Unit - IV

Definition of IOT:

A dynamic global network infrastructure with self configuring capabilities based on standard comm protocols where phy & virtual things have identities, phy attributes, use intelligent interfaces & are integrated seamlessly into the info network, often comm data associated with users & their environments.

Characteristics:

1. Dynamic & self-adapting:

→ IOT devices and sys have the ability to dynamically adapt with changing contexts & take actions based on their operating conditions, user's context or sensed environment.

Ex: consider a surveillance sys with a no. of surveillance cameras. The cameras can adapt their modes based on whether it is day or night. Cameras could switch from lower resolution to higher resolution modes when any motion is detected & alert nearby cameras to do the same. In this ex, the surv sys is adapting itself based on context & changing

2. Self-Configuring:

→ IOT devices may have self-configuring capability, allowing large no. of devices to work together to provide certain functionality.

Ex: weather monitoring. They have the ability config themselves and fetch latest software upgrades with min manual.

3. Interoperable Comm protocols:

→ IOT devices may support a no. of interoperable comm protocols.
→ They can comm with other devices & also with the infrastructure.

4. Unique identity:

→ each IOT device has a unique identity & an unique identifier
→ such as IP address / URI.
→ IOT device interfaces allow monitoring their status, control them remotely, manage their infrastructure & configuration.

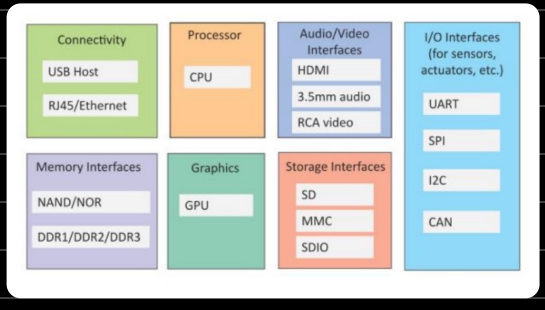
5. Integrated into info network:

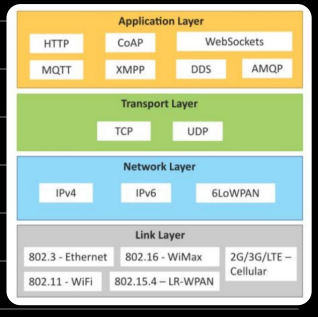
→ IOT devices are integrated into the info network that allows them to comm and exchange data with other devices & systems.
→ integration into info network helps in making IOT systems "smarter" due to collective intelligence of the individual devices in collaboration with the infrastructure.

Ex: Weather monitoring node can comm & exchange data with another connected node. Data from a large no. of connected nodes can be used to analyze & predict the weather.

Physical design of IOT: → interfaces → protocols

→ IOT device may consist of several interfaces for connections to other devices
→ can be wired / wireless.
① memory interfaces
② storage
③ graphics
④ Connectivity
⑤ audio / video interfaces.
⑥ I/O interfaces for sensors, actuators etc.





→ An IOT device can collect data from attached sensors such as temp, humidity etc.
→ The sensed data can be communicated to other devices / cloud-servers.
→ IOT devices can be conn to actuators ex: relay switches.

IOT Protocols:

→ Link layer: determines how the data is physically sent over network (via copper wire, other hardware devices) using link layer protocols.

Link layer Protocols:

1. Ethernet: 802.3 standard. Collec of wired cables
2. WiFi: 802.11 standard. Collec of wireless L.A.N (WLANS).
3. WiMax: 802.16 standard. Collec of ~ broadband standards.
4. LR-WPAN: 802.15.4 standard. Collec of low-rate wireless personal area networks
5. 2G / 3G / 4G - mobile comm: collec of diff generations of mobile comm standards.

→ **Network Layer**: resp to send datagrams from source to destination network.

Protocols:

1. IPV4 : Internet protocol version 4. Uses 32-bit add scheme. most common.
2. IPV6 : Int protocol version 6. Uses 128-bit add scheme. newest version of I.P.
3. 6LoWPAN : IPV6 over low-power devices which have limited processing capability

→ **Transport Layer**: provide end-to-end message transfer capability independent of underlying network. provides error control, flow control, segmentation.

