Internet Of Things

Syllabus :

CAT - 1

Definition and Characteristics :

Physical design of IOT :

- IoT Protocols
  - Link Layer:
  - Network/Internet Layer:
  - Transport Layer:
  - Application Layer

Logical design of IoT :

- IoT Functional blocks :
- IoT Communication Model :
- IoT Communication APIs:
  - REST :
  - WebSocket :

IoT Enabling Technologies :

- A Wireless Sensor Network (WSN) :
- Cloud Computing :
- Big data Analytics :

Definition and Characteristics :

A dynamic global network infrastructure with

- self-configuring capabilities based on

- standard and interoperable communication protocols where
  - physical and virtual “things" have identities, physical attributes, and virtual personalities
  - and use intelligent interfaces,
  - and are seamlessly integrated into the information network.
  - often communicate data associated with users and their environments.

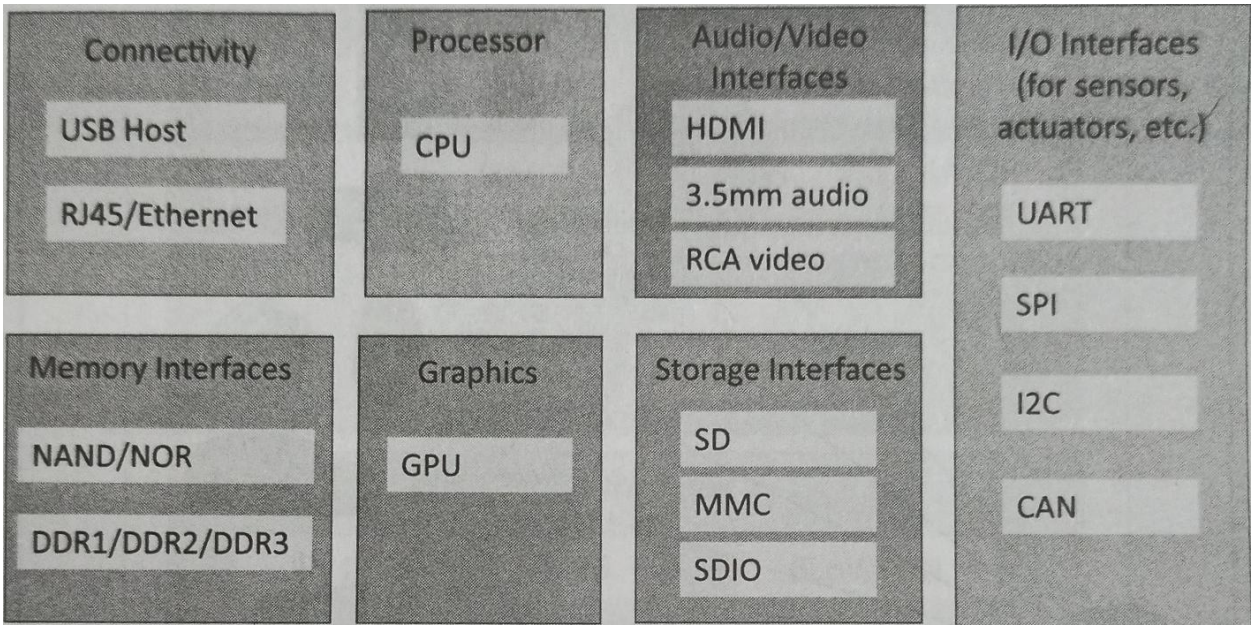
1. **Dynamic & Self - Adapting**

   Capable to adapt dynamically with the changing contexts and take actions based on their operating conditions, user’s context or sensed environment. Eg : CCTV

2. **Self- Configuring**
   - Allows large number of devices to work together to provide certain functionality
   - Ability to configure themselves with IoT infrastructure
   - Setup the networking, and fetch latest software upgrades with minimal manual or
     user intervention.
3. **Interoperable Communication Protocols**
   - Support a number of interoperable communication protocol
   - Communicate with other devices and other IoT infrastructure
4. **Unique Identity**
   - IoT devices – unique identity – IP address / URI
   - IoT systems – intelligent interfaces – adapt to context , user and environment
   - IoT device interface – user query the device, monitor the status, control remotely
5. **Integrated into Information Network**
   - IoT devices dynamically discovered in the network

- Eg: Weather Monitoring

Physical design of IOT :

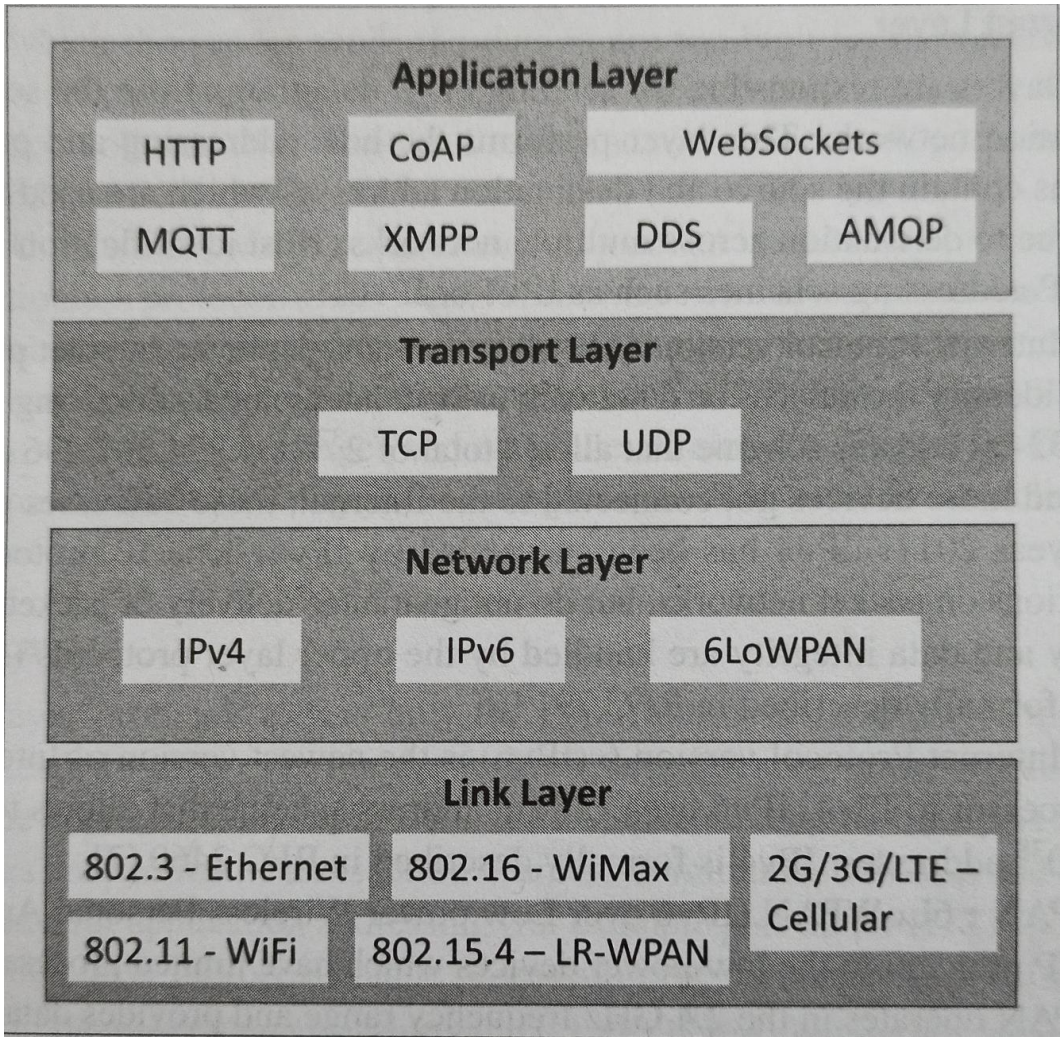


Generic block Diagram of IoT device

IoT Protocols

Link Layer:

- Link layer protocols determine how the data is physically sent over the network's physical
  layer or medium (e.g., copper wire, coaxial cable, or a radio wave).
- Link layer determines how the packets are coded and signaled by the hardware device over the medium to which the host is attached (such as a coaxial cable).
- 802.3 - Ethernet : IEEE 802.3 is a collection of wired Ethernet standards for the link layer



- 802.11 - WiFi : IEEE 802.11 is a collection of wireless local area network (WLAN) communication standards, including extensive description of the link layer.
- For example, 802.1 la operates in the 5 GHz band, 802.1 lb and 802.1 lg operate in the 2.4 GHz band, 802.1 In operates in the 2.4/5 GHz bands, 802.1 lac operates in the 5 GHz band and 802.1 lad operates in the GHz band.
- 802.16 - WiMax : IEEE 802.16 is a collection of wireless broadband standards, including extensive descriptions for the link layer.
- 802.15.4, LR-WPAN IEEE 802.15.4 is a collection of standards for low-rate wireless personal area network (LR-WPANs). These standards form the basis of specifications for high level communication protocols such as ZigBee.

- 2G/ 3G/ 4G - Mobile Communication : There are different generations of mobile communication standards including second generation (2G including GSM and CDMA), third generation (3G - including ÜMTS and CDMA2000) and fourth generation (4G - including LTE), based on these standards can communicate over cellular networks.

Network/Internet Layer:

- The network layers are responsible for sending of IP datagrams from the source network
  to the destination network. This layer performs the host addressing and packet routing.
- Host identification is done using=hierarchical IP addressing schemes such as IPv4 or IPv6.
- IPv4 : Internet Protocol version 4 (IPv4) is the most deployed Internet protocol that is used to identify the devices on a network using a hierarchical addressing scheme , IPv4 uses a 32-bit address scheme that allows total of 2^{32} or 4,294,967,296 addresses.
- As more and more devices got connected to the Internet, these addresses got exhausted in the year 2011. IPv4 has been succeeded by IPv6.
- IPv6 : Internet Protocol version 6 (IPv6) is the newest version of Internet protocol and successor to IPv4. IPv6 uses 128-bit address scheme that allows total of 2^{128} or 3.4 x 1038 addresses.
- 6LoWPAN : 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) brings IP protocol to the low-power devices which have limited processing capability, 6LoWPAN operates in the 2.4 GHz frequency range and provides data transfer rates of 250 Kb/s.

Transport Layer:

- The transport layer protocols provide end-to-end message transfer capability independent

of the underlying network.

- The message transfer capability can be set up on connections, either using handshakes (as in TCP) or without handshakes/acknowledgements (as in UDP). The transport layer provides functions such as error control, segmentation, flow control and congestion control.
- TCP : Transmission Control Protocol (TCP) is the most widely used transport layer protocol, that is used by web browsers (along with HTTP, HTTPS application layer protocols), email programs (SMTP application layer protocol) and file transfer (FIT) TCP is a connection oriented and stateful protocol.
- UDP : Unlike TCP, which requires carrying out an initial setup procedure, UDP is a connectionless protocol UDP is useful for time-sensitive applications that have very small data units to exchange and do not want the overhead of connection setup.

Application Layer

- Application layer protocols define how the applications interface with the lower layer protocols to send the data over the network
- Port numbers are used for application addressing (for example port 80 for HTTP, port 22 for SSH. etc.), Application layer protocols enable process-to-process connections using ports.

1. HTTP : [TCP → RR]

- Hypertext Transfer Protocol (HTTP) is a application layer protocol includes commands such as GET, PUT, POST, DELETE, HEAD, TRACE, OPTIONS, etc.
- The protocol follows a request-response model where a client sends requests to a server using the HTTP commands.
- HTTP protocol uses Universal Resource Identifiers (URIS) to identify HTTP resources. HTTP is described in RFC 2616 (11).

2. CoAP : [UDP → RR]

- Constrained Application Protocol (CoAP) is an application layer protocol for machine-to machine (M2M) applications, meant for constrained environments with constrained devices and constrained networks.

- Like HTTP, CoAP is a web transfer protocol and uses a request-response however it runs on top of UDP instead of TCP.
- CoAP uses a client-server architecture where clients communicate with servers using connectionless datagrams. CoAP is designed to easily interface with HTTP.
- Like HTTP, CoAP supports methods such as GET, PUT, POST, and DELETE.

3. WebSocket :[TCP → Exclusive Pair]

- WebSocket allows full-duplex communication over a single connection messages between client and server.
- WebSocket is based on TCP and allows streams of messages to be back and forth between the client server while keeping the TCP Open.
- Client can be a browser, application or an IoT device.

4. MQTT:

- Message Queue Telemetry Transport (MQTT) is a light-weight messaging protocol based on publish-subscribe.
- MQTT uses a architecture where client (such as an IoT &device)connects to server (also called MQTT Broker) and messages to topics on the server. The broker forwards the messages to the clients subscribed to topics.
- MQTT is well for constrained environment where devices have limited processing and memory resources and the network bandwidth is low.

5. XMPP :

- Extensible Messaging and Presence (XMPP) is a for real-time communication and streaming XML data network entities.
- XMPP powers wide range of applications including messaging. presence, data syndication.
  gaming. multi-party chat and voice/video calls.

- XMPP allows sending small chunks of XML data from one network entity to another in near real-time.
- XMPP is a decentralized and uses a client-server architecture.
- XMPP supports both client-to-server and server-to-server communication paths.

6. DDS :

- Data Distribution Service (DDS) is a data-centric middleware for device-to-device or machine-to-machine communication .
- DDS a publish-subscribe model where publishers (e.g. devices that generate data) create topics to which subscribe , Publisher is responsible data distribution and subscriber is responsible for receiving published data.
- DDS provides quality-of-service (QoS) control and configurable reliability.

