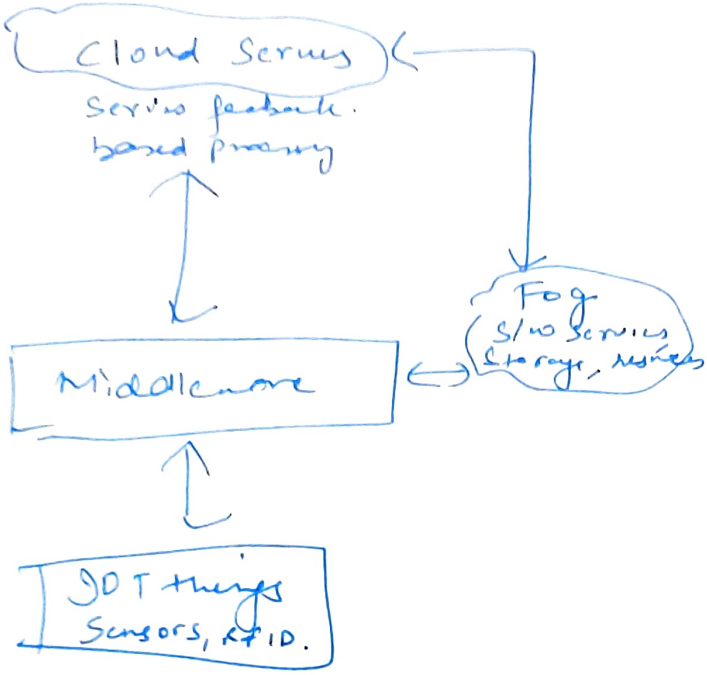


The Communication Module is in charge of communicating with various IoT devices.
Easy Integration, development & transparent data exchange with different IoT Entities. for this purpose protocols like Zigbee is used.

Decision maker Module has access to database of authorised things using their ID's

API Modules offers an Integrated & Improved access Interface for IoT applicatn.

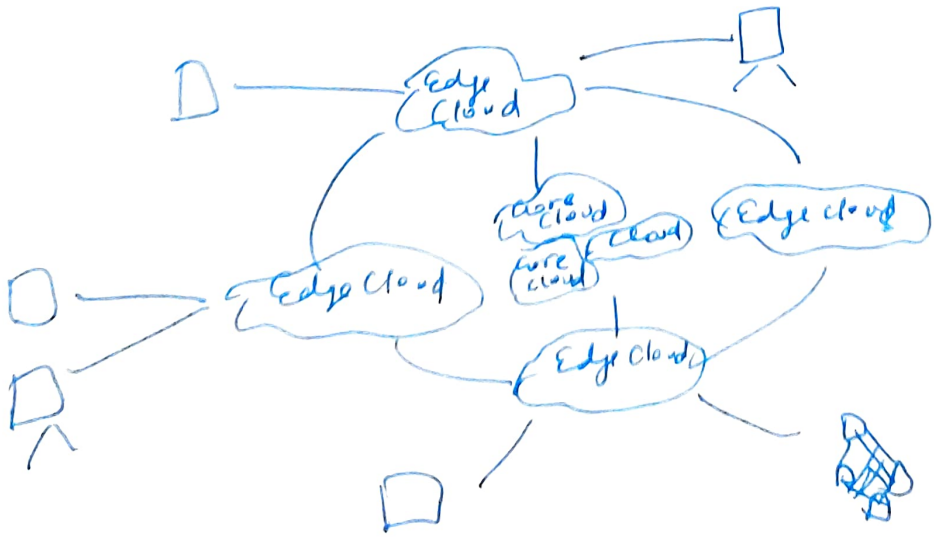
IoT applicatn - These are not resource Constrained Standard Security Sol. like SSL to Send request over Http.



=> CLUSTERS FOR LIGHT WEIGHT EDGE CLOUDS.

Cloud technology is moving towards more distribution across Multi Clouds & the Inclusion of various devices. Lightweight virtualisation Sol. are beneficial for this architectural setting with smaller but to manage solution.

Cloud edge Computing is pushing Computing applicatns, data & Service away from Centralised cloud data Centre architecture to edge of N/w.