⇒ protocols : transfer control protocol ↑ user datagram protocol. ↑

T.C.P	U.D.P
→ reliable, connec are verified	→ unreliable, connec are un-verified.
→ connection oriented protocol	→ connectionless protocol.
→ speed is lower	→ speed is higher.
→ data packets are arranged in order.	→ data packets are unordered.
→ transfers segments.	→ transfers datagrams.
→ provides feedback / ack.	→ no feedback / ack.
→ has to establish a connec b4 data transfer.	→ detects errors but does not specify which error.
→ used for web browsing, mailing, file transfers.	→ no need to establish a connec b4 data transfer.
→ used for reliable but slow transmission.	→ used in games, video conferencing.
	→ used for only its speed, reliability does not matter.

→ **Application Layer**: these protocols define how the applications interface with lower layer protocols to send data over the network.

→ used by network applications that use internet ex: web browsers.
→ allows diff ways to enable any user to access the network with ease.

1. HTTP : * Hypertext transfer protocol.
* forms foundation of world wide web www.
* client - server protocol.
* comm b/w web browsers & web servers

2. CoAP : * Constrained application protocol.
* for machine to machine applications (M2M).
* for constrained environments & devices.
* designed easily to interface with HTTP.
* client - server architecture.

3. WebSocket : * allows full duplex comm over single socket connection for sending messages b/w client & server.
* allows streams of messages to be sent back & forth b/w client & server.
* Based on TCP.

4. MQTT : * message queue telemetry transport.
* light - weight messaging protocol.
* client - server architecture.
* suited for devices with less processing power & less memory.

5. XMPP : * extensible messaging & presence protocol.
* used for real - time comm.
* allows sending small chunks of XML data from one network to another in real - time.
* client - server arch.

6. DDS : * data distribution service.
* for device - to - device / machine to machine comm.
* provides (QoS) quality of service.
* uses publish - subscribe model (publishers create topics that the subscribers can subscribe.

7. AMQP : * advanced message queuing protocol.
* open application layer protocol for business messaging
* publish - subscribe model.

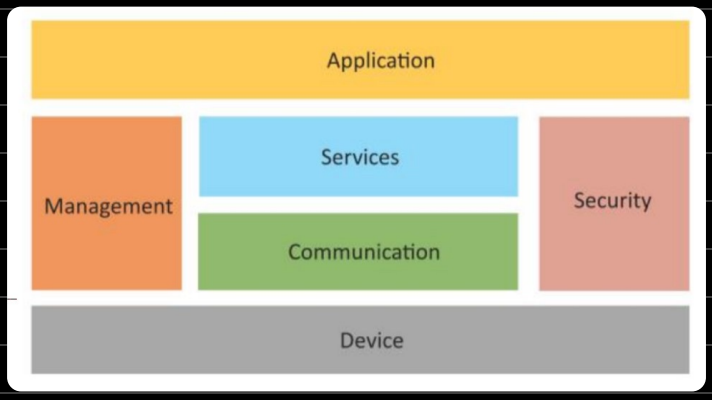
Logical Design of IoT:

→ func^c blocks
→ comm models
→ comm APIs

→ refers to an abstract representation of the entities & processes without going into low-level specifics of the implementation.

→ IoT sys comprises a no. of functional blocks that provide the sys the capabilities for identification, sensing, actuation, comm and management.

→ functional Blocks of IoT:

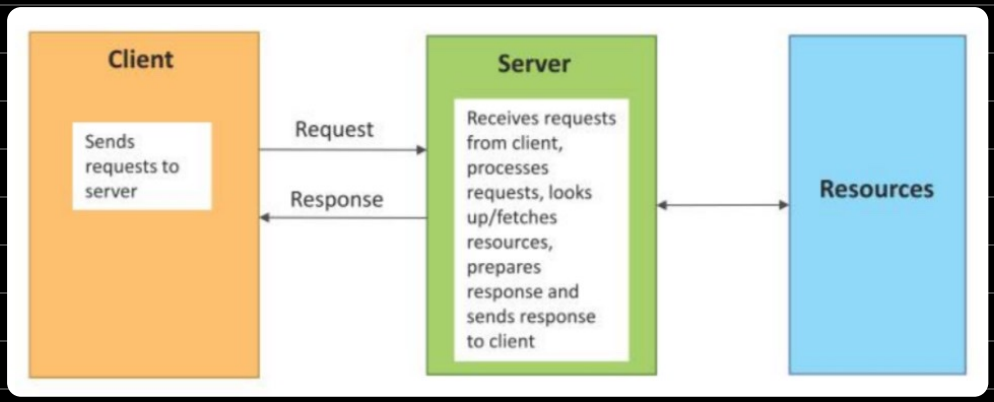


1. device: IoT sys comprises of devices that provide sensing, monitoring and actuation.
2. comm: comm block handles comm for IoT sys.
3. Services: various IoT services such as device monitoring, control services and device discovery services.
4. Management: Block provides various func^n to govern the IoT sys.
5. Security: secures the IoT sys by authentication, authorization, integrity etc
6. Application: IoT apps provides an interface that the users can use to control & monitor various aspects of IoT system. eg: view system status, analyze processed data etc