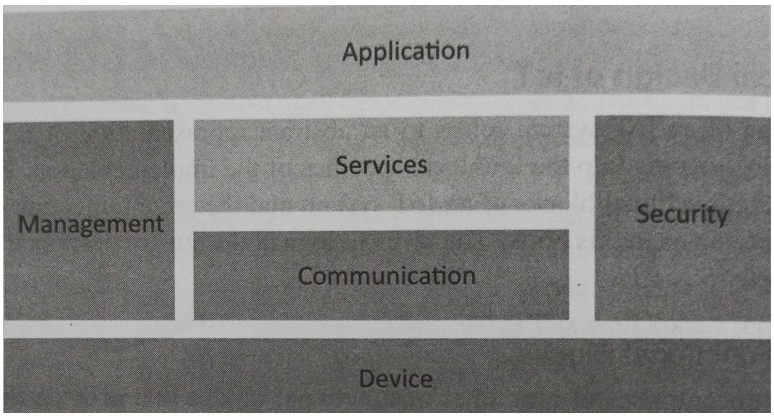
7. AMQP :

- Advanced Message Queuing Protocol (AMQP) is an layer protocol for business messaging.
- AMQP supports both point-to-point and publisher/subscriber models, routing and queuing.
- AMQP brokers receive from publishers (e.g.. devices or applications that generate data) and route them over connections to consumers (applications that process data).
- Publishers publish the messages to exchanges which the distribute message copies to queues.
- Messages are either delivered by the broker or the consumers can pull the messages from the queues.

Logical design of IoT :

IoT Functional blocks :

An IoT system comprises of a number of functional blocks that provide the system the capabilities for identification, sensing, actuation, communication, and management,

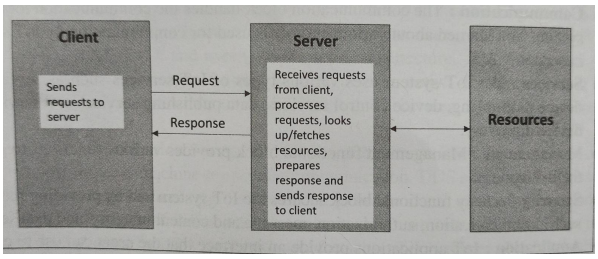


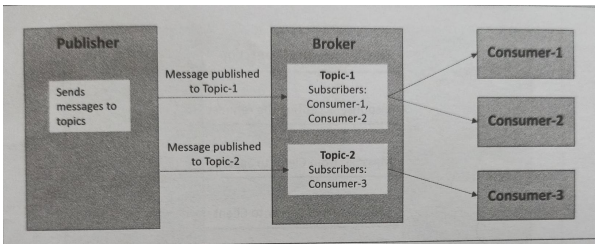
Functional block of IoT

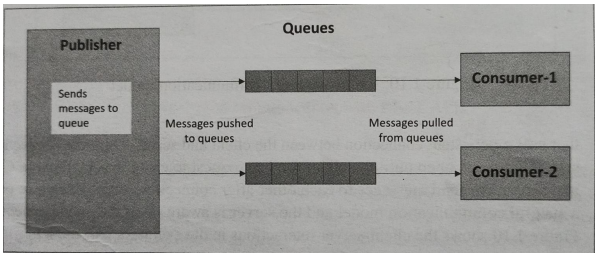
- Device : An IoT system comprises of devices that provide sensing, actuation, monitoring and control functions.
- Communication : The communication block handles the communication for the IoT system.
- Service : An IoT system uses various types of IoT services such as services for device monitoring, device control services, data publishing services and services for device discovery.
- Management : Management functional block provides various functions to govern the IoT system.
- Security: Security functional block secures the IoT system and by providing functions such as authentication, authorization, message and content integrity, and data security.
- Application : IoT applications provide an interface that the users can use to control and monitor various aspects of the IoT system. Applications also allow users to view the system status and view or analyze the processed data.

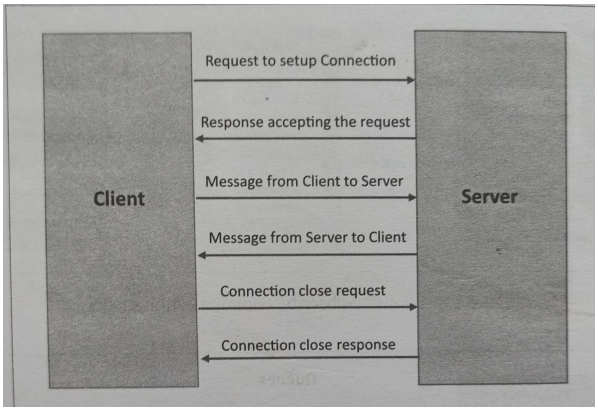
IoT Communication Model :

Request-Response :	Publish-Subscribe :	Push-Pull	Exclusive Pair :
Request-Response is a communication model in which the client sends requests to the server and the server responds to the requests.	Publish-Subscribe is a communication model that involves publishers. brokers and consumers. Publishers are the source of data. Publishers send the data to the 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.	A communication model in which the data producers push the data to queues and the consumers pull the data from the queues.	Exclusive Pair is a bi-directional, fully duplex communication model that uses a persistent connection between the client and server.
When the server receives a request, it decides how to respond, fetches the data, retrieves resource representations, prepares the response, and then sends the response to the client. Request-Response model is a stateless communication model and each request-response pair is independent of others.	When the broker receives data for a topic and , it sends the data to all the subscribed consumer.	Producers do not need to be aware of the consumers. Queues help in decoupling the messaging between the producers and consumers. Queues also act as a buffer which helps in situations when there is a mismatch between the rate at which the producers push data and the rate rate at which the consumers pull data.	Once the connection is setup it remains open until the client sends a request to close the connection. Client and server can send messages to each other after connection setup. Exclusive pair is a stateful communication model and the server is aware of all the open connections.
EG HTTP			eg : WebSocket

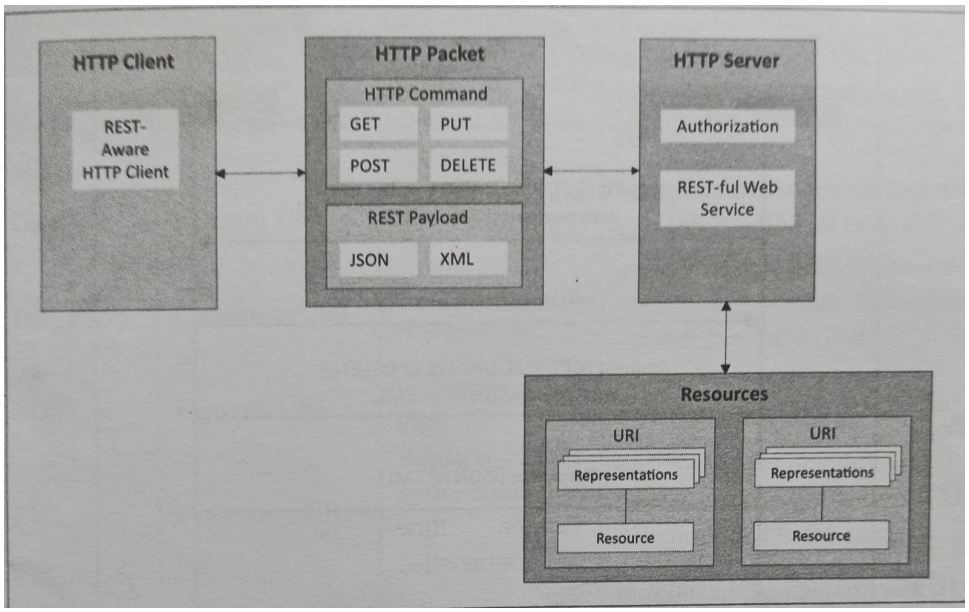






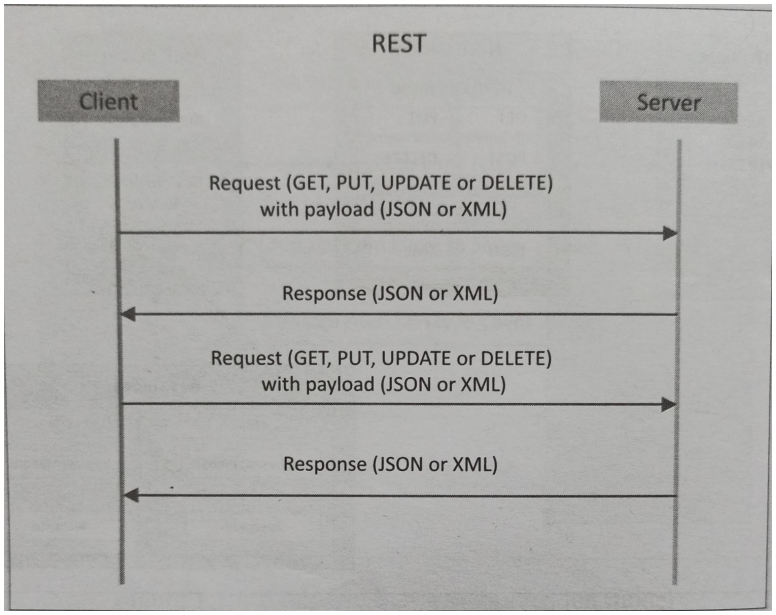


IoT Communication APIs:



REST :

- Representational State Transfer (REST) [88] is a set of architectural principles by which
  you can design web services and web APIs that focus on a system's resources and how
  resource states are addressed and transferred.
- REST APIs follow the request-response communication model.



- The REST architectural constraints apply to the components, connectors, and data elements, within a distributed hypermedia system.
- The REST architectural constraints are as follows:

1. Client-Server:

- The principle behind the client-server constraint is the separation of concerns, For example, clients should not be concerned with the storage of data which is a concern of the server. Similarly, the server should not be concerned the user interface, which is a concern of the client. Separation allows client and server to independently developed and updated.

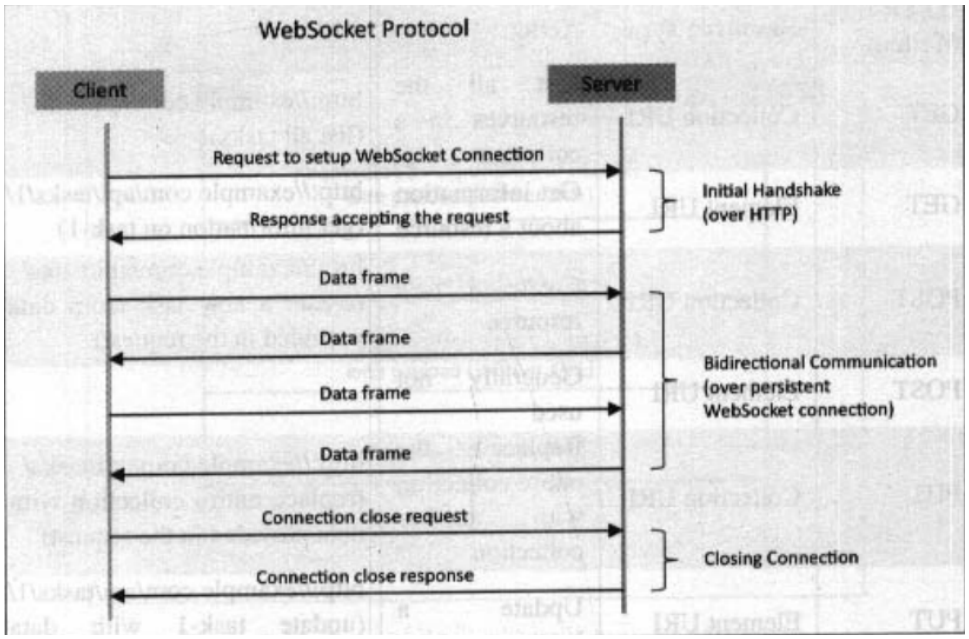
2. Stateless: Each request from client to server must contain all the information necessary to understand the request, and cannot take advantage of any stored context on the server. The session state is kept entirely on the client.
3. Cache-able: Cache constraint requires that the data within a response to a request be implicitly or explicitly labeled as cache-able or non-cache-able. If a response is cache-able, then a client cache is given the right to reuse that response data for later, equivalent requests. Caching can partially or completely eliminate some interactions and improve efficiency and scalability.
4. Layered System: Layered system constraint, constrains the behavior of components such that each cannot see beyond the immediate layer with which they are interacting. For example, a client cannot tell whether it is connected directly to the end server, or to an intermediary along the way. System scalability can be improved by allowing intermediaries to respond to requests instead of the end server, without the client having to do anything different.
5. Uniform Interface: Uniform Interface constraint requires that the method of communication between a client and a server must be uniform. Resources are identified in the requests (by URIs in web based systems) and are themselves separate from the representations of the resources that are returned to the client. When a client holds a representation of a resource it has all the information required to update or delete the resource (provided the client has required permissions). Each message includes enough information to describe how to process the message.

6. Code on demand: Servers can provide executable code or scripts for clients to execute
   in their context. This constraint is the only one that is optional.

> "***A RESTful web service is a "web API" implemented using HTTP and REST principles.***"

WebSocket :

- WebSocket APIs allow bi-directional, full duplex communication between clients and servers.
- WebSocket APIs follow the exclusive pair communication model
- Unlike request-response APIs such as REST ,the WebSocket API follows full duplex communication and do not require a new connection to be setup for each message to be sent.
- WebSocket communication begins with a connection setup request sent by the client to the server. This request (called a WebSocket handshake) is sent over HTTP and the server interprets it as an upgrade request.
- If the server supports WebSocket protocol, the server responds to the WebSocket handshake response.
- After the connection is setup, the client and server can send data/messages to each other in full-duplex mode.
- WebSocket APIs reduce the network traffic and latency as there is no overhead for connection setup and termination requests for each message.
- WebSocket is suitable for IoT applications that have low latency or high throughput requirements.



Exclusive pair model used by WebSocket API

IoT Enabling Technologies :

IoT is enabled by several technologies including

- Wireless sensor networks
- Cloud computing
- Big data analytics embedded systems, security protocols and architectures, communication protocols, web services, mobile Internet, and semantic search engines.

