IOT architecture

1) ONE M2M :

→ It's about networking the machines & devices that pervade our every day lives.

→ One M2M is the global standards initiative that covers requirements, architecture, API specifications, security solutions & interoperability for machine-to-machine & IOT technologies.

→ One M2M specifications provide a framework to support apps & services such as the smart grid, connected car, home automation, public safety & health

2) IOT World Forum Reference Model -

There are 7 layers of this world forum reference model:

Collaboration & Processes ——

Application ——

Data Abstraction ——

Data Accumulation ——

Edge Computing ——

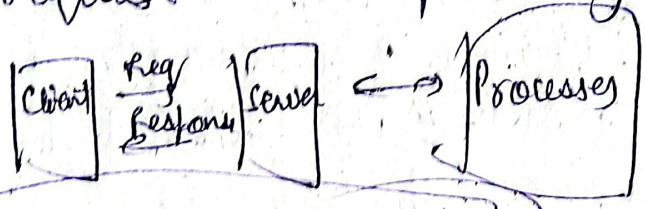
Connectivity ——

Physical Devices & Controllers ——

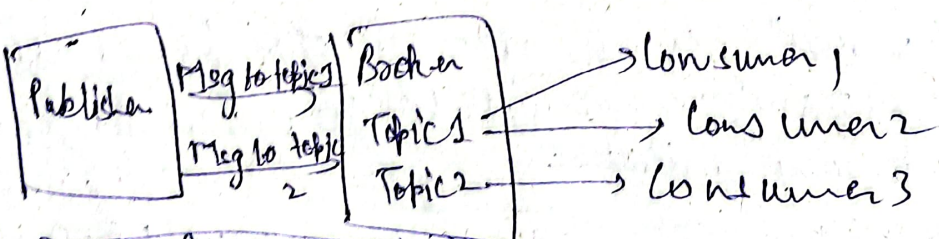
- Physical Device & Controllers → The model calls this layer the "things" of Internet of Things. "Things" are the sensors & devices that are directly managed by IOT architecture.
- Connectivity → This layer spans from the "middle of an edge node device upto transport to the cloud.
- Edge Computing → Required to some degree in any IOT system. This layer interfaces the data & control plains to the higher layer of cloud or entreprises nlw layer.
- Data Accumula" → Given the vol-, vel. & variety that a system can provide, its imp. to provide data storage for processing, integration etc. This layer uses intermediate storage for incoming storage & outgoing traffic.
- Data Abstraction — In this layer, we make 'sense' of the data, collections 'like' info. from multiple IOT sensors or measurements.
- Appⁿ Layer → This is where control plane & data plane logic is executed, monitoring, process optimisa" etc.
- Collaboraⁿ & Processes → Applica" processing is presented to users & data processed at lower layers is integrated into business applica".

Communication Models :-

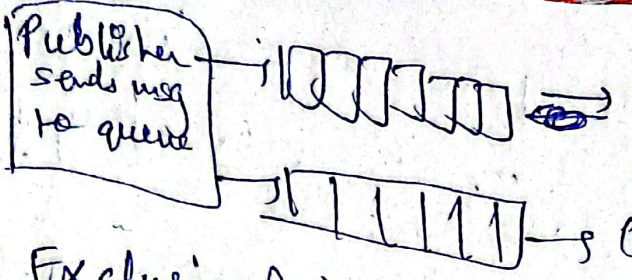
1) Request-Response Model → Client is the IOT device that sends a request to the server.
   - → The request may be for transfer or upload of data.
   - → The server may be remote or local.
   - → The server can handle requests of multiple clients.
   - → This model is stateless & hence each request is independently handled.



2) Publish Subscribe Model → There are 3 entities Publisher, Server & Consumer.
   - → Publisher sends the data to the brokers on topics managed by the brokers.
   - → Consumers subscribe to topics & brokers send the data on topics to consumers.