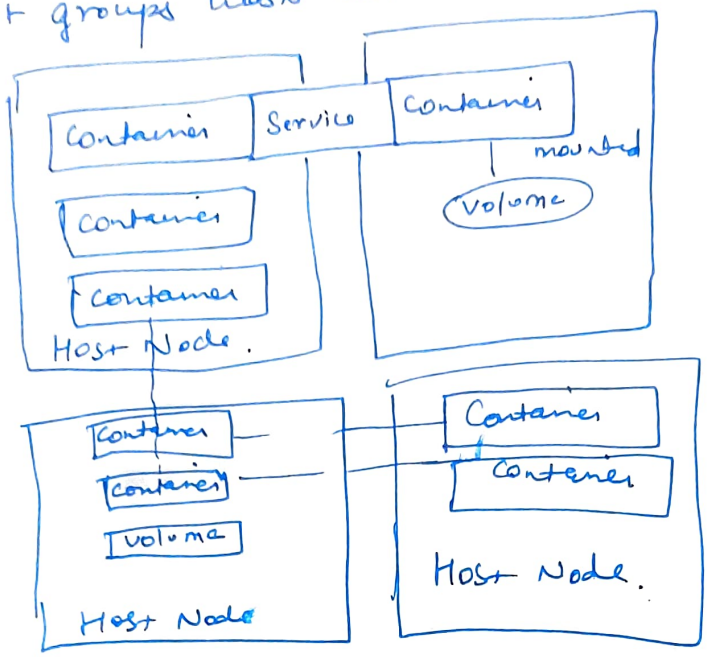


Resource architecture as cluster based Container architecture

In Centre & Edge clouds, but also IoT objects linked to Compute & Storage resources. & Applⁿ need to be Managed. i.e packaged, deployed.

Containers Clusters

It groups hosts into clusters.



Container based Cluster architecture.

Each cluster Consist of several host Nodes - where Nodes are virtual servers. Each host holds several containers with Common Services like Scheduling, load balancing & Applications. Each Container can hold Continuously provided Service payload Service. Links allow 2 or more Containers to Connect & Communicate.

Edge clouds move the focus from heavy weight data centre clouds to more light weight resources.

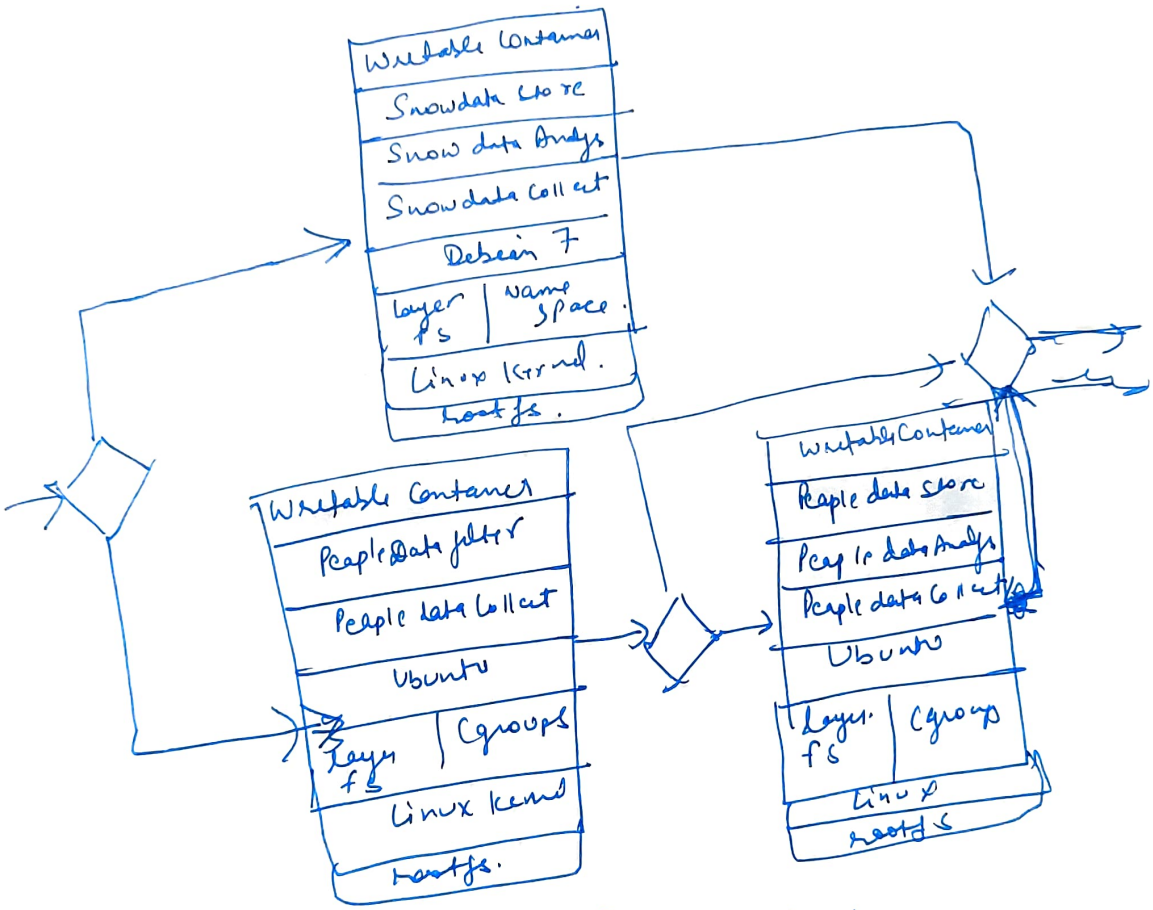
In the following we Explain how to build platform that are lightweight in terms of H/W & S/W.