A Wireless Sensor Network (WSN) :

A Wireless Sensor Network (WSN) comprises of distributed devices with sensors which are used to monitor the environmental and physical conditions.

A WSN consist of a number of end-nodes and routers and a coordinator.

- End nodes have several sensors attached to them. End nodes can also act as routers.
- Routers are responsible for routing the data packets from end-nodes to the coordinator.
- The coordinator collects the data from all the nodes. Coordinator also acts as a gateway that connects the WSN to the Internet.

- Some examples of WSNs used in IoT systems are described as follows:
  - Weather monitoring systems use WSNs in which the nodes collect temperature, humidity and other data, which is aggregated and analyzed.
  - Indoor air quality monitoring systems use WSNs to collect data on the indoor air quality and concentration of various gases.
  - Soil moisture monitoring systems use WSNs to monitor soil moisture at various locations.
  - Surveillance systems use WSNs for collecting surveillance data (such as motion detection data)
  - Smart grids use WSNs for monitoring the grid at various points.
  - Structural health monitoring systems use WSNs to monitor the health of structures (buildings, bridges) by collecting vibration data from sensor nodes deployed at various points in the structure.

- WSNs are enabled by wireless communication protocols such as IEEE 802.15.4.
- ZigBee is one of the most popular wireless technologies used by WSNs. ZigBee specifications are based on IFFF 802.15.4. ZigBee operates at 2.4 GHz frequency and offers data rates upto
  250 KB/s and range from IO to 100 meters depending on the power output and

environmental
conditions.

- The power of WSNs lies in their ability to deploy large number of low-cost and low-power sensing nodes for continuous monitoring of environmental and physical conditions.
- WSNs are self-organizing networks. Since WSNs have large number of nodes, manual configuration for each node is not possible.
- The self-organizing capability of WSN makes the network robust. In the event of failure of some nodes or addition of new nodes to the network, the network can reconfigure itself.

Cloud Computing :

- Cloud computing is a transformative computing paradigm that involves delivering applications and services over the Internet.
- Cloud computing involves provisioning of
  - Computing,
  - Networking and storage resources on demand and providing these resources as metered services to the users, in a "pay as you go" model. Cloud computing resources can be provisioned on-demand by the users, without requiring interactions with the cloud service

Cloud computing services offered :

- Infrastructure-as-a-Service (IaaS) :
  - IaaS provides the users the ability to provision computing and storage resources.
  - These resources are provided to the users as virtual machine instances and virtual storage.

- Users can start, stop, configure and manage the virtual machine instances and virtual storage.
- Users can deploy operating systems and applications of their choice on the virtual resources provisioned in the cloud.
- The cloud service provider manages the underlying infrastructure. Virtual resources provisioned by the users are billed based on a pay-per-use paradigm.

- Platform-as-a-Service (PaaS) :
  - PaaS provides the users the ability to develop and deploy application in the cloud using the development tools, application programming interfaces (APIs), software libraries and services provided by the cloud service provider.
  - The cloud service provider manages the underlying cloud infrastructure including servers, network, operating systems and storage.
  - The users, themselves, are responsible for developing, deploying, configuring and managing applications on the cloud infrastructure.

- Software-as-a-Service (SaaS) :
  - SaaS provides the users a complete software application or the user interface to the application itself.
  - The cloud service provider manages the underlying cloud infrastructure including servers, network, operating systems, storage and application software, and the user is unaware of the underlying architecture of the cloud.
  - Applications are provided to the user through a thin client interface (e.g. a browser).
  - SaaS applications are platform independent and can be accessed from various client devices such as workstations, laptop, tablets and smart-phones, running different operating systems. Since the cloud service

provider manages both the application and data, the users are able to access the applications from anywhere.

Big data Analytics :

- Big data is defined as collections of data sets whose volume, velocity (in terms of its temporal variation) or variety, is so large that it is difficult to store, manage, process and analyze the data using traditional databases and data processing tools.
- Big data analytics involves several steps starting from data cleansing, data munging (or wrangling), data processing and visualization.
- Some examples of big data generated by IoT systems are described as follows:
  - Sensor data generated by IoT systems such as weather monitoring stations.
  - Machine sensor data collected from sensors embedded in industrial and energy systems for monitoring their health and detecting failures.
  - Health and fitness data generated by IoT devices such as wearable fitness bands.
  - Data generated by IoT systems for location and tracking of vehicles.
  - Data generated by retail inventory monitoring systems.

- The underlying characteristics of big data include:
  - Volume:
    - Though there is no fixed threshold for the volume of data to be considered as big data, however, typically, the term big data is used for massive scale data that is difficult to store, manage and process using traditional databases and data processing architectures.
    - The volumes of data generated by modern IT, industrial, and health-care systems, for example, is growing exponentially driven by the lowering costs of data storage and processing architectures and the need to extract valuable insights from the data to improve business processes, efficiency and service to consumers.

- Velocity:
  - Velocity is another important characteristic of big data and the primary reason for exponential growth of data.
  - Velocity of data refers to how fast the data is generated and how frequently it varies.
  - Modern IT, industrial and other systems are generating data at increasingly higher speeds.

- Variety:
  - Variety refers to the forms of the data.
  - Big data comes in different forms such as structured or unstructured data, including text data, image, audio, video and sensor data.

IoT Levels and Deployment Templates :

In this section we define various levels of IoT systems with increasing completely. An IoT
system comprises of the following components:

Components of IoT :

1. Device : An IoT device allows identification, remote sensing, actuating and remote monitoring capabilities.
2. Resource: Resources are software components on the IoT device for accessing, processing, and storing sensor information, or controlling actuators connected to the device. Resources also include the software components that enable network access for the device.

3. Controller Service: Controller service is a native service that runs on the device and interacts with the web services. Controller service sends data from the device to the web service and receives commands from the application (via web services) for controlling the device.
4. Database: Database can be either local or in the cloud and stores the data generated by the IoT device.
5. Web Service: Web services serve as a link between the IoT device, application. database and analysis components. Web service can be either implemented using HTTP and REST principles (REST service) or using WebSocket protocol (WebSocket service). A comparison of REST and WebSocket is provided below:

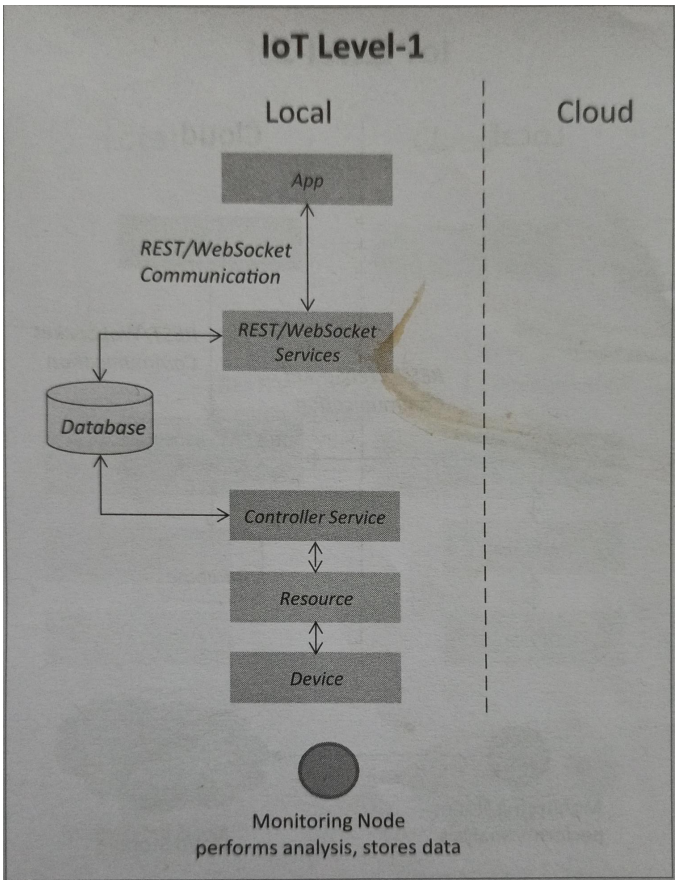
	REST	WebSocket
Stateless/Stateful:	REST services are stateless in nature. Each request contains all the information needed to process it. Requests are independent of each other.	WebSocket on the other hand is stateful in nature where the server maintains the state and is aware of all the open connections.
Uni-directional/Bi-directional:	REST services operate over HTTP and are Uni-directional. Request is always sent by a client and the server responds to the requests.	WebSocket is a bi-directional protocol and allows both client and server to send messages to each other.
Request-Response/Full Duplex:	REST services follow a request-response communication model where the client sends requests and the server responds to the requests.	WebSocket on the other hand allow full-duplex communication between the client and server, i.e., both client and server can send messages to each other independently.
TCP Connections	For REST services, each HTTP request involves setting up a new TCP connection.	WebSocket on the other hand involves a single TCP connection over which the client and server communicate in a full-duplex mode.

	REST	WebSocket
Header Overhead:	REST services operate over HTTP, and each request is independent of others. Thus each request carries HTTP headers which is an overhead. Due to the overhead of HTTP headers, REST is not suitable for real-time applications.	WebSocket on the other hand does not involve overhead of headers. After the initial handshake (that happens over HTTP), the client and server exchange messages with minimal frame information. Thus WebSocket is suitable for real-time applications.
Scalability:	Scalability is easier in the case of REST services as requests are independent and no state information needs to be maintained by the server. Thus both horizontal (scaling-out) and vertical scaling (scaling-up) solutions are possible for REST services.	For Web Sockets, horizontal scaling can be cumbersome due to the stateful nature of the communication. Since the server maintains the state of a connection, vertical scaling is easier for Web Sockets than horizontal scaling.

6. Analysis Component: The Analysis Component is responsible for analyzing the IoT data and generate results in a form which are easy for the user to understand. Analysis of IoT data can be performed either locally or in the cloud. Analyzed results are stored in the local or cloud databases.
7. Application: IoT applications provide an interface that the users can use to control and monitor various aspects of the IoT system. Applications also allow users to view the system status and view the processed data.

Levels of IoT :

Level - 1 :



Single node + Local analysis and Storage

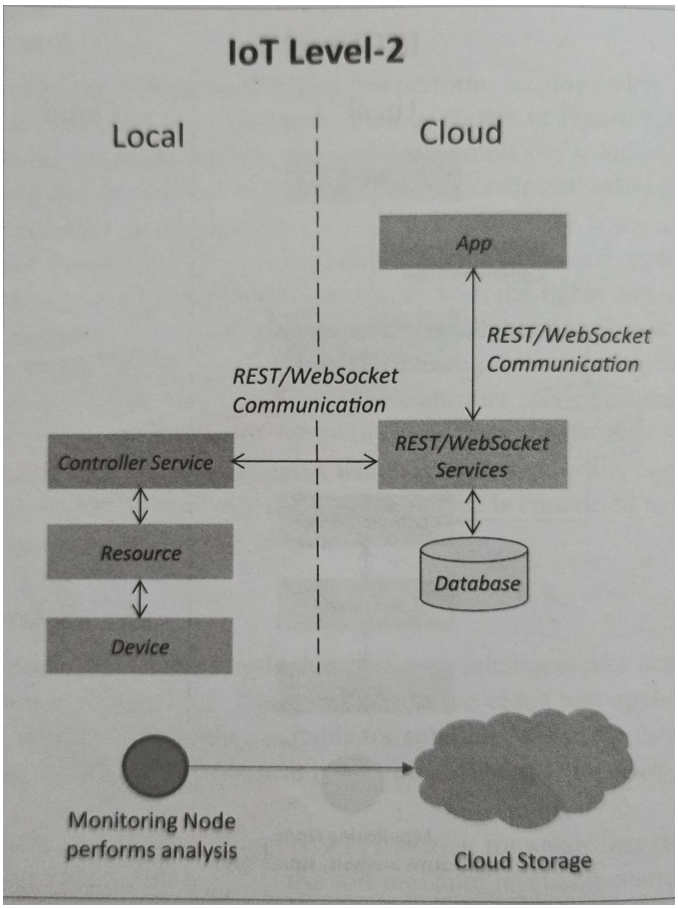
💡 DEFENITION :

1. A level-I IoT system has a single node/device that performs sensing and/or actuation, stores data, performs analysis and hosts the application.
2. Level-I IoT systems are suitable for
    a. Modeling low-cost and low-complexity solutions where the data involved is not big
    b. The analysis requirements are not computationally intensive.

Let us now consider an example of a level-I IoT system for home automation.

- **The system** consists of a single node that allows controlling the lights and appliances in a home remotely.
- **The device** used in this system interfaces with the lights and appliances using electronic relay switches.
- The status information of each light or appliance is maintained in a local database.
- REST services deployed locally allow retrieving and updating the state of each light or appliance in the status database.
- **The controller service** continuously monitors the state of each light or appliance (by retrieving state from the database) and triggers the relay switches accordingly.
- The application which is deployed locally has a user interface for controlling the lights or appliances. Since the device is connected to the Internet, the application can be accessed remotely as well.