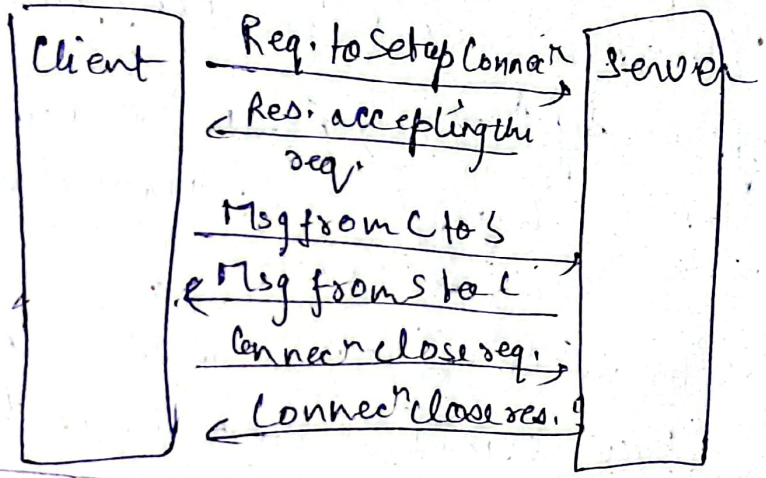
3) Push-Pull Model → Data producers push data to queues & consumer pull data from queues.
   - → Producers & consumers are not aware of each other.
   - → Queue acts as buffer & are useful when producers produce data at which is faster than the rate at which consumers can download



4) Exclusive Pair Model → It is a bidirectional, full duplex comm^n model that uses a persistent connect b/w client & server.

→ The comm^n is persistent & remains open until client sends a request to close the connection.

→ It is stateful connect model. & server is aware of all open connections.

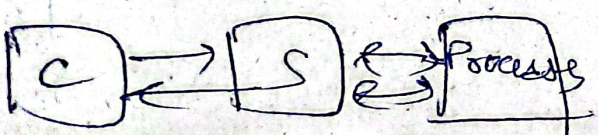


IOT Communica^n APIs :-

API → set of f^ns, protocols, routines & tools used for building app^n s/w.

1) Rest based Communica^n API →

→ Representa^n al State Transfer is a set of architectural principles by which u can design web services & web APIs that focus on system resources & how system resource states are addressed & transferred.



→ follow request response comm^n model.
→ REST architectural constraints apply to the components :-

1) Client-Server → principle behind client-server constraint is the separation of concerns.

2) Stateless → each request is handled independently.

3) Cacheable → data within a response to a request be implicitly or explicitly labelled as cacheable (right to reuse) or non-cacheable.

4) Layered System → Each component can't see beyond the intermediate layers during interact.

5) Uniform Interface → method of comm^n b/w client & server must be uniform.

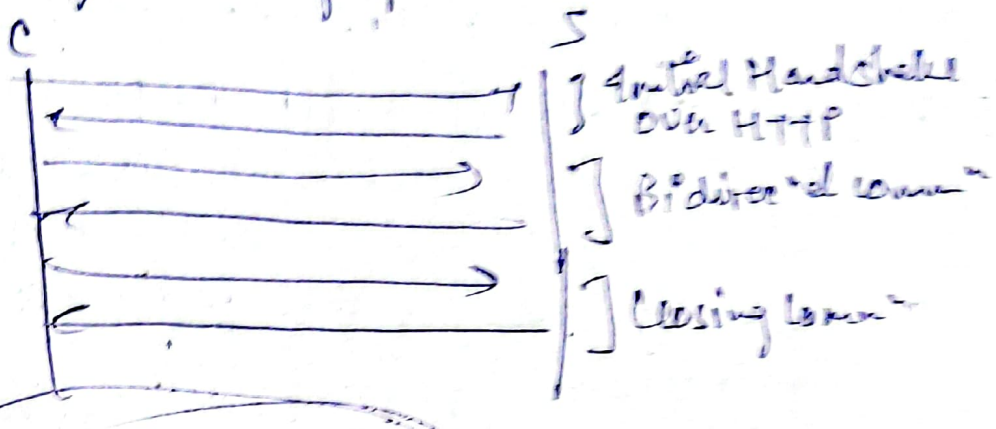
6) Code on Demand → optional
→ servers can provide executable code for clients to execute in their context.

(2) Web Socket Based Comm^n API :-

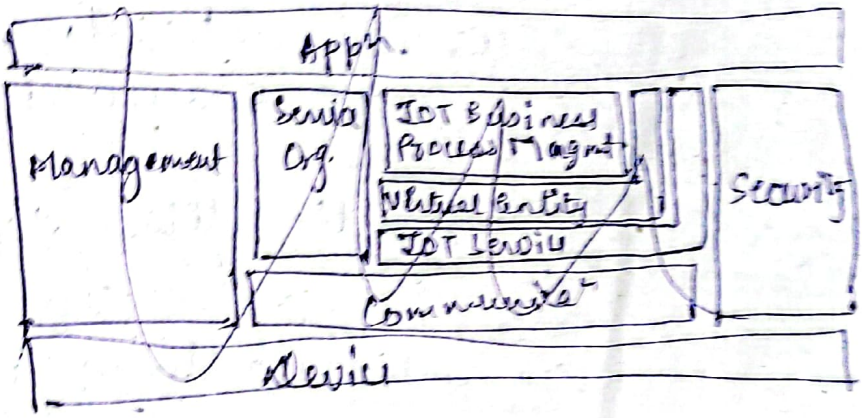
→ They allow bidirectional, full duplex comm^n b/w client & server.
→ follow exclusive pair comm^n model.
→ Unlike REST (Req-Res), these allow full duplex comm^n & do not require a new connect^n setup for each msg · eg/