① Lightweight Software - Containerization

Containerization that allow a light weight virtualization through the Construction of Containers. as application Packages from individual Images. This address performance & portability weakness of Current cloud Solutions.

Orchestration is about Constructing & Managing a possibly distributed assembly of container based S/w applⁿ. Container Orchestration is not only about Initial deployment Starting & Stopping of Containers but also about the Mgmt. A Multi-Pass based on Container clusters Can serve as Sol. for managing distributed S/w applⁿ

Orchestration plan for Ski resort Case study



(2) Light weight Hardware. — Raspberry Pi clusters.

Raspberry Pi's as the H/w device Infrastructure & Illustrate orchestration for Raspberry Pi clusters in Eg. cloud Env.

A Raspberry Pi (RPi) is relatively cheap & low power Consumption which make it possible to create an affordable & Energy efficient cluster that is suitable for Environment for which high-tech. Installation not Possible

=> NEED FOR FOG & EDGE COMPUTING MIDDLEWARE

Fog & Edge Computing are gaining acceptance due to high availability, low latency & low cost.
The Efficient design of middleware enables the realization of fog & Edge architectures. Middleware handles different tasks like N/w Mgmt, task Scheduling, Mobility Management, Security Management thereby reducing complexity of distributed mobile Appl^n design.
Middleware design is challenging because of appl^n requirements such as

→ Availability of Content on Sensing devices
→ Cost of data transfer
→ Limitation in No of edge device present
→ Strict latency constraints

=> DESIGN GOALS OF MIDDLEWARE

(1) Adhoc device discovery —
Data Sources in fog/Edge belong to wide category of devices ranging from IoT Sensors, mobile devices to fixed Servers. A channel of communication needs to be setup b/w requesting devices & adhoc discovered devices.

(2) Run time Execution Environment —
Middleware provides platform that Executes appl^n task remotely on Edge device.

(3) Minimal Task Disruption —
It affects the reliability of Execution of task.
It results in reinitialization of task. or unavailable results
Anticipatory ~~task~~ techniques can be used to Minimize the Interruption in task.

(4) Overhead of operational parameters

Establishg Comm. b/w adhoc edge devices, Selection of Candidate edge devices b/w Multiple edge devices. Incurs additional usage of Bandwidth & Energy Consumption on Edge devices. As these resources are expensive, minimizing the operational parameter is an Imp. aspect.

(5) Quality of Service - It is highly dependent on application. Many fog/edge appl" use Multidemensional data for achieving Specific goals.
Acquiring & processg such huge sensor data w.r.t. real time Constraint is requirement for these appl".
Real time response is an Imp. QOS measure.