Level - 2 :

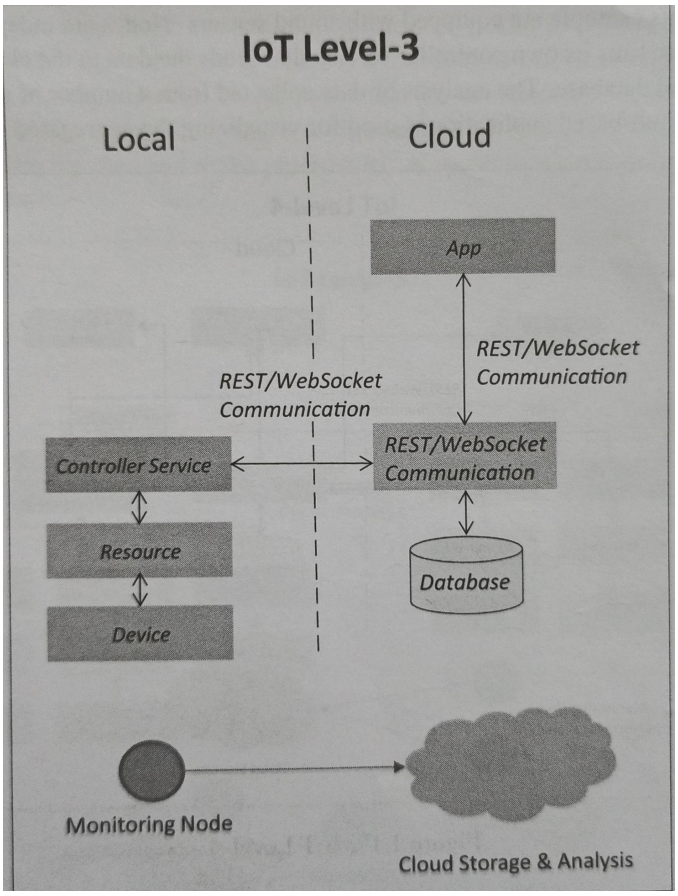


Single node + Cloud storage + Local Analysis

Let us consider an example of a level-2 IoT system for smart irrigation.

- **The system** consists of a single node that monitors the soil moisture level and controls the irrigation system.
- **The device** used in this system collects soil moisture data from sensors.
- **The controller service** continuously monitors the moisture levels.
- If the moisture level drops below a threshold, the irrigation system is turned on.
- For controlling the irrigation system actuators such as solenoid valves can be used. The controller also sends the moisture data to the computing cloud.
- A cloud-based REST web service is used for storing and retrieving moisture data which is stored in the cloud database.
- A cloud-based application is used for visualizing the moisture levels over a period of time, which can help in making decisions about irrigation schedules.

Level -3 :

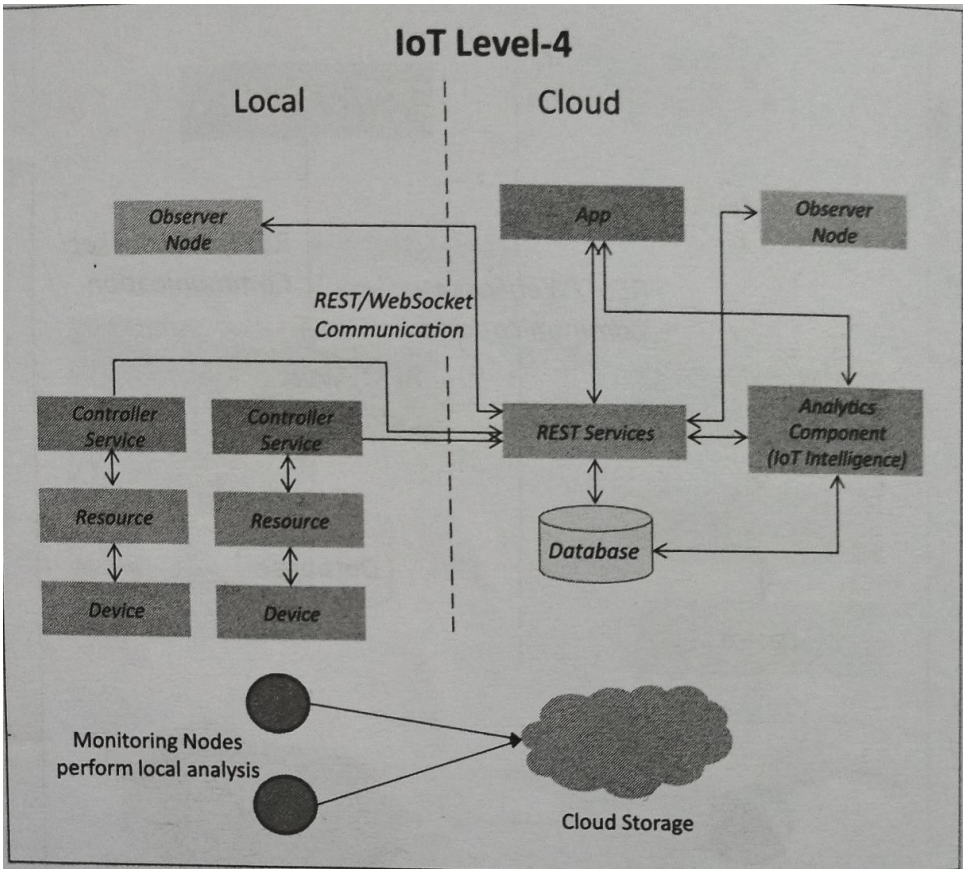


Single node + Cloud analysis and Storage

Let us consider an example of a level-3 IoT system for tracking package handling.

- **The system** consists of a single node (for a package) that monitors the vibration levels for a package being shipped.
- **The device** in this system uses accelerometer(vibration) and gyroscope(orientation) sensors
  for monitoring vibration levels.
- **The controller service** sends the sensor data to the cloud in real-time using a WebSocket service. The data is stored in the cloud and also visualized using a cloud-based application.
- The analysis components in the cloud can trigger alerts if the vibration levels greater than a threshold.
- The benefit of using WebSocket service instead of REST service in this example is that, the sensor data can be sent in real time to the cloud.
- Moreover, cloud based applications can subscribe to the sensor data feeds for viewing the real-time data.

Level - 4 :

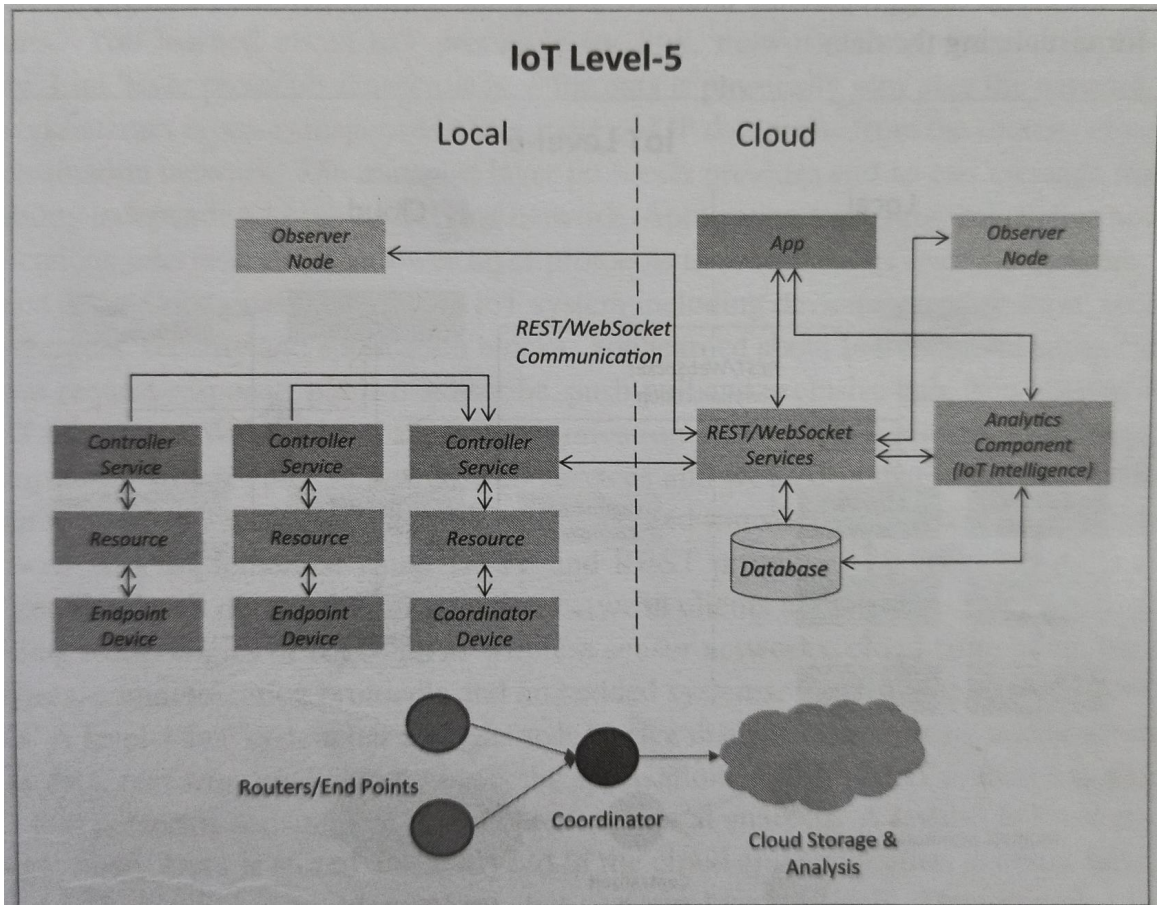


Multiple indie nodes + Observer Node + Local Analysis + Cloud Storage

Let us consider an example of a level-4 IoT system for noise monitoring.

- **The system** consists of multiple nodes placed in different locations for monitoring noise levels in an area.
- The nodes in this example are equipped with sound sensors. Nodes are independent of each other.
- Each node runs its own controller service that sends the data to the cloud.
- The data is stored in a cloud database.
- The analysis of data collected from a number of nodes is done in the cloud. A cloud-based application is used for visualizing the aggregated data.

Level - 5 :

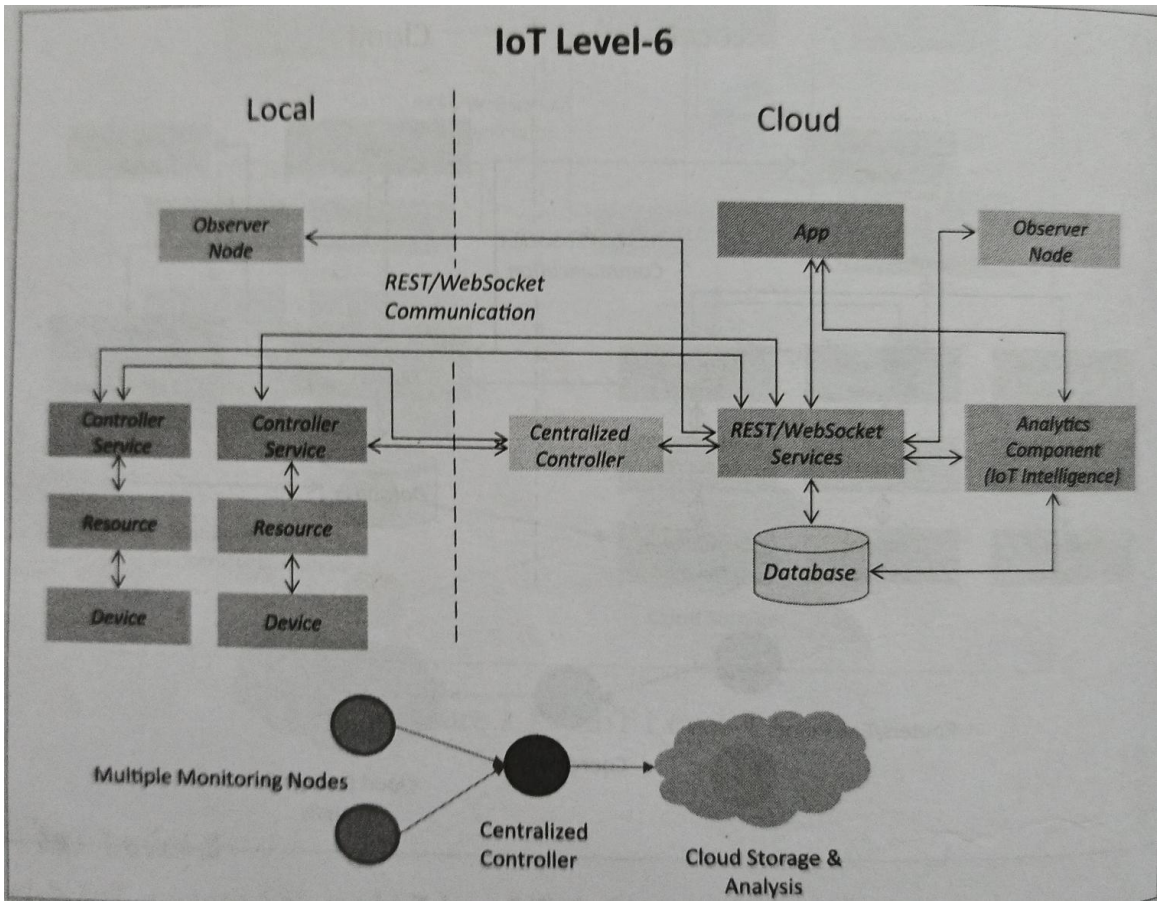


Multiple end nodes + Observer Node + Co-Ordinator device + Cloud Storage and Analysis

Let us consider an example of a level-5 IoT system for forest fire detection.

- **The system** consists of multiple nodes placed in different locations for monitoring temperature, humidity and carbon dioxide (C02) levels in a forest.
- The end nodes in this example are equipped with various sensors (such as temperature, humidity and C02).
- The coordinator node collects the data from the end and acts as a gateway that provides Internet connectivity to the IoT system.
- **The controller service** on the coordinator device sends collected data to the cloud.
- The data is stored in a cloud database.
- The analysis of data is done in the computing cloud to aggregate the data and make predictions.
- A cloud-based application is used for visualizing the data.