→ IoT comm Models:

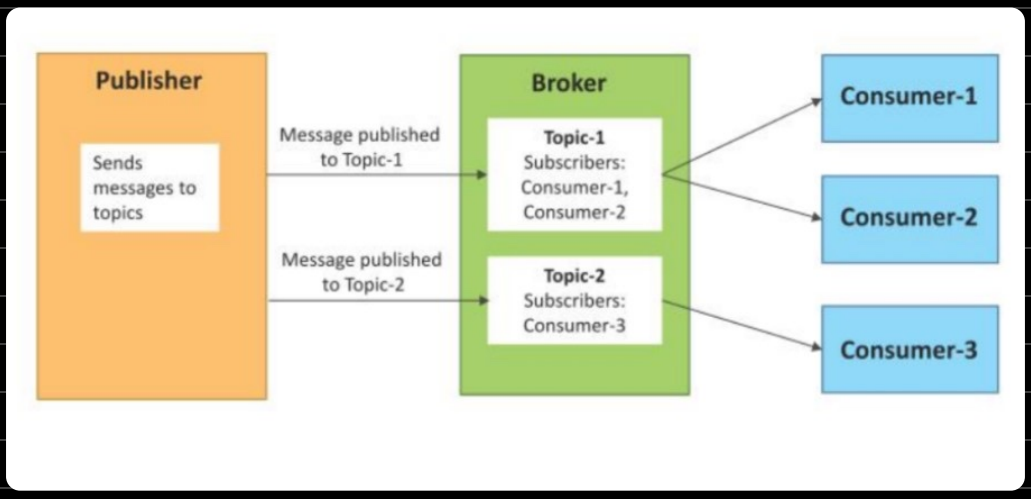
1. Request-response model:

* Client sends reqs to the server & the server responds to the req
* When the server receives a req, it decides how to respond, fetches the data, prepares response & sends the response to the client.



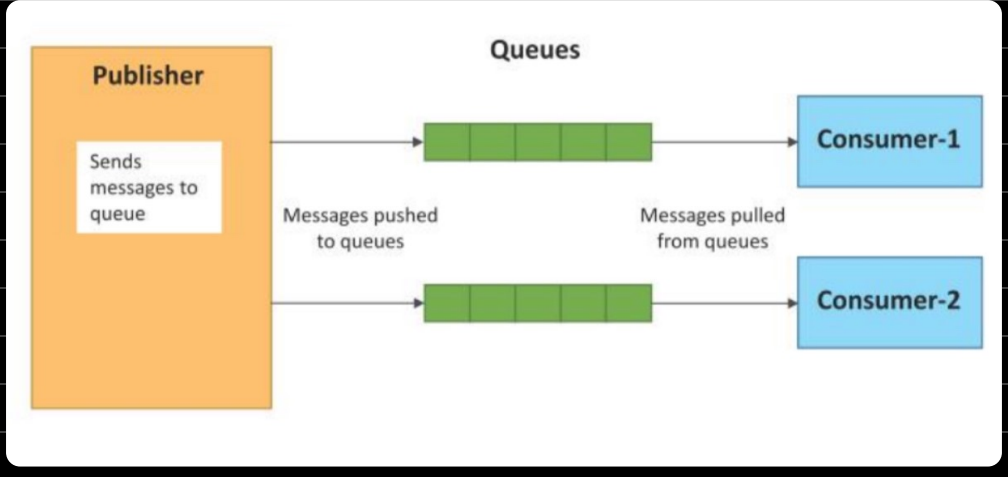
2. Publish-subscribe comm model:

→ involves publishers, subscribers, brokers, consumers.
→ publishers are source of data. They send data topics which are managed by the broker.
→ publishers are not aware of the consumers.
→ Consumers subscribe to the topics which are managed by the broker.
→ when the broker receives data for a topic from the publisher, it sends the data to all the subscribed consumers.