→ STATE OF ART MIDDLEWARE INFRASTRUCTURE

Real time appl" that process the request for emergency rescue in disaster & Searchg for Missg persons.
Common requirement of these appl" need for Middleware to support easy design & development of such appl".
Popular IOT platform like Google Fit have a Cloud based IOT Middleware for Smartphones.
Service oriented middleware like GSN are proposed for processg data in distributed Environment.
In femtocloud System, the Mobile devices in edge can be configured to provide the Service to requesting devices.
Process on one or more devices where a data Stream generated on each device is processed on itself by Nakumera

	Devices	Security	Mobility Support	Context awareness	Data Analysis	optimized selection of devices
femto cloud.	Mobile	N	N	Y	Y	Y
Nakmore et al	Mobile & Sensor	N	N	N	Y	N
Aazam	Fog, MEC, cloud	Y	N	N	Y	Y
Bonomi	Fog, cloud.	N	Y	N	Y	Y
verbelan	Mobile Cloudlet	N	N	N	N	Y
cloudware	Cloudlet	N	N	Y	N	X
Piro	Cloud.	Y	Y	Y	Y	X

⇒ **SYSTEM MODEL**

FEA includes devices that classified into 5 types :-

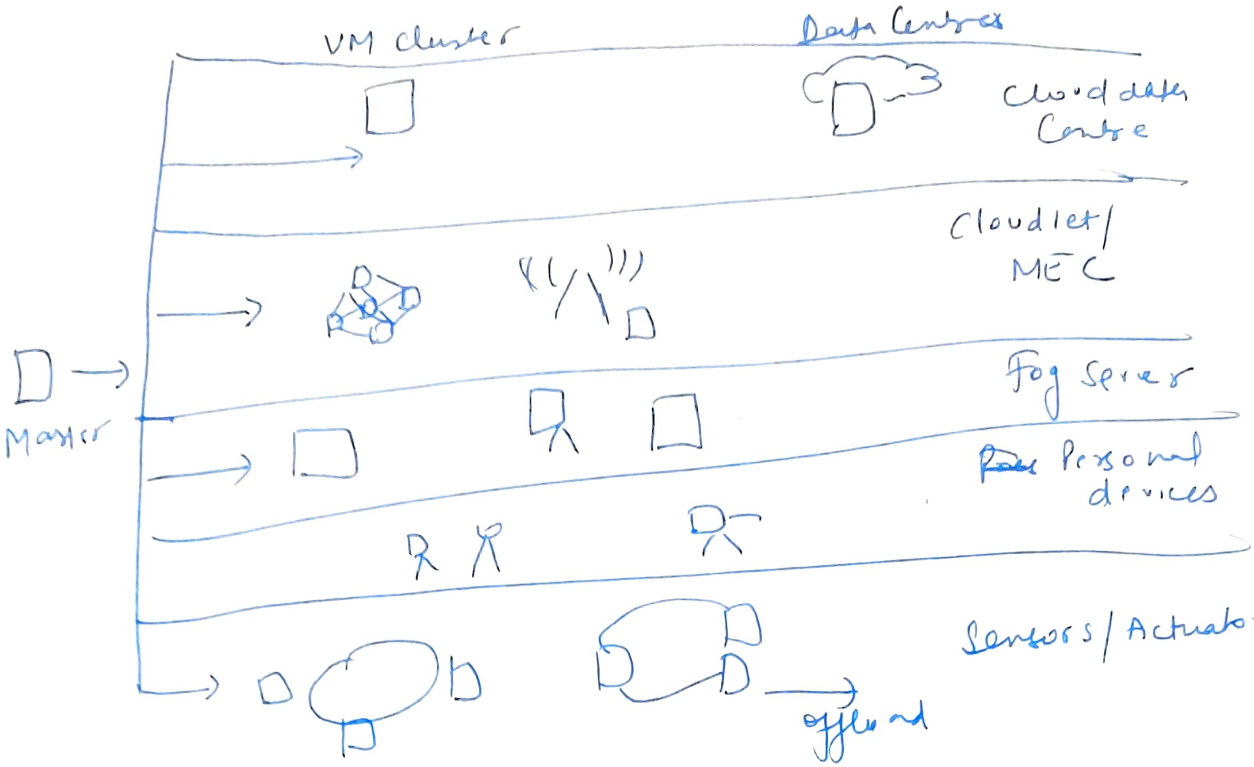
(1) Embeded Sensor / Actuators —
They are installed in physical structures or deployed on human body. Sensors are responsible for obtaining the Env. signals processed by body. while Actuators executes the action initiated by system.