Level - 6 :



Multiple independent nodes + Observer node + Centralized Controller node + Cloud storage and Analysis

Domain Specific IoT :

Home Automation :

Name	Description and usages	Devices

Name	Description and usages	Devices
Smart Lightning :	Smart lighting for homes helps in saving energy by adapting the lighting to the ambient conditions and switching on/off or dimming the lights when needed. Smart lighting solutions for home achieve energy savings by sensing the human movements and their environments and controlling the lights accordingly. Wireless-enabled and Internet connected lights can be controlled remotely from IoT applications such as a mobile or web application. Smart lights with sensors can configured to adapt the lighting (by changing the light intensity, color, etc.) based the ambient conditions sensed,	Solid State Lighting(LED) ⇒ IP-enabled lights ⇒ Occupancy sensor ⇒ Lux ⇒ Temperature sensor

Name	Description and usages	Devices
Smart Appliances	Modern homes have a number of appliances such as TVs, refrigerators, systems, washer/dryers, etc. Managing controlling these appliances can be cumbersome, with each appliance having its own controls or remote controllers Smart appliances make the management easier and also provide status information to the users remotely. For example, Smart washer/dryers that can be controlled remotely and notify when the washing/drying cycle is complete. Smart thermostats allow controlling the temperature remotely and can learn the user preferences Smart refrigerators can keep track of the items stored (using RFID tags) and send updates to users when an item is low on stock. Smart TVs allows users to search and stream videos and movies from the Internet on a local storage drive, search TV channel schedules and fetch news, weather ,updates and other content from the Internet OpenRemote is an source automation platform for homes and buildings. OpenRemote is platform agnostic and works with standard hardware. With OpenRemote, users can control various appliances using mobile or web applications. **OpenRemote comprises of three components : Controller** that manages scheduling and runtime integration between devices. **Designer** that allows you to create both configurations for the controller and create user interface designs **Control Panels** that allow you to interact with devices and control them.	

Name	Description and usages	Devices
Intrusion Detection	Home intrusion detection systems use security cameras and sensors (such as PIR sensors and door sensors) to detect intrusions and raise alerts. Alerts can be in the form of an SMS or an email sent to the user. Advanced systems can even send detailed alerts such as an image grab or a short video clip sent as an email attachment. A cloud controlled intrusion detection system that uses location-aware services, where the geo-location of each node of a home automation system is independently detected and stored in the cloud. In the event of intrusions, the cloud services alert the accurate neighbors (who are using the home automation system) or local police. An intrusion detection system based on UPnP technology is described. The system uses image processing to recognize the intrusion and extract the intrusion subject and generate Universal-mug-and-Play (UPnP-based) instant messaging for alerts.	- PIR Sensor - Door Sensor
Smoke/Gas Detectors	Smoke detectors are installed in homes and buildings to detect smoke that is typically an early sign of fire. Smoke detectors use optical detection, ionization or air sampling techniques to detect smoke. Alerts raised by smoke detectors can be in the form of signals to a fire alarm system. Gas detectors can detect the presence of harmful gases such as carbon monoxide (CO), liquid petroleum gas (LPG), etc. A smart smoke/gas detector (22) can raise alerts in human voice describing where the problem is, send or an SMS or email to the user or the local fire safety department and provide visual feedback on its status (healthy, battery-low, etc.).	Smoke Detectors

Cities :

Name	Description and Usage

Name	Description and Usage
Smart Parking	Smart parking make the search for parking space easier and convenient for drivers. Smart parking are powered by IoT systems that detect the number of empty parking slots and send the information over the Internet to smart parking application back-ends. These applications can be accessed by the drivers from smart-phones, tablets and in-car navigation systems. In smart parking, sensors are used for each parking slot, to detect whether the slot is empty or occupied. This information is aggregated by a local controller and then sent over the Internet to the database.
Smart Lightning	Smart lighting allows lighting to be dynamically controlled and also adaptive to the ambient conditions Smart lights connected to the Internet can controlled remotely to configure lighting schedules and lighting intensity. Custom lighting configurations can be set for different situations such as a foggy day, a festival etc. Smart lights equipped with sensors can communicate with other lights and exchange information on the sensed ambient conditions to adapt the lighting.
Smart Roads	Smart roads equipped with sensors can provide information on driving conditions, travel time estimates and alerts in case of poor driving conditions, traffic congestions and accidents. Information sensed from the roads can be communicated via Internet to cloud-based applications and social media and disseminated to the drivers who subscribe to such applications. In a distributed and autonomous system of sensor network nodes for improving driving safety on public roads in proposed. The system can provide the drivers and passengers with a consistent view of the road situation a few hundred meters ahead of them or a few dozen miles away, so that they can react to potential dangers early enough.
Structural Health Monitoring	Structural Health Monitoring systems use a network of sensors to monitor the vibration levels in the structures such as bridges and buildings. The data collected from these sensors is analyzed to assess the health of the structures. By analyzing the data it is possible to detect cracks and mechanical breakdowns, locate the damages to a structure and also calculate the remaining life of the structure. Using such systems, advance warnings can be given in the case of imminent failure of the structure.
Surveillance	Surveillance of infrastructure, public transportation and events in cities is required to ensure safety and security. City wide surveillance infrastructure comprising of large number Of distributed and Internet connected video surveillance cameras can be created. The video feeds from surveillance cameras can be aggregated in cloud-based scalable storage solutions. Cloud-based video analytics applications can be developed to search for patterns or specific events from the video feeds.

Name	Description and Usage
Emergency Response	IoT systems can be used for monitoring the critical infrastructure in cities such as buildings, gas and water pipelines, public transport and power substations. IoT systems for fire detection, gas and water leakage detection can help in generating alerts and minimizing their effects on the critical infrastructure. IoT systems for critical infrastructure monitoring enable aggregation and sharing of information collected from large amount of sensors. Using cloud-based architectures, multi-modal information such as sensor data, audio, video feeds can be analyzed in pear real-time to detect adverse events. Response to alerts generated by such systems can be in the form of alerts sent to the public, re-routing of traffic, evacuations of the affected areas, etc. The system adapts by dynamically adjusting traffic lights, changing related driving policies, recommending behavior change to drivers. and applying essential security controls. Such systems can reduce the latency of emergency services for vehicles such as ambulances and police cars while minimizing disruption of regular traffic.

Environment :

Name	Description and Usage
Weather Monitoring Systems	The data collected in the cloud can then analyzed and visualized by cloud-based applications. Weather alerts can be sent to the subscribed users from such application The data collected in the cloud can then analyzed and visualized by cloud-based applications. Weather alerts can be sent to the subscribed users from such applications. AirPi [38] is a weather and air quality monitoring kit capable of recording and uploading information about temperature, humidity, air pressure, light levels, UV levels carbon monoxide. nitrogen dioxide and smoke level to the Internet.
Pollution Monitoring System :	IoT based air pollution monitoring systems can monitor emission of harmful gases (C02, CO, NO, N02, etc.) by factories and automobiles using gaseous and meteorological sensors. The collected data can be analyzed to make informed decisions on pollutions control approaches. In a real-time air quality monitoring system is presented that comprises of several distributed monitoring stations that communicate via wireless with a back-end server using machine-to-machine communication. In an air pollution system is described that integrates a single-chip microcontroller, several air pollution sensors, GPRS-Modem, and GPS module.

Name	Description and Usage
Noise Pollution Monitoring	Noise pollution monitoring can help in generating noise maps for cities. Urban noise maps can help the policy makers in urban planning and making policies to control noise levels near residential areas, schools and parks. The data on noise levels from the stations is collected on servers or in the cloud. The collected data is then aggregated to generate noise maps and to a noise visualization map.
Forest Fire Detection	A system for early detection of forest fires is described in (451 that provides early warning of a potential forest fire and estimates the scale and intensity of the fire if it materializes. Early detection of forest fires can help in minimizing the damage. IoT based forest fire detection systems use a number of monitoring nodes deployed at different locations in a forest. Each monitoring node collects measurements on ambient conditions including temperature, humidity, light levels, etc. The system uses multi-criteria detection which is implemented by the artificial neural network (ANN). The ANN fuses sensing data corresponding to multiple attributes of forest fire (such as temperature, humidity, infrared and visible light) to detect forest fires.
Flood Detection	IoT based river flood monitoring system use a number of sensor nodes that monitor the water level (using ultrasonic sensors) and flow rate (using the flow velocity sensors). Early warnings of floods can be given by monitoring the water level and flow rate. Data from a number of such sensor is aggregated in a server or in the cloud. Monitoring applications raise alerts when rapid increase in water level and flow rate is detected. The system includes a water level monitoring module, network video recorder module, and data processing module that provides flood information in the form of raw data, predicted data, and video feed.

Energy :

Name	Description and Usage

Name	Description and Usage
Smart Grids	Smart Grid is a data communications network integrated with the electrical grid that collects and analyzes data captured in near-real-time about power transmission, distribution, and consumption. Smart Grid technology provides predictive information and recommendations to utilities, their suppliers, and their customers on how best to manage power. Smart Grids collect data regarding electricity generation (centralized or distributed), consumption (instantaneous or storage or conversion of energy into and equipment health data. Smart grids use high-speed, fully integrated, two-way communication technologies for real-time information and power exchange. By using IoT based sensing measurement technologies, the of real-time and the integrity of the grid can be evaluated. Smart meters can capture almost real-time consumption, remotely control the consumption of electricity and remotely switch off supply when required. Power thefts can be prevented using smart metering. By analyzing the data on power generation, transmission and consumption smart girds can efficiency throughout the electric system. Storage collection and analysis of smarts grids data in the cloud can help in dynamic optimization of system operations, maintenance, and planning. Cloud-based monitoring of smart grids data can improve energy usage levels via to users coupled with real-time pricing information. Real-time demand response and management strategies can used for lowering peak demand and overall via appliance control and energy storage mechanisms. Condition monitoring data collected from power generation and transmission systems can help in detecting faults and predicting outages.
Renewable Energy System	Due to the variability in the output from renewable energy resources (such as solar and wind), integrating them into the grid can cause grid stability and reliability problems. Variable output produces local voltage swings that can impact power quality. Existing grids were designed to handle power flows from centralized generation sources to the loads through transmission and distribution lines. When distributed renewable energy sources are integrated into the grid, they create power bi-directional power flows for which the grids were not originally designed. IoT based systems integrated with the transformers at the point of inter connection measure the electrical variables and much power is fed into the grid. To ensure the grid stability, one solution is to simply cut off the overproduction.

Name	Description and Usage
Prognostics	Energy systems (smart grids, power plants, wind turbine farms, for instance) have a large number of critical components that must function correctly so that the systems can perform their operations correctly. For example, A wind turbine has a number of critical components. e.g.. bearings, turning gears, for instance. that must be monitored carefully as wear and tear in such critical components or sudden change in operating conditions of the machines can result in failures. In systems such as power grids, real-time information is collected using Specialized electrical sensors called Phasor Measurement Units (PMU) at the substations. The information received from PMUs must be monitored in real-time for estimating the state of the system and for predicting failures. Energy systems have thousands of sensors

Retail :

Name	Description and Usage
Inventory Management	While over-stocking of products can result in additional storage expenses and risk (in case of perishables), under-stocking can lead to loss of revenue. IoT systems using Radio Frequency Identification (RFID) tags can help in inventory management and maintaining the right inventory levels. RFID tags attached to the products allow them to be tracked in real-time so that the inventory levels can be determined accurately and products which are low on stock can be replenished. Tracking can be done using RFID readers attached to the retail store shelves or in the warehouse. IoT systems enable remote monitoring of inventory using the data collected by the RFID readers
Smart Payments	Smart payment solutions such as contact-less payments powered by technologies such as Near field communication (NFC) and Bluetooth. Near field communication (NFC) is a set of standards for smart-phones and other devices to communicate with each other by bringing them into proximity or by touching them. Customers can store the credit card information in their NFC-enabled smart-phones and make payments by bringing the smart-phones near the point of sale terminals. NFC maybe used in combination with Bluetooth, where NFC (which offers low speeds) initiates initial pairing of devices to establish a Bluetooth connection while the actual data transfer takes place over Bluetooth.

Name	Description and Usage
Smart vending machines	Smart vending machines connected to the Internet allow remote monitoring of inventory levels, elastic pricing of products, promotions, and contact-less payments using NFC. Smart-phone applications that communicate with smart vending machines allow user preferences to be remembered and learned with time. When a user moves from one vending machine to the other and pairs the smart-phone with the vending machine, a user specific interface is presented. Users can save their preferences and favorite products. Sensors in a smart vending machine monitor its operations and send the data to the cloud which can be used for predictive maintenance. Smart vending machines can communicate with other vending machines in their vicinity and share their inventory levels so that the customers Can be routed to the nearest machine in case a product out of stock in a machine. For perishable items, the smart vending machines can reduce the price as the expiry date nears. New products can be recommended to the customers based on the purchase history and preferences.

Logistics

Name	Description and Usage
Route Generation & Scheduling	Modern transportation systems are driven by data collected from multiple sources which is processed to provide new services to the stakeholders. By collecting large amount of data from various sources and processing the data into useful information, data-driven transportation systems can provide new services such as advanced route guidance ,dynamic vehicle routing, anticipating customer demands for pickup and delivery problem. Route generation and scheduling systems can generate end-to-end routes using combination of route patterns and transportation modes and feasible schedules based on the availability of vehicles. IoT based systems backed by the cloud can provide fast response to the route generation queries and can be scaled up to serve a large transportation network.