→ Reduce N/w traffic & latency as there is no overhead for connection setup & teardown time.

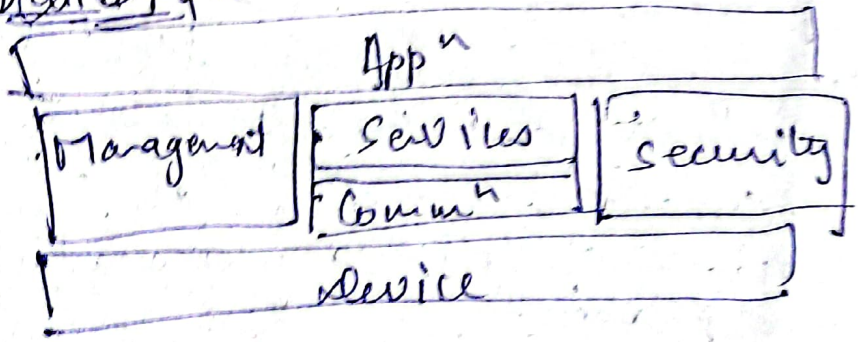
→ Suitable for app's having low latency & high throughput requirements.



IOT Functional Model → APIs at describing mainly the functional grp (FG) & their interface of the "API".



D Diagram -



Device Pg → contains all possible functions hosted by the physical devices that are used for interacting the physical entities.
→ includes sensing, actuation, processing, storage & identification components.

Comm^n Pg → Abstracts all the possible comm^n mechanisms used by the relevant devices in an actual system in order to transfer info. to other devices.

IOT service Pg → ~~mainly~~ corresponds mainly to the service class ~~from the top~~ from the IOT domain model & contain single IOT services exposed by resources hosted on devices or in the ~~internet~~ nlw.
→ handles all the services of the IOT system

Management Pg → functionalities that allow
→ smooth integration of IOT related services with the business process.
→ enabling fault & performance monitoring of the system.
→ configuration for enabling flexibility to changing user demands.

Security Pg → functions that ensure the secure operation of system as well as management of privacy.
→ contains components for authentication, authorisation etc.

Protocols for IOT

App^n Layer → HTTP, CoAP, Websockets, MQTT, XMPP, DDS, AMQP

Transport Layer → TCP, UDP

N/w Layer → IPV4, IPV6, 6LOWPAN

Link Layer → 802.3 - Ethernet
802.13 - Wimax
802.11 - Wifi
802.15.4 - LR-WPAN

1. HTTP → Hyper Text Transfer Protocol
   - → protocol for transferring hypermedia doc such as HTML.
   - → method for encoding & transporting info. b/w web browser & web server
   - → classical client server model.
   - → stateless protocol
   - → generally use TCP connections to communicate with servers.

2. CoAP → Constrained App^n Protocol
   - → enables devices to connect over the internet.
   - → used for constrained devices such as 8-bit microcontroller, low power sensors that can't run on HTTP.
   - → simplification of HTTP protocol
   - → running on UDP instead of ~~HTTP~~ TCP, that helps save bandwidth.

③ Web socket → low level web friendly comm mechanism (FB, messenger etc).
→ full duplex comm over a single socket connection for sending msgs b/w client & server.
→ Client can be web browser, IOT device or mobile app.
→ TCP based.

④ MQTT → Message Queuing Telemetry Transport
→ Publish-subscriber model based
→ runs over TCP/IP
Publisher → Server
Customer/Subscriber → client
MQTT → broker

⑤ XMPP → Extensible Messaging & Presence Protocol
→ protocol for message-oriented middleware based on XML.
→ suitable for voice/video calls, chats, messaging, gaming, multiparty chat, IOT apps such as smart grid & social networking services etc.