(2) Personal devices —
It demonstrate mobility as they are owned by users. It act as an intermediate data hub or computing platform & provides comm. link to services.

(3) Fog Servers — It is more powerful than personal mobile devices. It provides cheaper option for offload virt. comm. costs. It used to process data.

(4) Cloudlets — It has high Bandwidth Internet Connectivity
They are Mobile Edge Computing Servers

(5) Cloud Server — It is associated with pay as you go Model. It can easily scale the no of VMs as to meet



Fog & Edge Computing devices.

⇒ PROPOSED ARCHITECTURE

Fog & Edge Computing appl^n include the following.

(i) Batch processing that needs large scale data acquisition

(ii) Quick response appl^n that needs response in real time

(iii) Stream appl^n that require processing of continuous data stream in real time.

Due to huge processing requirement, appl^n need a large distributed architecture that processes data in Multiple tiers. Lower tier performs filtering, preprocessing & Extraction. while Edge & fog servers used for processing & Analytics.

(1) API Code - Services designed as an API. It is then integrated into app.

(2) Security -

(3) Authentication -

(4) Privacy - fog Node provides Encryption Capabilities

(5) Device Discovery - It allow the new devices to participate & leave the N/w as they are available in N/w.

(6) Middleware

→ Context Monitoring & prediction -
It can adapt to dynamic changes in user Env.

→ Selection of participating devices -
It can process the data

→ Data Analytics - Some of the analytics task can be used to extract the essential info from raw data. Obtained on user devices.

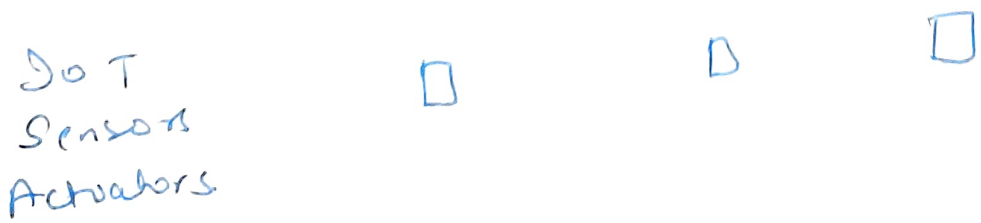
→ Scheduling & Resource Mgmt - Monitors the Incoming task & their assignment.

→ N/w Mgmt.

→ Execution Mgmt - It facilitates appl^n Specific code.

(7) Sensors / Actuators -

Applications	Health Care	Transport	Safety & rescue	
API	API Code	Security Authentication Privacy Encryption	Device Discovery	
Middleware Services	Context Monitoring & Prediction	Execution Mgmt Data Analytics	Scheduling & Resource Mgmt N/w Mgmt	Mobility Mgmt



CASE STUDY EXAMPLE

This is a Mobile appl" that performs real time tracking & Perpetrators through Video Surveillance using Surrogate Mobile phones available in the vicinity.

→ Device Discovery – one of the devices Initiates the Perpetrator tracking appl" by Sending a request on the publish Subscribe channel. Participating devices respond to request & comm. channel is established

→ Context Monitoring – Location is the main context that is required using GPS data on device.

→ Data Analytics – Instead of Sending all the Image data from Mobile devices for recognition Only Images that have faces is Sent.

→ Mobility Support – As perpetrator moves from one location to another Set of devices Select to run the appl" Change.. Moving devices are not ~~to~~ ~~be~~ to be used.

→ N/w Mgmt.

→ Execution Mgmt

→ Scheduling – GPS location of Perpetrator changes as the Preprator is on move.

→ Security Authentication of new device is performed on fog server. Data Encryption is performed.