Name	Description and Usage
Fleet Tracking	Vehicle fleet tracking systems use GPS technology to track the locations of the vehicles in real-time Cloud-based fleet tracking systems can be scaled up on demand to handle large number of vehicles Alerts can be generated in case of deviations in planned routes. The vehicle locations and routes data can be aggregated and analyzed for detecting bottlenecks in the supply chain such as traffic congestions on routes, assignments and generation of alternative routes, and supply chain optimization. The system can analyze messages sent from the vehicles to identify unexpected incidents and discrepancies between actual and planned data, so that remedial actions can be taken.
Shipment monitoring	Shipment monitoring solutions for transportation systems allow monitoring the conditions inside containers. For example, containers carrying fresh food produce can be monitored to prevent spoilage of food. IoT based shipment monitoring systems use sensors, such as pressure, humidity, for instance, to monitor the conditions inside the containers and send the data to the cloud, where it can be analyzed to detect food spoilage. The analysis and interpretation of data on the environmental conditions in the container and food truck positioning can enable more effective routing decisions in real time. Therefore, it is possible to take remedial measures such as - the food that has a limited time budget before it gets rotten can be re-routed to a closer destinations, alerts can be raised to the driver and the distributor about the transit conditions, such as container temperature exceeding the allowed limit, humidity levels going out of the allowed limit, for instance, and corrective actions can be taken before the food gets damaged. A cloud-based framework for real-time fresh food supply tracking and monitoring was proposed for fragile products, vibration levels during shipments can be tracked using accelerometer and gyroscope sensors attached to IoT devices. A system for monitoring container integrity and operating conditions is described. The system monitors the vibration patterns of a container and its contents to reveal information related to its operating environment and integrity during transport, handling and storage.

Name	Description and Usage
Remote Vehicle Diagnostics	Vehicle diagnostic systems can detect faults in the vehicles or warn of impending faults These diagnostic systems use on-board IoT devices for collecting data on vehicle operation (such as speed. engine RPM, coolant temperature, fault code number) and status of various vehicle sub-systems. Such data can be captured by integrating on-board diagnostic systems with IoT devices using protocols such as CAN bus. Modern commercial vehicles support on-board diagnostic (OBD) standards such as OBD-II. OBD systems provide real-time data on the status of vehicle sub-systems and diagnostic trouble codes which allow rapidly identifying the faults in the vehicle. IoT based vehicle diagnostic systems can send the vehicle data to centralized servers or the cloud where it can analyzed to generate alerts and suggest remedial actions.

Agriculture :

Name	Description and usage
Smart Irrigation	Smart irrigation systems can improve crop yields while saving water. Smart irrigation systems use IoT devices with soil moisture sensors to determine the amount of moisture in the soil and release the flow of water through the irrigation pipes only when the moisture levels go below a predefined threshold. Smart irrigation systems also collect moisture level measurements on a server or in the cloud where the collected data can be analyzed to plan watering schedules. Cultivar's RainCloud (561 is a device for smart irrigation that uses water valves, soil sensors and a WiFi enabled programmable computer.

Name	Description and usage
Green House Control	Green houses are structures with glass or plastic roofs that provide conducive environment for growth of plants. The climatological conditions inside a green house can be monitored and controlled to provide the best conditions for growth of plants. The temperature, humidity, soil moisture, light and carbon dioxide levels are monitored using sensors and the climatological conditions are controlled automatically using actuation devices (such as valves for releasing water and switches for controlling fans). IoT systems play an important role in green house control and help in improving productivity. The data collected from various sensors is stored on centralized servers or in the cloud where analysis is performed to optimize the control strategies and also correlate the productivity with different control strategies. The system uses wireless sensor network to monitor and control the agricultural parameters like temperature and humidity in real time for better management and maintenance of agricultural production.

Industry :

Name	Description and Usage
Machine Diagnosis & Prognosis	Machine prognosis refers to predicting the performance of a machine by analyzing the data on the current operating conditions and how much deviations exist from the normal operating conditions. Machine diagnosis refers to determining the cause of a machine fault. IoT plays a major role in both prognosis and diagnosis Of industrial machines. Industrial machines have a large number of components that must function correctly for the machine to perform its operations. Sensors in machines can monitor the operating conditions such as (temperature and vibration levels). The sensor data measurements are done on timescales of few milliseconds to few seconds, which leads to generation of massive amount of data. IoT based systems integrated with cloud-based storage and analytics back-ends can help in storage, collection and analysis of such massive scale machine sensor data. Number of methods have been proposed for reliability analysis and fault prediction in machines. Case-based reasoning (CBR) is a commonly used method that finds solutions to new problems based on past experience.

Name	Description and Usage
Indoor Air Quality Monitoring	Monitoring indoor air quality in factories is important for health and safety of the workers. IoT based gas monitoring systems can help in monitoring the indoor air quality using various gas sensors. The indoor air quality can vary for different locations. Wireless sensor networks based IOT devices can identify the hazardous zones, so that corrective measures can be taken to ensure proper ventilation. In a hybrid sensor system for indoor air quality monitoring is presented, which contains both stationary sensors (for accurate readings and calibration) and mobile sensors (for coverage).

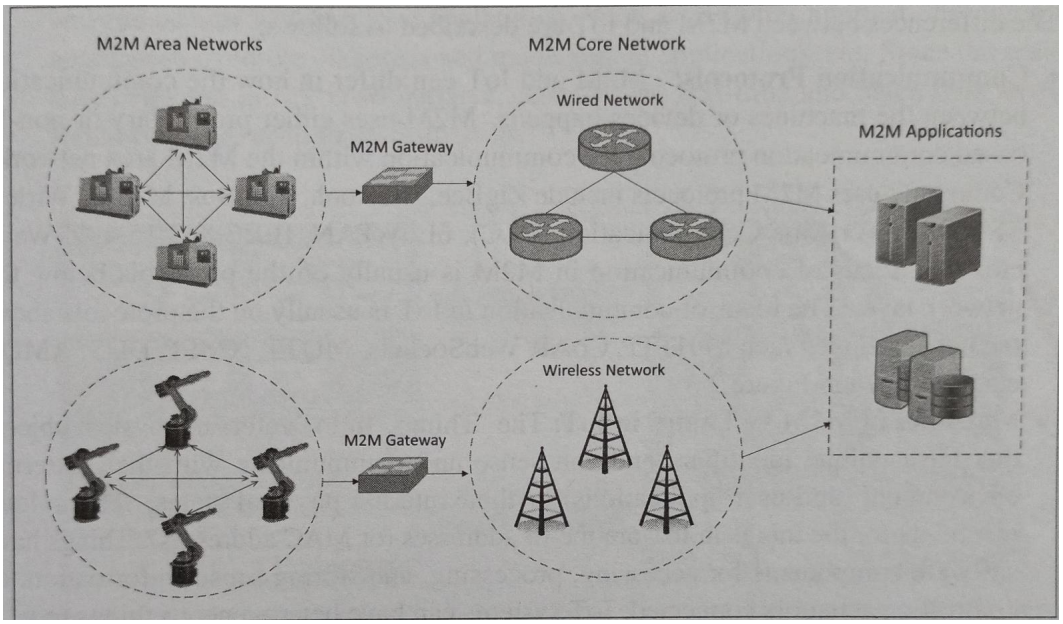
Health and Lifestyle :

Name	Description and Usage
Health & Fitness Monitoring	Wearable IoT devices that allow non-invasive and continuous monitoring of physiological parameters can help in continuous health and fitness monitoring. These wearable devices may can be in various forms such as belts and wrist-bands. The wearable devices form a type of wireless sensor networks called body area networks in which the measurements from a number of wearable devices are continuous sent to a master node (such as a smart-phone) which then sends the data to a server or a cloud-based back-end for analysis and archiving. Health-care providers can analyze the collected health-care data to determine any health conditions or anomalies. Commonly uses body sensors include: body temperature, heart rate, pulse oximeter oxygen saturation (SP02), blood pressure, electrocardiogram (ECG) movement (with accelerometers), and electroencephalogram (EEG). A wearable ubiquitous health-care monitoring system is presented that uses integrated electrocardiogram (ECG), accelerometer and oxygen saturation (Sp02) sensors. Fitbit wristband [741 is a wearable device that tracks steps, distance, and calories burned during the day and sleep quality at night.

Name	Description and Usage
Wearable Electronics	Wearable electronics such as wearable gadgets (smart watches, smart glasses, wristbands, etc.) and fashion electronics (with electronics integrated in clothing and accessories, (e.g. Google Glass or Moto 360 smart watch) provide various functions and features to assist us in our daily activities and making us lead healthy lifestyles. Smart watches that run mobile operating systems (such as Android) provide enhanced functionality beyond just timekeeping. With smart watches, the users can search the Internet, play audio/video files, make calls (with or without paired mobile phones), play games and use various kinds of mobile applications Smart glasses allows users to take photos and record videos, get map directions, check flight status, and search the Internet by using voice commands Smart shoes monitor the walking or running speeds and jumps with the help of embedded sensors and be paired with smart-phones to visualize the data Smart wristbands can track the daily exercise and calories burnt.

IoT and M2M :

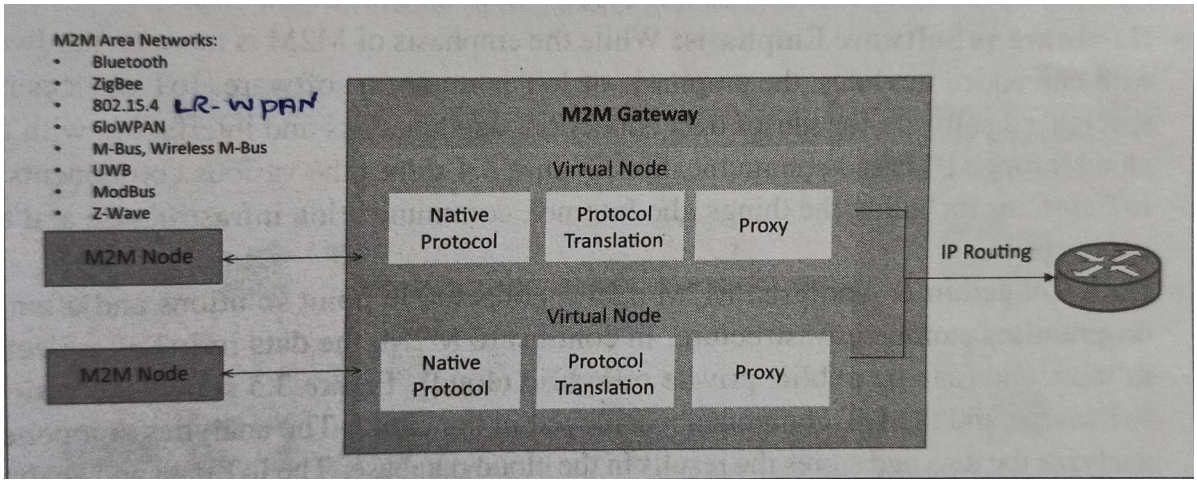
1. Machine-to-Machine (M2M) refers to networking of machines (or devices) for the purpose of
    - Remote monitoring
    - Control
    - and Data exchange.
2. An M2M area network comprises of machines (or M2M nodes) which have embedded hardware modules for sensing, actuation and communication.
3. Various communication protocols can be used for M2M local area networks such as ZigBee, Bluetooth, Modbus, M-Bus, Wireless M-Bus, Power Line Communication (PLC), 6LoWPAN, IEEE 802.15.4, etc. These communication protocols provide connectivity between M2M nodes within an M2M area network.



M2M architecture

- The communication network provides connectivity to remote M2M area networks.
- The communication network can use either wired or wireless networks (IP-based).
- While the M2M area networks use either proprietary or non-IP based communication protocols
- The communication network uses IP-based networks. Since non-IP based protocols are used within M2M area networks
- The M2M nodes within one network cannot communicate with nodes in an external network.

4. To enable the communication between remote M2M area networks, M2M gateways are used.



M2M Gateway

- The communication between the M2M nodes and the M2M gateway is based on the communication protocols which are native to the M2M area network.
- M2M gateway performs protocol translations to enable IP-connectivity for M2M area networks.
- M2M gateway acts as a proxy performing translations from/to native protocols to/from Internet Protocol (IP).
- With an M2M gateway, each node in an M2M area network appears as a virtualized node for external M2M area networks.
- The M2M data is gathered into point solutions such as enterprise applications, service
  management applications, or remote monitoring applications.
- M2M has various application domains such as smart metering, home automation, industrial automation, smart grids, etc.
- M2M solution designs (such as data collection and storage architectures and applications)
  are specific to the M2M application domain.
- The M2M data is gathered into point solutions such as enterprise applications, service
  management applications, or remote monitoring applications.

- M2M has various application domains such as smart metering, home automation, industrial automation, smart grids, etc.
- M2M solution designs (such as data collection and storage architectures and applications)
  are specific to the M2M application domain.

Difference between IoT and M2M :

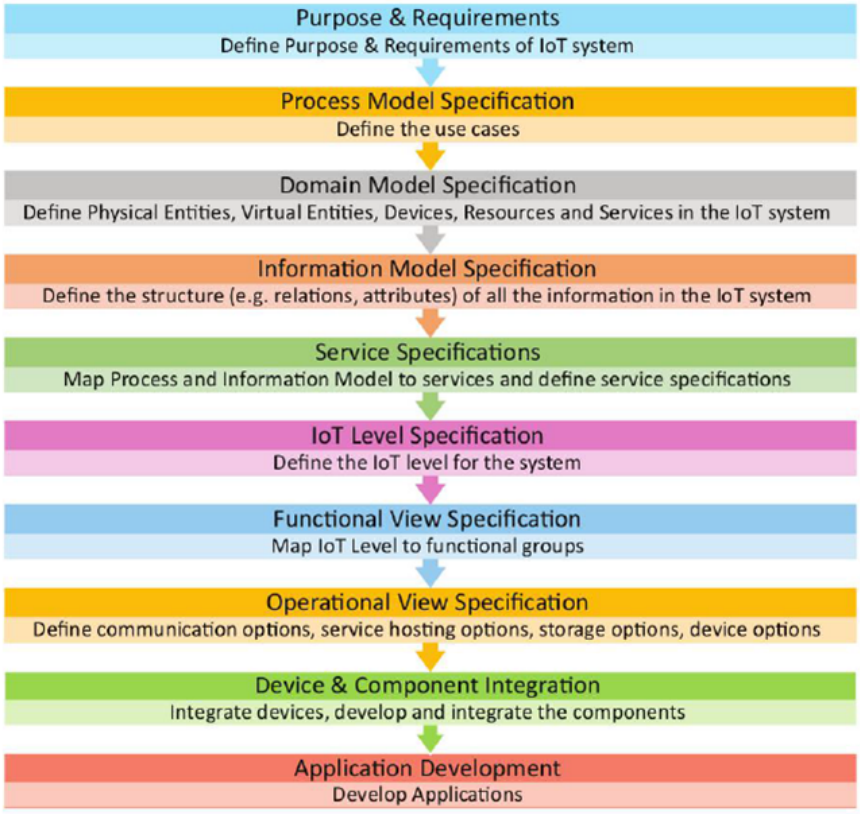
	IoT	M2M
Communication protocol	The focus of communication in IoT is usually on the protocols above the network layer such as HTTP, CoAP, Web Sockets, MQTT, XMPP, DDS, AMQP, etc.	M2M uses either proprietary or non-IP based communication protocols for communication within the M2M area networks. Commonly uses M2M protocols include ZigBee, Bluetooth, ModBus, M-Bus, Wireless M-Bus, Power Line Communication (PLC), 6LoWPAN, IEEE 802.15.4, Z.-wave, etc. The focus of communication in M2M is usually on the protocols below the network layer.