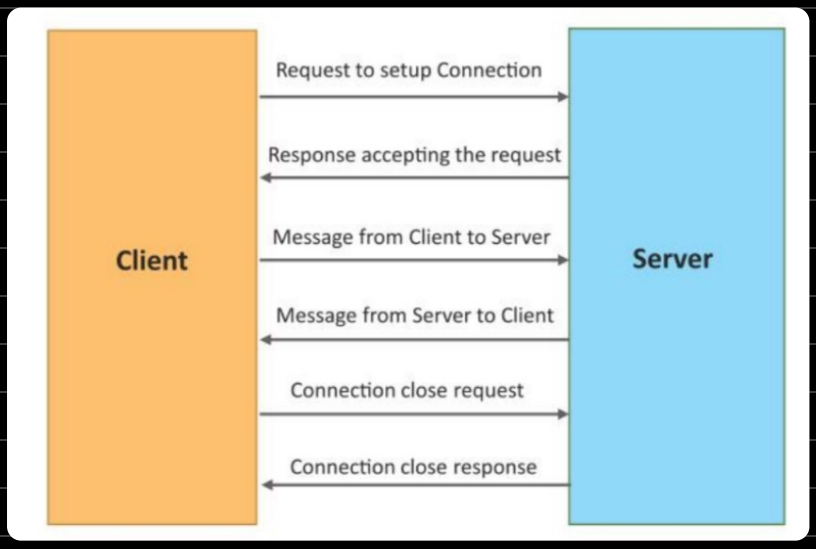
3. Push-pull comm model:

→ data producers push the data to queues & consumers pull the data from the queues.
→ producers do not need to be aware of consumers.
→ queues act as buffers and help in decoupling the messages b/w the producers & consumers.
→ buffers help to synchronize b/w producers & consumers by maintaining pushing rate & pulling rate



4. Exclusive Pair comm model:

→ Bi-directional, fully duplex comm model.
→ persistent connection b/w client & server.
→ Once comm is setup, it remains open until the client sends a req to close it.
→ client & server can start sending/receiving message once the comm is setup.

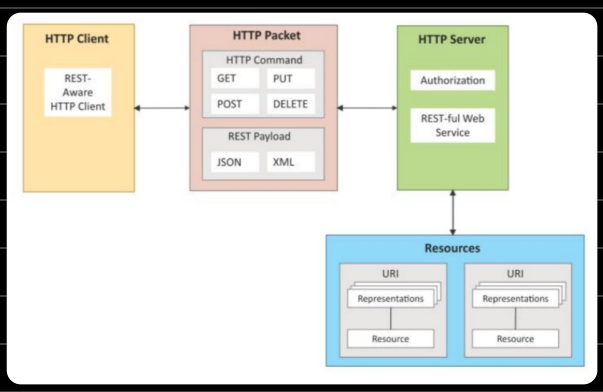


→ IoT comm APIs:

1. REST - based comm APIs:

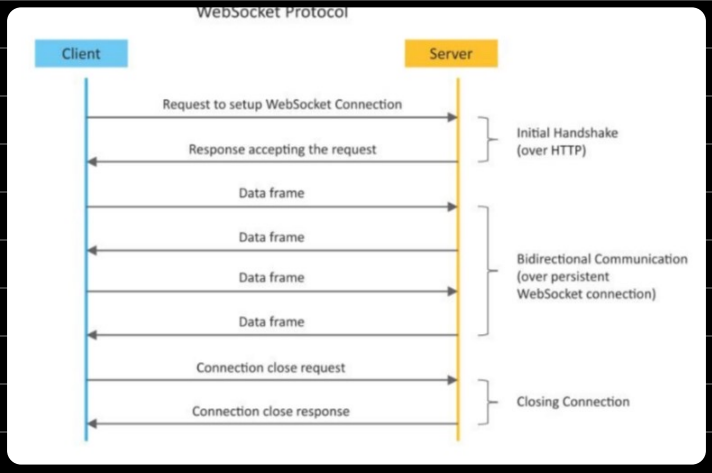
* representational state transfer.
* it is a set of architectural principles by which you can design web services & web APIs that focus on a system's resources and how resource states are addressed & transferred.
* follows req-response comm model.
* The REST architectural constraints apply to the components, connectors, data elements.
* The constraints are:

1. client-server
2. stateless
3. cache-able
4. layered system
5. uniform interface.
6. code on demand.