⑥ DDS → Data Distribution Service
→ IOT protocol developed for M2M comm by object Mgmt. Grp. comm.
→ enables data exchange via publish-subscriber methodology.
→ integrates the components of a system together providing low latency data connectivity.
→ extreme reliability & a scalable architecture

(1) AMQP → Advanced Msg Queuing Protocol.
→ open std. appn layer protocol for msg. oriented middleware.
→ Msg. pushed by broker or pulled by customer.
→ 3 additives & Binding → states the connec^n b/w M&E exchange.

Exchange → Receives msgs from publisher & routes them to msg. queues.

Message queue → Stores messages until completely processed by client s/w.

(2) TCP → Transmission Control Protocol
→ connec^n Oriented
→ works with IP which defines how comps. sends pkt. of data to each other
→ helps in exchange of msgs. b/w computing devices in a nw.
eg. FTP, SMTP

(2) UDP → User Datagram Protocol
→ connec^n less like DNS
→ unreliable & connec^n less protocol, so there is need to establish connec^n prior to data transfer.
→ Suitable for multicasting.

IPV4	IPV6
defines an IP address as a 32-bit no.	128-bit no.
2^{32} addresses	2^{128} adresses.
numeric address separated by dot.	Alphanumeric address separated by colon.
12.244.135.165	3001:1ab6:0000:0000

6 LOWPAN → IPV6 over Low-Power Personal Area N/w

→ allows smallest device with ltd. processing ability to transmit info wirelessly using IP

→ can connect with IP n/w like WiFi

~~Link~~

① 802.3 ethernet → used primarily in LAN

→ 1st standardised in 1980 by IEEE 802.3 std.

→ 2 categories.

- classic (original)
- Switches (uses switches)

② 802.11 - wifi

→ defines an interface b/w 2 wireless clients.

→ provide secure end-to-end comm^{n} for WLANs.

→ uses various diff frequencies.

3) 802.16 WiMAX → std. for wireless MAN
- specialised in pt. to multipt broadband wireless access.

4) 802-15.4-LR-WPAN - Low Rate Wireless PAN
→ focuses on low cost, low speed comm.
→ achieves extremely low cost manufactg, oper cost & technological simplicity without sacrificing flexibility.

5) 2g/3g/4g Mobile connec -
→ dif. types of telecomm gens.
→ IOT devices based on these stds. can communicate over ~~these~~ cellular n/w.

Components of IOT

1) Sensors/Devices → hardware device that takes i/p from environment & gives it to the system (gateway) by converting it.

Temp → Thermometer → electrical signal → System

Actuator → reverse of sensors.
- converts electrical signals into physical events.
→ takes i/p from system & gives o/p to environment.

(2) Gateway → configured to perform preprocessing of the collected data from lots of sensors locally before transmitting it to next layer (cloud).

→ acts as middleware b/w devices & cloud to protect system from malicious attacks & unauthorised access.

(3) Cloud & Analytics →

→ cloud offers tools to collect, process, manage & store huge amt. of data created by devices, app's & users

→ Analytics is converting analog data from sensors into useful insight ~~which can be later~~ for detailed analysis.

(4) ~~Output~~ User Interface

→ UI should be well designed so that users can perform min. efforts to operate the IOT devices throughout

→ Like multicolor touch panels have replaced hardswitches in our household appliances.

Smart objects

→ when u r going to connect un physical thing with internet, it will become smart.

→ A smart object as defined by IPSO (Internet protocol for smart objects) as.

-) An intelligent (RFID) tag.
-) A sensor
-) An actuator
-) An embedded device
-) any combiner of the above features to form a more complex entity.

Commⁿ pattern for smart objects

1) Range → how far the signal need to be propagated.

- SShortRange → ~~Wired~~ ~~Serial cable~~.
  10 meter max.
  Bluetooth etc.
- Medium → 10s to 100s m.
  WiFi, WPAN, Ethernet.
- Long → > 1 mile
  optical fibre, broadband power line commⁿ.

② Frequency bands –

- Licensed spectrum → IOT long range access technology deployed by broad public service provider.
  - eg. WiMAX
- Unlicensed → short range devices → no guarantee or protection
  - eg → Bluetooth.

③ Power Consumption –

- Powered Nodes → direct connection to power source.
- Battery Powered → provide much more flexibility to IOT devices.

④ Topology: Peer to Peer Topology allows any IOT device to communicate with any other device.

⑤ Constrained Devices → Constrained nodes have ltd. resources that impact their networking feature set & capabilities.

⑥ Constrained Node Nlws → low power, lossy Nlws (LLNs)
eg. LR-WPAN (802.15.4)