	IoT	M2M
Machines in M2M vs Things in IoT:	The "Things in IoT refers top physical objects that have unique identifiers and can sense and communicate with their external environment (and user applications) or their internal physical states. The unique identifiers for the things in IoT are the IP addresses (or MAC addresses). Things have software components for accessing, processing. and storing sensor information, or controlling actuators connected. IoT system can have heterogeneous things (e.g., a home automation IoT system can include IoT devices of various types, such as fire alarms, door alarms, lighting control devices, etc.)	M2M systems, in contrast to IoT, typically have homogeneous machine types within an M2M area network.
Hardware vs Software Emphasis	The emphasis of IoT is more on software. IoT devices run specialized software for sensor data collection; data analysis and interfacing with the cloud through IP-based communication. IoT systems including the things, the Internet, communication infrastructure and the applications.	While the emphasis of M2M is more on hardware with embedded modules,

	IoT	M2M
Data Collection & Analysis:	In IoT data is collected in the cloud (can be public, private or hybrid cloud). The analytics component analyzes the data and stores the results in the cloud database. The IoT data and analysis results are visualized with the cloud-based applications The centralized controller is aware of the status of all the end nodes and sends control commands to the nodes. Observer nodes can process information and use it for various applications, however, observer nodes do not perform any control functions.	M2M data is collected in point solutions and often in on-premises storage infrastructure.
Applications:	IoT data is collected in the cloud and can be accessed by cloud applications such as analytics applications, enterprise applications, remote diagnosis and management applications, etc. Since the scale of data collected in IoT is so massive, cloud-based real-time and batch data analysis frameworks are used for data analysis.	M2M data is collected in point solutions and can access on-premises applications such as diagnosis applications, service management applications, and on-premises enterprise applications.

IoT Platform Design methodologies :

Example : Home Automation System



1. Purpose and Requirements Specification :

First step is to define the purpose and requirements of the system. In this step, the system purpose, behavior and requirements are captured.

Requirements can be:

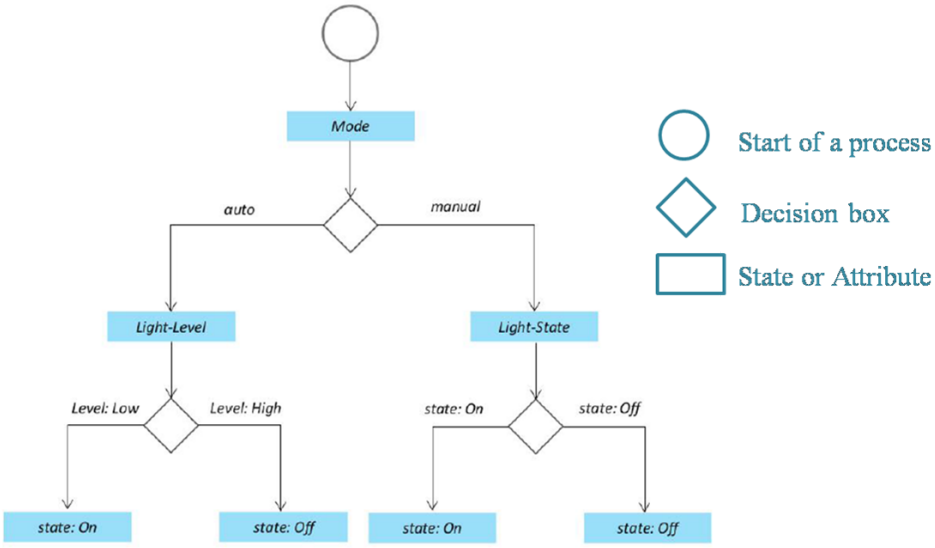
- Data collection requirements
- Data analysis requirements
- System management requirements
- Security requirements
- User interface requirements

Purpose	A home automation system that allows controlling the lights remotely using a

	web application
Behavior	Home automation system should support two modes: auto and manual **Auto:** System measures the light level in the room and switches on the light when it is dark **Manual:** Allows remotely switching lights on and off
System Management	System should provide remote monitoring and control functions
Data Analysis	System should perform local analysis of the data
Application Deployment	Application should be deployed locally, but should be accessible remotely
Security	Should provide basic security like user authentication

2. Process Specification

The use cases of the IoT system are formally described based on or derived from the purpose and requirements specifications. The process specification for home automation system is as shown below.



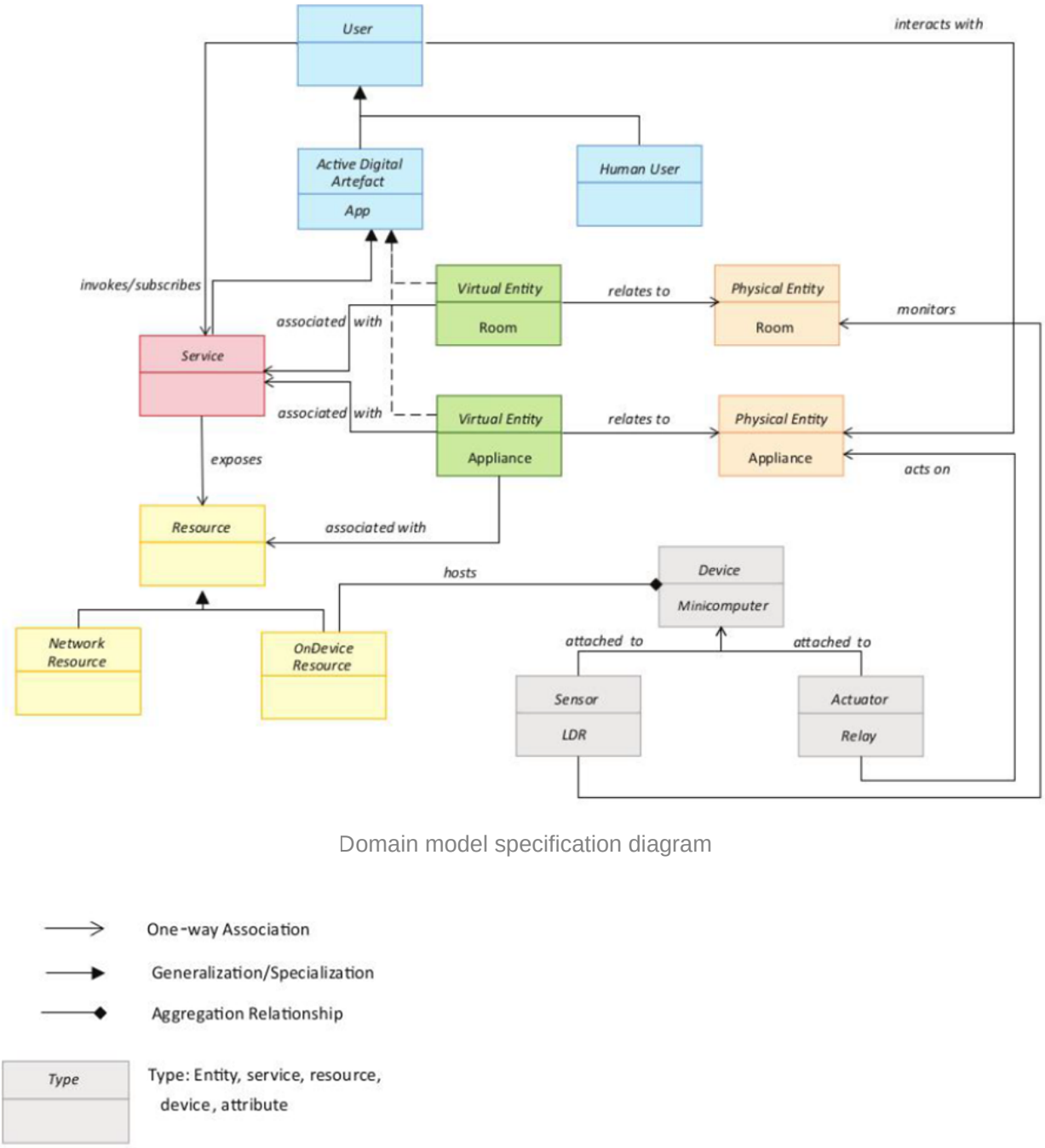
3. Domain Model Specification

The domain model describes the main concepts, entities and objects in the domain of the IoT system to be designed. Domain model defines the attributes of the objects and

relationships between objects. The domain model is independent of any specific technology or platform.

Using domain model, system designers can get an understanding of the IoT domain for which the system is to be designed. The entities, objects and concepts defined in the domain model of home automation system include the following:

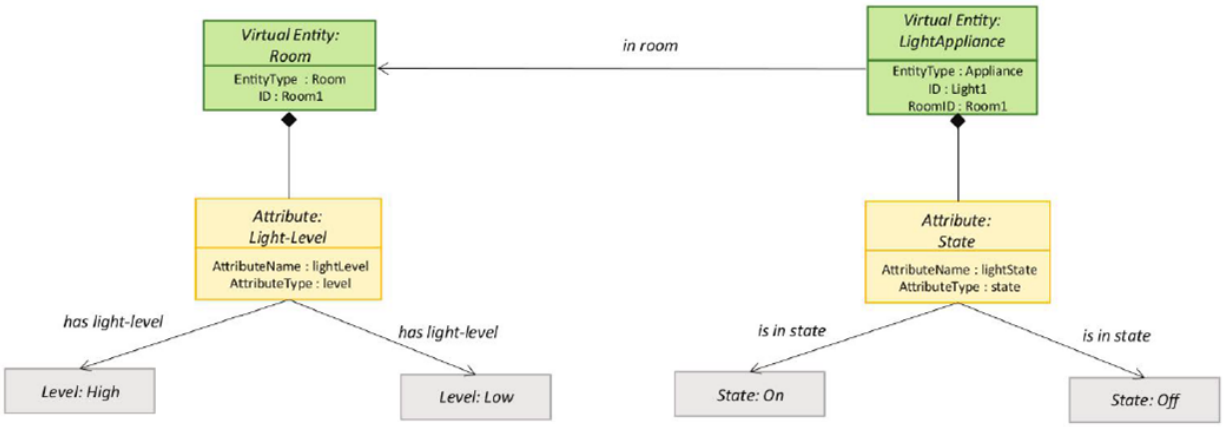
Physical Entity	• The physical identifiable objects in the environment • IoT system provides information about the physical entity (using sensors) or performs actuation upon the physical entity
Virtual Entity	• Virtual entity is a representation of the physical entity in the digital world • For every physical entity there is a virtual entity
Device	• Devices provide a medium for interaction between physical and virtual entities • Devices are used to gather information from or perform actuation on physical entities
Resource	• Resources are software components which can be either on-device or network-resources • On-device resources are hosted on the device and provide sensing or actuation (eg: operating system) • Network-resources include software components that are available on the network (eg: database)
Service	• Services provide an interface for interacting with the physical entity • Services access resources to perform operations on physical entities



Domain model specification diagram

4. Information Model Specification

Information model defines the structure of all the information in the IoT system. Does not describe how the information is stored and represented. To define the information model, we first list the virtual entities. Later more details like attributes and relationships are added. The information model specification for home automation system is as shown below:

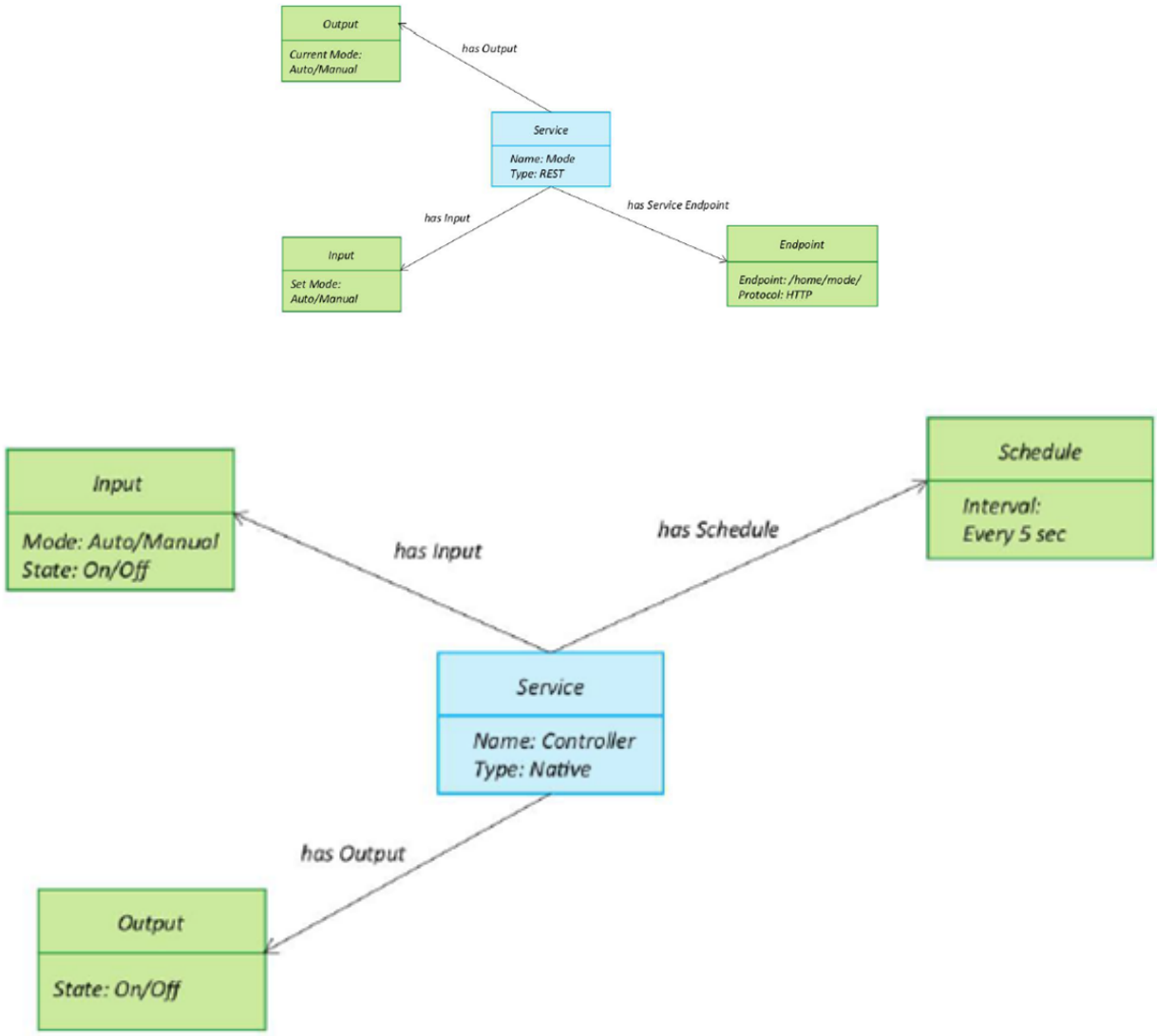


5. Service Specifications

The service specification defines the following:

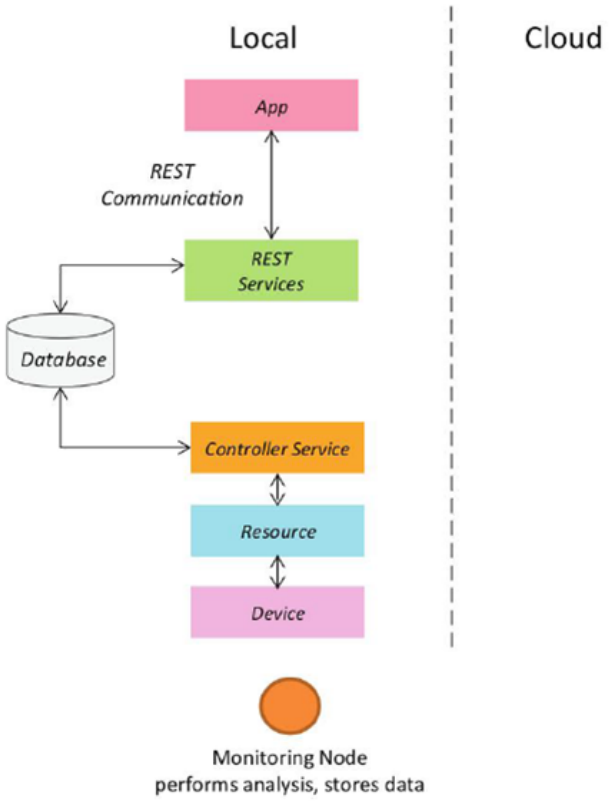
- Services in the system
- Service types
- Service inputs/output
- Service endpoints
- Service schedules
- Service preconditions
- Service effects

For each state and attribute in the process specification and information model, we define a service. Services either change the state of attributes or retrieve their current values. The service specification for each state in home automation systems are as shown below:



6. IoT Level Specification

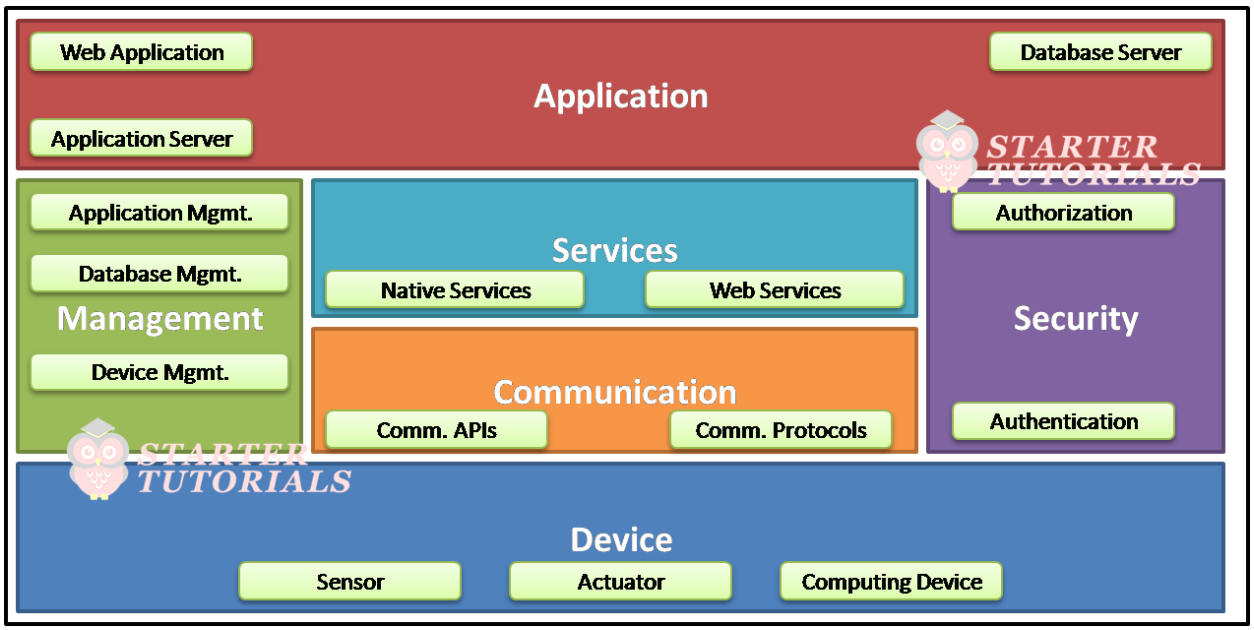
Based on the requirements we will choose the IoT application deployment level. The deployment level for home automation system is shown in the below figure.



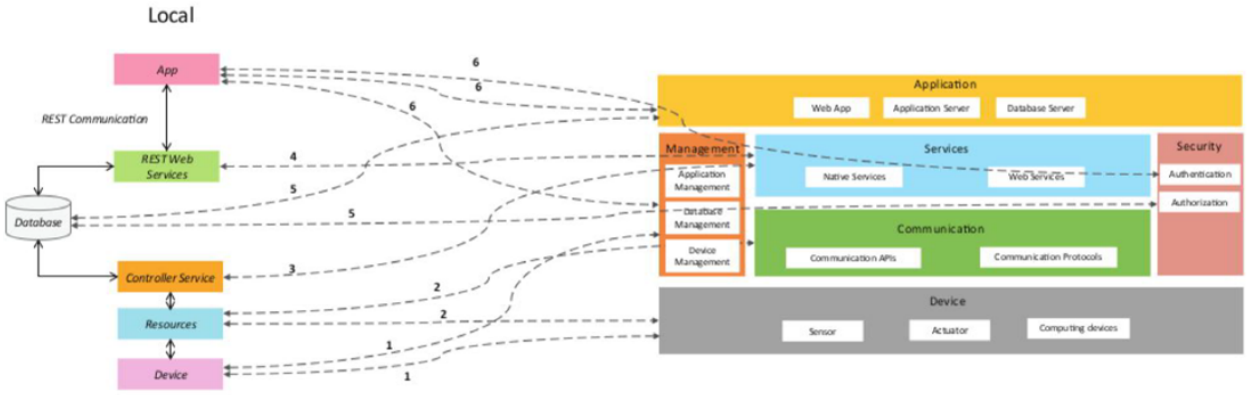
7. Functional View Specification

The functional view defines the functions of the IoT systems grouped into various functional groups. Each functional group provides functionalities for interacting with concepts in the domain model and information related to the concepts.

The functional groups in a functional view include: Device, Communication, Services, Management, Security, and Application. The functional view specification for home automation system is shown in the below figure:



Functional view Specification



Mapping between the IoT level and the functional groups.

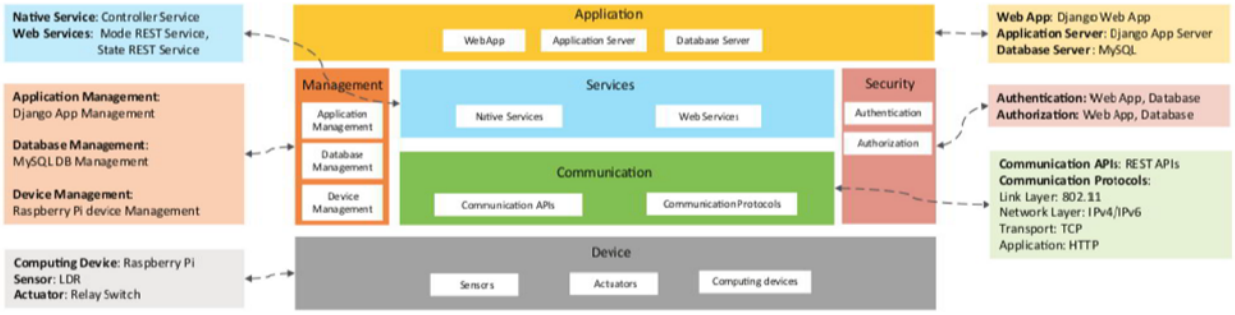
8. Operational View Specification

In this step, various options related to the IoT system deployment and operation are defined, such as:

- Service hosting options
- Storage options
- Device options

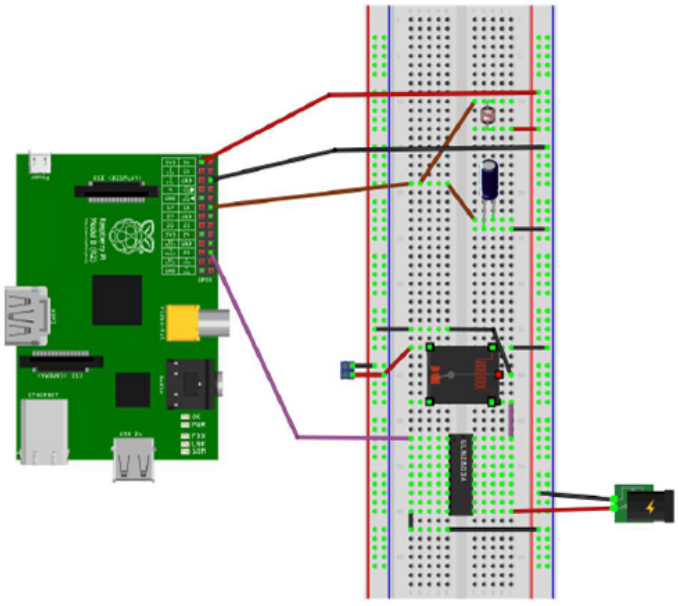
- Application hosting options

The options chosen for home automation system are as shown in the below figure.



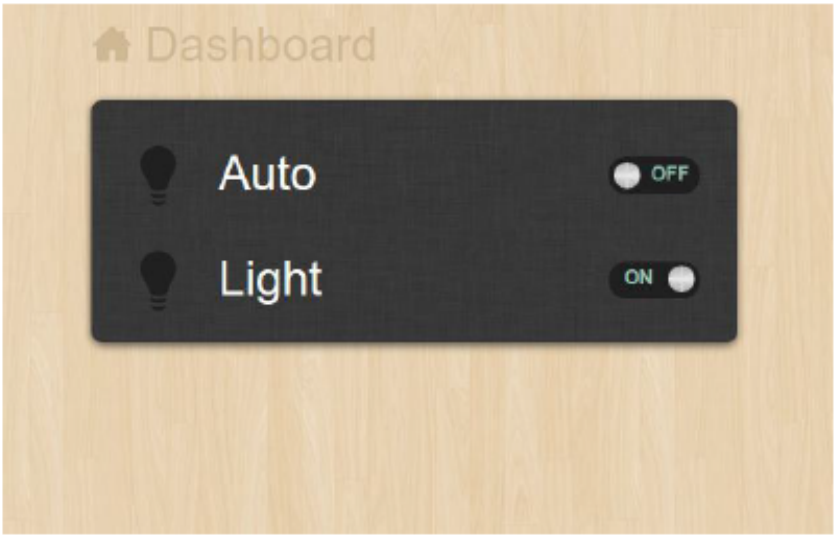
9. Device and Component Integration

In this step the devices like sensors, computing devices and other components are integrated together. The interconnection of different components in our home automation system are as shown in the figure given below.



10. Application Development

Using all the information from previous steps, we will develop the application (code) for the IoT system. The application interface for home automation system is shown below.



- [x] ~~1.logical design of IOT~~
- [x] ~~2.communication model~~
- [x] ~~3.IOT LEVELS~~
- [x] ~~4.SMTP LIB,HTTP LIB~~
- [x] ~~5.interface~~