2. WebSocket - based comm:

* WebSocket APIs allows bi-direc, full duplex comm b/w clients & servers.
* they follow exclusive pair comm model.



IOT enabling Technologies:

1. Wireless sensor networks:

* a wireless sensor network comprises of devices with sensors which are used to monitor environmental & phy conditions.
* WSNs used in IOT systems:
  1. weather monitoring systems. (nodes collect temp, humidity often data)
  2. Indoor air quality monitoring systems. (use WSNs to collect data on indoor air quality & conc of gases)
  3. Soil - moisture monitoring system (use WSNs to monitor soil mois at diff locations).
  4. Surveillance systems (use WSNs for collecting surveillance data eg: motion detection data)
  5. Smart grids (use WSNs for mon grid at diff points)
  6. Structural health mon systems (mon health of structures by coll various data at diff points in the struc)
* Self-organizing networks.
* has large no. of nodes, manual config of nodes is not possible.
* robust network.
* network can re-config itself.

2. Cloud Computing:

* It is a computing method to deliver applications & services over the internet.
* provides storage resource & services to the users.
* Cloud computing resources can be accessed over the network using standard acc mechanisms

* C. Comp services are offered in diff forms:
1. Infrastructure-as-a-service (IaaS)
→ provide users ability to provision computing & storage resources.
2. Platform-as-a-service (PaaS)
→ provide users the ability to develop & deploy applications in cloud.
3. Software-as-a-service (SaaS)
→ provides users a complete software application / user interface to the application itself.

3. Big data Analytics:

→ defined as colles of data sets whose vol, velocity, variety is so large that it is diff to store, manage, process & analyze the data using traditional data bases & tools.
→ B.D.A involves several steps: sorting data, cleaning data, processing data & visualizations.
→ examples of big data: Sensor data, machine sensor data, health & fitness data, GPS data.
→ Chara: of Big data:
1. volume : growing exponentially, diff to store, manage & process.
2. velocity : how fast the data is generated & how frequently it varies. Speed is also increasing.
3. variety : refers to forms of data. Big data comes in diff forms eg: structured, unstructured etc

4. Comm Protocols:

* backbone of IOT Systems.
* allows devices to exchange data over the network.
* allows 1. sequence control (helps in ordering packets, determine lost packets)
2. error control. (helps in controlling rate at which data is sent / buffer is not overwhelmed).
3. flow control (ack is sent back from receiver, resends data)

5. Embedded Systems:

* Emb sys is a comp sys that has comp hardware & software embedded to perform specific tasks.
* designed to perform specific set of tasks.
* key comps: microcontroller / processor.
memory (RAM, ROM, cache)
networking units (ethernet, wifi)
I/P / O/P units (display, keyboard)
storage (flash mem)
* low cost.
* can run RTOS (real time op systems)
* simple V.I
eg: Washing machines, microwaves, digital cameras etc.

IOT levels & deployment:

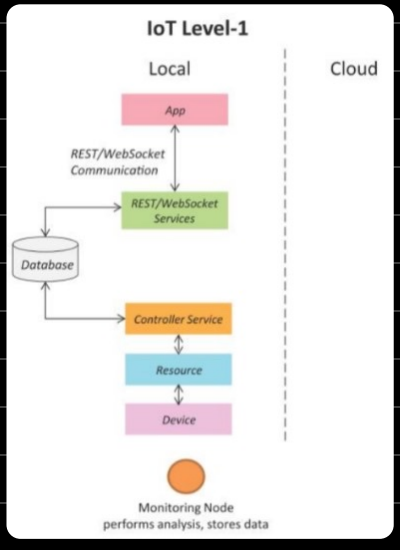
IOT sys comprises of foll comps:

1. device (IOT device allows identification, remote sensing, monitoring capabilities.
2. resource (software components on IOT device that allow network access for the device)
3. Controller service (receives commands from the application for controlling the device)
4. database (either local, cloud & stores data gen by the IOT device)
5. web service (serves as link b/w the IOT device, app, database & other components)
6. analysis component (resp for analyzing the IOT data & generate results)
7. application (provide an interface that the users can use to control and monitor various aspects of the IOT sys, view sys status etc)

IOT levels:

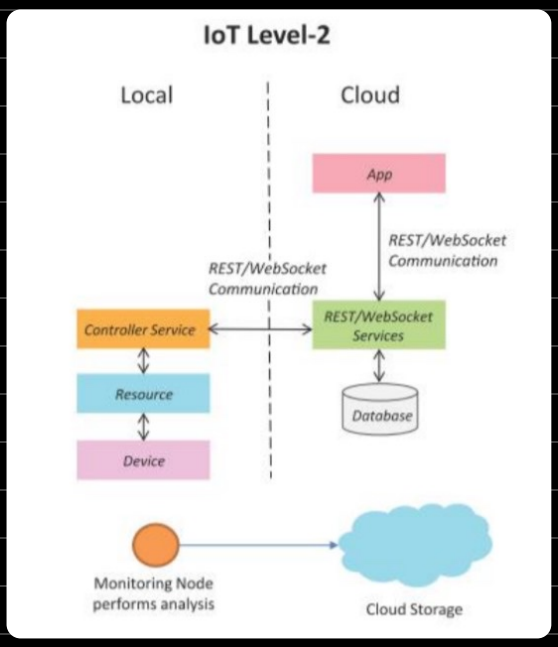
① IOT level-1:

* single node / device that performs sensing, actuation, stores data, performs analysis, hosts the app.
* suitable for modeling low-cost & low-complexity solutions
* suitable for scens where data involved is not big.
* analysis requirements are not computationally intensive.



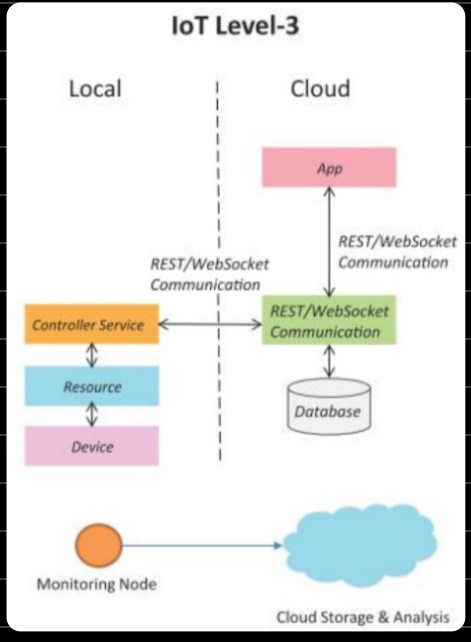
② IOT level-2:

* single node that performs sensing & actuation, analysis.
* data is stored in cloud. Application is cloud based
* data involved is big.
* analysis requirement is not computationally intensive.



③ IOT – level – 3 :

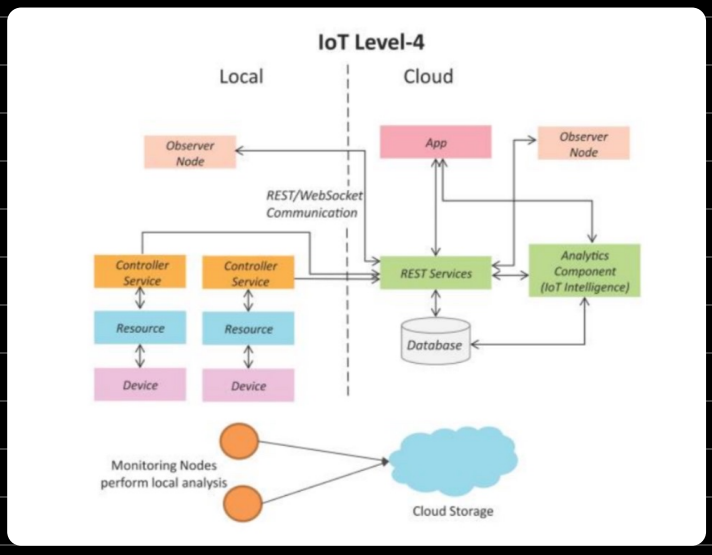
→ has a single node
→ data is stored & analyzed in cloud.
→ application is cloud based.
→ suitable for 80% where data is big.
→ analysis requirements are computationally intensive.



④ IOT level – 4 :

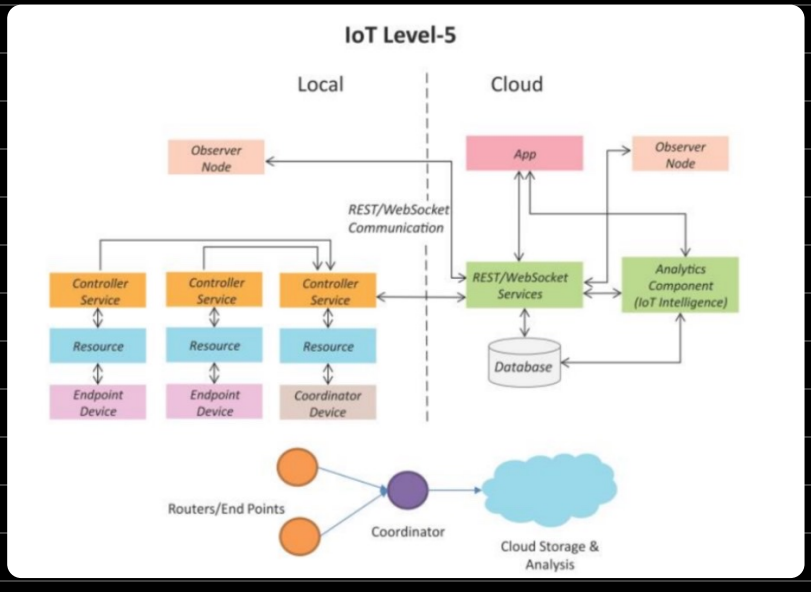
→ has multiple nodes that perform local analysis.
→ data is stored in cloud
→ application is cloud based.
→ contains local & cloud based observer nodes which can subscribe & receive info collected in cloud from IOT devices

→ suitable for solns where multiple nodes are req.
→ data involved is big.
→ Analysis reqs are computationally intensive



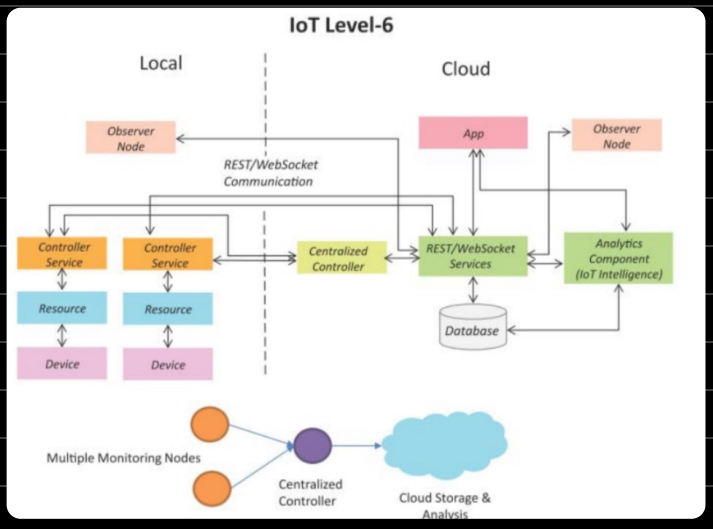
⑤ IoT Level-5:

→ multiple nodes & one coordinator node.
→ end nodes perform sensing & actuation.
→ coordinator node collects data from end nodes & sends it to the cloud.
→ data is stored & analyzed in cloud.
→ application is cloud based.
→ suitable for solns based on wireless sensor networks.
→ data involved is big.
→ analysis reqs are comp intensive.



⑥ IOT level-6:

→ multiple independent end nodes that perform sensing and actuation.
→ data is stored in cloud
→ application is cloud based.
→ data is stored in cloud by the analysis component.
→ results are visualized with cloud based app.
→ Centralized control is aware of all end nodes & sends control commands to the nodes.



IOT in diff domains:

① Home Automation:

* Smart lighting
* smart appliances
* Intrusion detection
* smoke / gas detection

② Cities:

* Smart Parking
* Smart lighting
* Smart roads
* Structural health monitoring
* surveillance
* emergency response.

③ Environment:

* weather monitoring.
* air pollution monitoring
* noise pollution monitoring
* forest fire detection
* river floods detection

④ Energy:

* Smart grids
* renewable energy systems
* Prognostics.

⑤ Retail:

* inventory management.
* Smart payments.
* Smart vending machines.

⑦ Logistics:

* route generation & scheduling
* fleet tracking.
* shipment monitoring
* remote vehicle diagnostics

⑧ Agriculture:

* Smart Irrigation
* green house control

⑨ Industry:

* machine diagnostics & Prognosis.
* indoor air quality monitoring

⑩ Health and Lifestyle:

* health and fitness monitoring
* wearable electronics.